Communicative Learning with Natural Gestures for Embodied Navigation Agents with Human-in-the-Scene

Qi Wu[1] Cheng-Ju Wu[1] Yixin Zhu[1] Jungseock Joo[1*]

Abstract— **Human-robot collaboration is an essential research topic in artificial intelligence (AI), enabling researchers to devise cognitive AI systems and affords an intuitive means for users to interact with the robot. Of note, communication plays a central role. To date, prior studies in embodied agent navigation have only demonstrated that human languages facilitate communication by instructions in natural languages. Nevertheless, a plethora of other forms of communication is left unexplored. In fact, human communication originated in gestures and oftentimes is delivered through multimodal cues, *e.g.*, "go there" with a pointing gesture. To bridge the gap and fill in the missing dimension of communication in embodied agent navigation, we propose investigating the effects of using gestures as the communicative interface instead of verbal cues. Specifically, we develop a VR-based 3D simulation environment, named Gesture-based THOR (Ges-THOR), based on AI2-THOR platform. In this virtual environment, a human player is placed in the same virtual scene and shepherds the artificial agent using only gestures. The agent is tasked to solve the navigation problem guided by natural gestures with unknown semantics; we do not use any predefined gestures due to the diversity and versatile nature of human gestures. We argue that learning the semantics of natural gestures is mutually beneficial to learning the navigation task—*learn to communicate and communicate to learn*. In a series of experiments, we demonstrate that human gesture cues, even without predefined semantics, improve the object-goal navigation for an embodied agent, outperforming various state-of-the-art methods.**

I. Introduction

Human-human communication takes place in various forms, of which gestures play a crucial role [1]. Gestures include movements of body, head, or hands and can facilitate the understanding of the speech or serve as emblems to deliver messages in place of speech [2,3]. They can significantly improve the communication efficacy for information conveyance [4].

Similarly, human-robot communication can also occur using multimodal cues [5]. Although robots and autonomous systems are designed to collaborate with humans who supervise, instruct, or evaluate the system to perform specific tasks, most existing communication interfaces assume that humans communicate to an artificial agent only using natural language, either verbally or through text. In stark contrast, the origin of human communications is primarily rooted in nonverbal forms [6], *e.g.*, gestures. Therefore, providing assistive or collaborative AI systems with nonverbal means of communication would open up new research venues to investigate the efficacy of alternative communication forms. Unlike natural languages, which suffer from intermittent conveyance and need continuous attention, nonverbal cues like gestures are immediate and intuitive, hence are less vulnerable to interruptions. In particular, when the environment is noisy, or the agent is listening to someone else, a user might refer to a location using a deictic gesture, *e.g.*, pointing with a finger, instead of describing it with a long sentence.

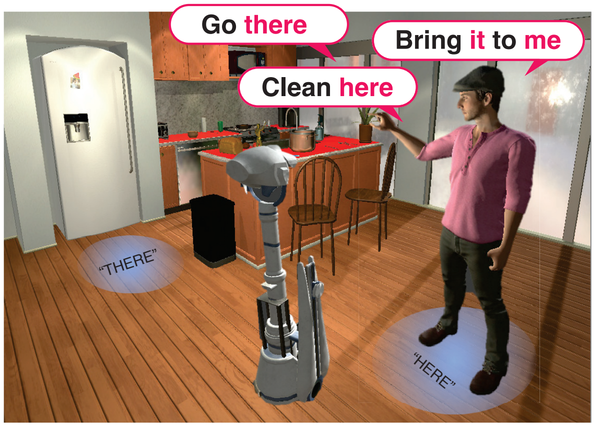

Fig. 1: **Natural gestures can succinctly deliver complex semantic meaning in a physical space.** A human user can instruct a robot or a virtual agent to complete a navigation task by simply referring to a target location or object using gestures. The agent should infer the user intent from the gestures.

To illustrate the significance of communications using gestures, let us take Fig. 1 as an example. A human intends to instruct the robot to navigate to the target location or object in the scene. Previous works in embodied visual navigation with language-based human interactions may require a lengthy text message, such as "go to the second brown chair next to the big white table in the living room". In contrast, gestures allow to express the same message in a much simpler and more natural way, *e.g*. "go there," "clean here," or "bring it to me." Such multimodal messages can only be correctly interpreted in a given physical space where a human and an agent are situated together. The meaning of a human message must be inferred from the joint understanding of the given scene by the agent who also understands the semantics of human natural gestures [7,8].

Inspired by the above crucial observation, we intend to bring in nonverbal communication cues [1,9,10] into the embodied agent navigation task—the most straightforward task of an embodied AI that interacts with the environments and other agents. Despite the progress reported for the embodied agent on the Vision-Language Navigation (VLN) task [11–19], we contemplate on prior arts and quest for the following questions: Instead of using natural language, can we replace the language grounding by gestures in a similar setting? Can we improve the performance of navigation with

[*] Corresponding author. Email: jjoo@comm.ucla.edu

[1] UCLA. Emails: {qi.wu,jimmy.wu,yixin.zhu}@ucla.edu

The work reported herein was supported by NSF NRI #1925360.

gestures incorporated? Can the learning agent acquire the underlying semantics of gestures, even when they are not predefined?

Specifically, we aim to use gestures to communicate with an embodied agent to navigate in a virtual environment. To provide gesture-based instructions for a navigation task, the agent needs a photorealistic simulation environment, and a human player needs to be situated in the same scene to have *joint attention* [20]. To support such a co-existing environment, we build our virtual environment Ges-THOR with Oculus, Kinect, and Leap Motion, based on the existing framework AI2-THOR [21].

Although human gestures have been used as a communicative interface between humans and robots in robotics [22–26], prior literature typically predefines the vocabulary of admissible gestures and their definite meanings (*e.g.*, "ok" sign means an approval). In contrast, human gestures are diverse; their meanings are also non-rigid and context-dependent [27]. One needs to develop a flexible system to address the variability and versatility of nearly-unlimited naturalistic human gestures without a predefined set of recognizable gestures. Without defining any gestures and their meanings ourselves, we have collected demonstrations from a group of volunteers who have diverse gesture preferences for the same message.

In our proposed framework, an agent should therefore solve two tasks: multimodal target inference and navigation. Inferring the meanings of human gestures and finding a path to the target location are two major goals of the agent, which mutually help each other, *i.e.* ***learn to communicate and communicate to learn.*** Experiments reveal that our model incorporating gestures outperforms a baseline model only with vision for navigation, as well as models on similar environments and tasks using different methods [28].

This paper makes four contributions: (i) By introducing human gestures as the new communicative interface for embodied AI learning and (ii) developing a simulation framework, Ges-THOR, that supports multimodal interactions with human users, (iii) we demonstrate that the embodied agent's navigation performance significantly improves after incorporating human gestures. (iv) We further demonstrate that the agent can learn the underlying meanings and intents of human gestures without predefining the associations.

II. Related Work

Language grounding: Language grounding is crucial for both parties involved in communication to understand each other. Natural language, the most common modality for human-human and human-robot communication, can realize the grounding in various ways. For communication with robots, language can be interpreted from instructional commands to actions [29, 30]. For static images or texts, it can be either visually grounded [31, 32] or text-based [33] Q&A. In our work, language grounding is replaced by "gesture grounding;" we provide gestures as the new communicative interface. The agent is tasked to learn by grounding human gestures into a series of actions and identify target objects.

Vision-language navigation: Image captioning with large datasets [34] and Visual Question Answering (VQA) [31, 35] has made significant progress in vision and language understanding, which enables visually-grounded language navigation agent to be trained. Many tasks following the VLN framework [11, 12, 14–19, 36–39] have been addressed and solved using end-to-end learning models, either in 2D world [40–42], 3D world [12, 43], or even photorealistic environments [15, 17–19, 44]. Some works have also explored the acoustic cue in navigation, but these are not mainly concerned with speech [45, 46]. Our work is built on the existing VLN framework but extended by incorporating gestures as a new modality for communications.

Simulated environments: To help the research in embodied AI learning, various simulated environments have spurred for the community's benefit. Those 3D environments are created from either synthetic scenes [21, 44, 47–50] or real photographs [51–54]; some of them use game engines to enable physical interactions [21, 44, 55–57]. In this paper, we choose AI2-THOR, which uses Unity as the physics engine, and build the environment on it. Exiting works using AI2-THOR for visual navigation tasks [28, 58] require either the target visualization or its context. In this paper, we propose a gesture-based method to eliminate the need for acquiring additional target information.

RL for navigation: Instead of using traditional path-planning approaches [59] to compute a route to the goal location, the embodied AI community has recently focused more on end-to-end learning for training navigation policies, especially with Reinforcement Learning (RL). Compared with other machine learning methods, such as supervised learning [60], RL benefits from simple reward definitions and easy implementations. As a result, RL becomes the core of the learning framework [13, 15, 28, 52, 53, 58]. In this paper, we choose Proximal Policy Optimization (PPO) [61] as the RL model.

Human AI interaction: Human-AI Interaction (HAI) has been intensively investigated in AI, Human-Computer Interaction, and robotics [62–64]. For the embodied navigation agents, the sprout of simulated environments makes users communicate with the agent interactively. Most existing frameworks achieve this goal using dialogues [14, 37, 40, 65–67]. However, as discussed, natural language is not the only cue for multimodal communication, and current collaborative frameworks have not yet fully explored a rich spectrum of communicative interfaces for embodied agent navigation. In this paper, we propose gestures as the communicative interface between human users and the artificial agent.

Meanwhile, there is a large body of work on human gestures as a communicative device either to humans or robots [68, 69]. Most of these approaches are based on a predefined gesture set with fixed meanings or focus on gesture type classification [70], pose estimation [71, 72], or both [73, 74]. In contrast, we let users use any natural gestures and demonstrate the agent can directly learn the semantics and underlying intents of these gestures.

III. Ges-THOR: A Simulation Framework for Human-Agent Interaction via Gestures

We build an interactive learning framework in Unity based on the iTHOR environment from AI2-THOR for the gesture-

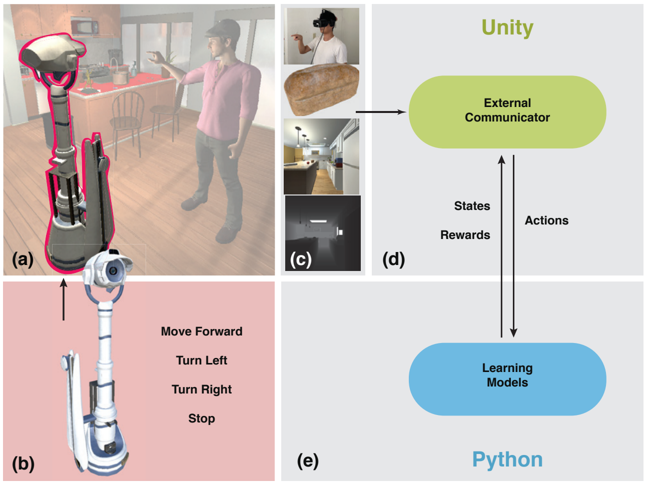

Fig. 2: **Overview of the learning framework.** (a) Scenes and the (b) learning agent are built in Unity. The agent can perform four actions: move forward, turn left, and turn right, and stop. (c) It receives several sensory inputs, including RGBD images, target labels, and human gestures. Unity contains (d) an external communicator that can communicate with the (e) learning model in PyTorch. The learning model receives states and rewards from the communicator and sends back chosen actions.

boosted embodied AI research, namely Ges-THOR—Gesture-based iTHOR environment.

A. Simulation and Learning Framework

There are many existing physics-based simulation frameworks for photorealistic indoor navigation tasks [21, 44, 51–53, 57]. We choose AI2-THOR specifically to build our learning environment because it provides a diversity of rooms and interactive features. It has been widely used for different visual navigation tasks [28, 58]. In addition, the game engine Unity provides the ability to deploy across platforms and integrate third-party resources, compatible with the sensory devices we use for this learning environment. We also use AllenAct [75] as the codebase for our modular framework.

B. Human Gesture Sensing

The following setup immerses human players into the virtual environment while allowing the system to capture human gestures:

Devices: We use Oculus Rift, Kinect Sensor v2, and Leap Motion Controller (hereinafter referred to as Oculus, Kinect, and Leap Motion) together for gesture sensing via pose estimation. Oculus gives the player the first-person view in the virtual environment; hence the player sees the virtual scene and knows where the target object is. Kinect is used to track overall body movements. However, Kinect is incapable of capturing fine in-hand motions. Leap Motion is brought in to detect hand movements.

Device Arrangement: Fig. 3 illustrates the device arrangement. During data acquisition, the human player is asked to wear the Oculus headset, face the Kinect sensor, and move hands in front of Leap Motion at a distance between 30cm and 60cm. In Unity, a humanoid character (see Fig. 3d) mirrors players' movements in real-time, including body composure and hand motions.

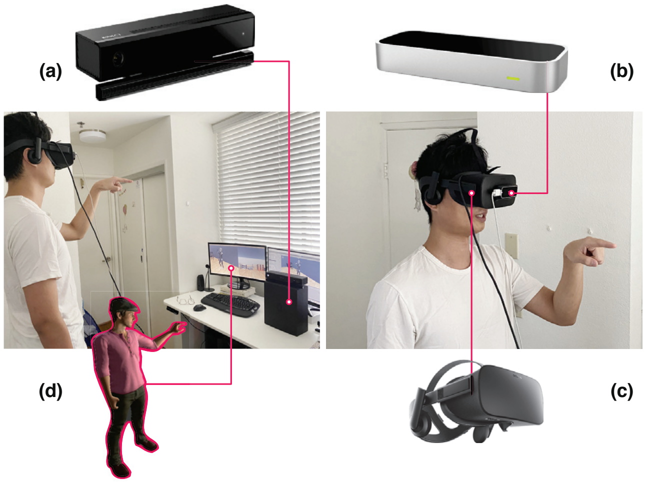

Fig. 3: **Device arrangement.** The human player wears an (c) Oculus headset with (b) Leap Motion. (a) Kinect is placed 1.5m from the human player and 1.5m above the ground. The screen displays (d) A humanoid model that are mirroring the body and hand movements.

Data collection: Ideally, learning would take place in real-time, where a human player continuously observes an agent's behavior and interacts with it such that the agent can respond to the feedback immediately. Unfortunately, this is infeasible because the entire training process may take hundreds of thousands of episodes. Therefore, we opt for using pre-recorded gestures to simulate real-time interactions between the human player and the agent as closely and efficiently as possible. There are two types of instructional gestures that humans can use: one is for referencing, and the other one is for intervention. To record referencing gestures, a volunteer is given the target object in a scene and asked to communicate with the agent to guide the direction with a gesture. We do not ask participants to use any specific gestures such as pointing with a finger but encourage them to use any gestures as if they are talking to another person. The gesture sequence, as well as the environmental information, is recorded as one episode in the dataset. We have over 230,000 unique episodes for training and 2,500 episodes for validation and testing. For intervention gestures, the player shows gestures in a rejective manner used to warn the agent if it is moving away from the target. We recorded ten different intervention gestures. Kinect Body and Leap Hands can duplicate the player's movements and save the recorded motions as animation clips in Unity. See Fig. 4 for examples of collected gestures.

C. Sensory Modalities

Multimodal perception is essential for artificial systems. We provide several sensory inputs in our environment to build a multimodal learning framework; see Fig. 2. The observational space consists of the following inputs:

Vision: Unity's built-in camera component allows a 2D view of the virtual space. It is attached to the eyesight of the embodied agent at 1.5m from the ground with a 90-degree field-of-view and provides real-time RGB images in the first-person view. The resolution of the RGB images is $3 \times 224 \times 224$, and each pixel contains scaled RGB values from 0 to 1.

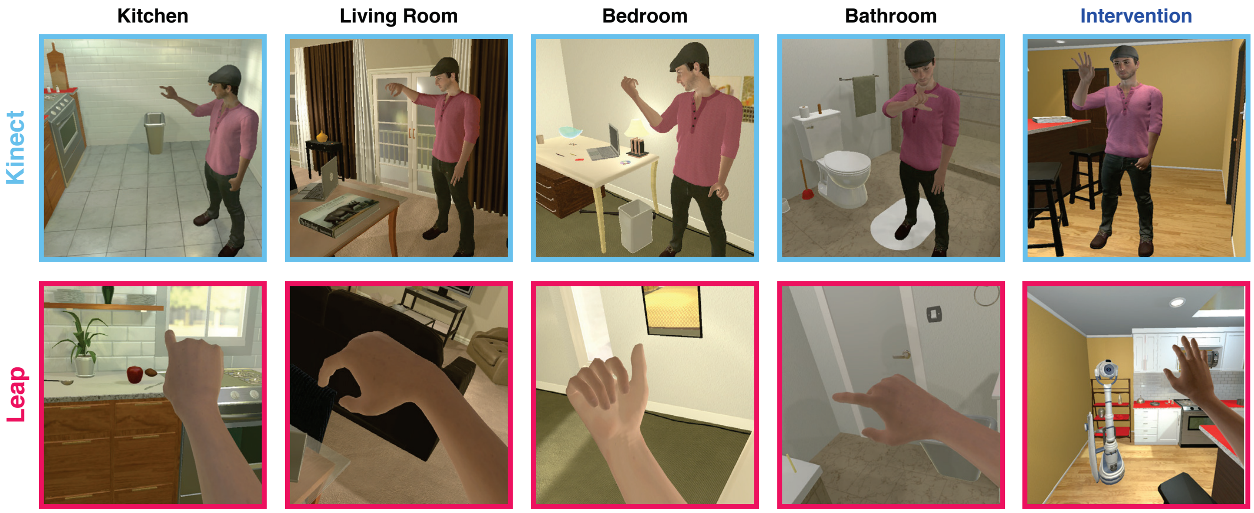

Fig. 4: **Examples of referencing and intervention gestures.** The first 4 columns are referencing gestures, while the last one is an intervention gesture. Human players perform different gesture styles while pointing at various target objects in the scene. Top row shows the body movements captured by Kinect, and the bottom row shows the hand configurations recorded by Leap Motion.

Depth: The depth image is extracted from the depth buffer of Unity's camera view. It has a size of $1 \times 224 \times 224$, and each pixel value is a floating-point between 0 and 1, representing the distance from the rendered viewpoint to the agent, linearly interpolated from 0m to 10m.

Collision: Unity checks for collisions dynamically in the learning environment. Every time the agent triggers a collision, it can report this event and prevent the agent from penetrating into the object meshes. Note that for our agent design, it can slide along the object surface it collides with. This "sliding" mechanics has been noticed by recent work [76] and may hinder sim2real transition. We rectify this issue by addressing penalties in rewards for such behaviors.

Gesture: As previously mentioned, we use Oculus, Kinect, and Leap Motion to capture human gestures. Each gesture motion is saved as a sequence of vectors with 100 steps and 95 features consisting of body and hand poses. Note that for referencing gestures, we select motions from the corresponding episode. For intervention gestures, we randomly sample one from saved recordings and use it only when the agent faces away from the target. The raw gesture inputs are encoded and piped into our learning model.

IV. Learning to Navigate with Gesture

In this section, we describe our end-to-end gesture learning model using Deep Reinforcement Learning (DRL). We start by introducing the formulation of the DRL model we use, followed by the other components of the entire architecture.

A. Problem Formulation

We take the ObjectGoal task [77] as our navigation task, where the agent must navigate to an object of a specific category, as our experimental testbed. The details of the task and the agent embodiment are explained below:

Agent Embodiment: The learning agent is represented by a robot character with a capsule bound. The agent has a rigid body component attached to it so that it can detect collisions with environmental objects. It has four available actions: *turn left*, *turn right*, *move forward*, and *stop*. Each turning action results in a rotation of 15°, and each forward action results in a forward displacement of 0.25m.

Task Definition: The agent is initiated at a random location, and an object is selected randomly as the target; we ensure that the agent can reach the target. Note that there can be more than one instance of the target object type in the same environment. To complete the task, the agent must navigate to the target object instance with a stopping distance equal to or less than 1.5m. The agent then needs to issue a termination (*i.e.*, *stop*) action in the proximity of the goal, and the object must also be within the agent's field of view in order to succeed. An episode is terminated if the above success criteria are met or the maximum allowed time step (which is 100 in our setup) is reached. We allow the agent to issue multiple stops in an episode but measure success rates using different numbers of maximum stops (1-3). We allow an unlimited number of stops in training; the agent needs to explore and learn after issuing incorrect stops in earlier episodes.

B. Policy Learning with PPO

We formulate our visual gesture navigation using DRL, specifically PPO. Our learning process can be viewed as a Markov Decision Process (MDP). At each time step t, the agent perceives a state s_t (*i.e.*, a combination of the sensory inputs), receives a reward r_t from the environment, and chooses an action a_t according to the current policy π:

$$a_t \sim \pi_\theta(a_t|s_t), \quad (1)$$

where θ represents parameters for the function approximator of the policy π. We implement PPO with a time horizon of 128 steps, batch size of 128 and 4 epochs for each iteration of gradient descent, and buffer size of 1280 for each policy update. We use Adam [78] as the optimizer with a learning rate of 0.0003 and a discount factor of 0.99.

The agent receives a positive reward of $+1$ if it completes the navigation successfully. Since we encourage the agent to

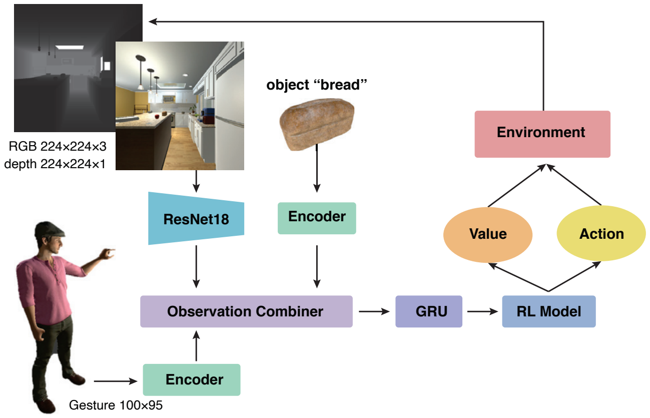

Fig. 5: **The model overview.** Our model fuses perceptions from different sensory modalities, and the actor-critic model samples an action in each step according to the updated policy and send it back to the environment.

reach the target object with the minimal amount of steps, the agent receives a small time penalty of -0.001 for each step. We further add a collision penalty of -0.005 for each collision detected; the collision penalty is added to mitigate the aforementioned "sliding" behavior. If the agent stops in an ineligible location, a penalty of -0.01 is added.

C. Model Overview

We equip the embodied agent with different sensory modalities, and each of them feeds into a part of the input network for the RL model. Below we introduce these components of the architecture.

Visual network: The backbone of the visual network is ResNet-18 [79] pre-trained on ImageNet. It takes 224 × 224 RGB and depth images as inputs. The weights of all layers in the network except the last fully connected layer are frozen during training.

Gesture network: The raw input of the gesture network is a sequence with 100 time steps and 95 features. Each feature represents the muscle value from the Unity humanoid model, which can be considered as the coordinates for tracked body and hand joints. The gesture input is flattened and encoded into a vector. In addition, we provide the target object category from a selected set and pass it to an embedding layer. This is equivalent to speech or text instructions from a user in prior work on interactive embodied agent learning. Since our focus in this paper is gesture, we simplify this part of the input as a categorical variable (*e.g.*, a single word in a fixed vocabulary). Note that this vector alone does not specify the target object location: There can be multiple instances of the same object category in the scene, and the agent needs to infer which instance the human player is referring to. This vector is concatenated with encoded gesture inputs and visual features by an observation combiner. There is a memory unit using Gated Recurrent Unit (GRU) [80] after this combiner. Fig. 5 illustrates the entire architecture.

V. Evaluation

We evaluate our methods in Ges-THOR environment. AI2-THOR provides 120 scenes covering four different room types: kitchen, living room, bedroom, and bathroom. Each room has its own unique appearance and arrangements. We randomly split 30 scenes for each scene type into 20 training rooms, 5 validation rooms, and 5 testing rooms.

There are 38 object categories available for all scenes. Since there is almost no overlap of objects for different scene types, we train and evaluate separately for each scene type. We evaluate each scene for 250 episodes and report the average results for each scene type.

Evaluation Metrics: We use 2 metrics to evaluate different methods:

- SR: for the i-th episode, the success can be marked by a binary indicator S_i. The success rate is the ratio of successful episodes over completed episodes N:

$$SR = \frac{1}{N}\sum_{i=1}^{N} S_i. \tag{2}$$

- SPL: this metric is proposed by Anderson *et al.* [77]. It measures the efficacy of the navigation. SPL is calculated as follows:

$$SPL = \frac{1}{N}\sum_{i=1}^{N} S_i \left(\frac{l_i}{\max(p_i, l_i)}\right), \tag{3}$$

  where l_i is the shortest path distance from the agent's starting position to the goal in episode i, and p_i is the actually path length taken by the agent.

We have three methods to evaluate the agent performance: (1) Baseline: the agent only has the visual (*i.e.*, RGB and depth images) and object category information. (2) Referencing Gesture: in addition to (1), the agent receives referencing gesture inputs. (3) Intervention Gesture: in addition to (1), the agent receives rejective gesture inputs when the forward direction forms an angle larger than 90 degrees between the agent and the target.

In our comparative setting, the baseline model does not use any gestures. While one may expect that it should always underperform, this is only true if the agent has learned and inferred the semantics of human gestures and incorporated the signals during navigation, which is the focus of our evaluation. Again, this is not trivial because we do not pre-define the meaning of any gestures. Similarly, the intervention gestures is a strong directive feedback from the human user, but we evaluate how well the agent can infer its meaning and adopt it in navigation.

Navigation Performance: Table I show the performance of different methods when evaluated at the first stop, and Table II show the performance at test scenes evaluated at a different number of stops. From the both results, we confirm that adding gestures can significantly improve the navigation success rate as well as the efficiency over the baseline model. Table I puts a hard constraint on the number of stops to 1 to match the state-of-art benchmarks [77]. Of note, models trained with intervention gestures outperform models trained with referencing gestures, both in SR and SPL, demonstrating that intervention gesture is a more effective kind of gesture to communicate with the agent. Table II reports results on test scenes with a different number of allowed stops. We should see that both SR and SPL increase with the number of allowed stops, and the improvement of

Scene Types	Methods	Success Rate (%)			Success weighted by Path Length (%)		
		Train	Validation	Test	Train	Validation	Test
Kitchen	Baseline	12.3	10.2	11.5	7.6	7.1	7.9
	Referencing	21.1	18.7	19.2	13.3	11.6	11.2
	Intervention	**44.9**	**31.5**	**40.3**	**27.0**	**20.0**	**24.0**
Living Room	Baseline	6.3	3.6	3.5	4.1	2.3	2.1
	Referencing	4.9	3.2	2.7	3.2	1.7	1.6
	Intervention	**13.0**	**9.0**	**9.5**	**7.8**	**5.3**	**5.4**
Bedroom	Baseline	15.2	9.1	8.7	9.1	5.3	5.4
	Referencing	**43.5**	10.7	15.4	**28.3**	6.6	10.5
	Intervention	42.4	**22.4**	**20.4**	27.5	**13.8**	**11.9**
Bathroom	Baseline	16.3	15.5	11.9	11.4	9.1	8.8
	Referencing	33.0	19.4	19.9	20.5	11.1	11.7
	Intervention	**40.5**	**32.2**	**35.0**	**29.1**	**21.0**	**23.0**
Average	Baseline	12.5	8.1	9.9	6.2	8.6	5.9
	Referencing	25.6	16.3	13.0	7.8	14.3	8.8
	Intervention	**35.2**	**22.9**	**23.8**	**15.0**	**26.3**	**16.1**
	Scene Prior [28]*			13.4			6.7

TABLE I: **Evaluation results for train/validation/test split.** Success Rate (SR) and Success weighted by Path Length (SPL) at the first stop are reported in this table. We compare models trained with referencing gestures and intervention gestures against a baseline model. * Reported from [28]. This method uses additional scene prior knowledge but not gestures.

Scene Types	Methods	Success Rate (%)				Success weighted by Path Length (%)			
		1 Stop	2 Stop	3 Stop	∞	1 Stop	2 Stop	3 Stop	∞
Kitchen	Baseline	11.5	18.0	23.1	49.3	7.9	12.1	14.6	25.1
	Referencing	19.2	26.3	29.7	47.9	11.2	15.0	16.6	24.1
	Intervention	**40.3**	**55.1**	**62.8**	**89.0**	**24.0**	**32.7**	**37.2**	**52.8**
Living Room	Baseline	3.5	6.4	9.5	23.3	2.1	3.7	5.1	9.8
	Referencing	2.7	3.9	4.7	7.8	1.6	2.4	2.9	4.7
	Intervention	**9.5**	**16.9**	**21.7**	**58.0**	**5.4**	**9.8**	**12.6**	**30.8**
Bedroom	Baseline	8.7	15.3	18.4	31.0	5.4	9.5	11.2	17.0
	Referencing	15.4	18.7	20.2	31.9	10.5	12.4	13.3	19.2
	Intervention	**20.4**	**27.9**	**33.5**	**51.2**	**11.9**	**16.3**	**19.5**	**29.7**
Bathroom	Baseline	11.9	18.9	23.7	57.8	8.8	13.8	16.9	33.1
	Referencing	19.9	30.5	35.9	64.3	11.7	18.1	21.3	32.4
	Intervention	**35.0**	**44.6**	**51.7**	**76.5**	**23.0**	**29.6**	**33.9**	**48.3**
Average	Baseline	8.9	14.7	18.7	40.4	6.1	9.8	12.0	21.3
	Referencing	14.3	19.9	22.6	38.0	8.8	12.0	13.5	20.1
	Intervention	**26.3**	**36.1**	**42.4**	**68.7**	**16.1**	**22.1**	**25.8**	**40.4**

TABLE II: **Evaluation results for test scenes with different number of allowed stops (∞ denotes infinte allowed stops).** SR and SPL are presented. We compare models trained with referencing gestures and intervention gestures against a baseline model.

SR and SPL with gestures is more evident in a lower number of allowed stops.

Qualitative Results: To visualize the effectiveness of our methods, we show some qualitative results in Figs. 6 and 7. Fig. 6 compares our referencing gesture model against the baseline model with visualized trajectories in different scenes and targets. It could be observed that in all scenes, our referencing gesture model enables the agent to navigate to the target more intelligently, while the baseline model often struggles to find the target and stop or takes a longer path to find the target. Fig. 7 demonstrates how our intervention gesture model works to improve the navigation significantly. In this example, the agent rotates at the place where it faces back to the target and is instructed with interventions gestures until the target is in its field of view before making any movements. This indicates that our agent is able to understand and react to the intervention gestures, resulting in much better navigation performance.

VI. Conclusion

In this paper, we propose a new framework for embodied visual navigation where human users can give instructions to the autonomous agent using gestures. Such agents and gesture based interface will be very useful for collaborative robots or virtual agents. We have built a VR-based interactive learning environment, Ges-THOR, based on AI2-THOR and designed an end-to-end deep reinforcement learning model for the navigation task. Our experiments show that the agent is able to interpret human instructions with gestures and improve its visual navigation. We also conclude that

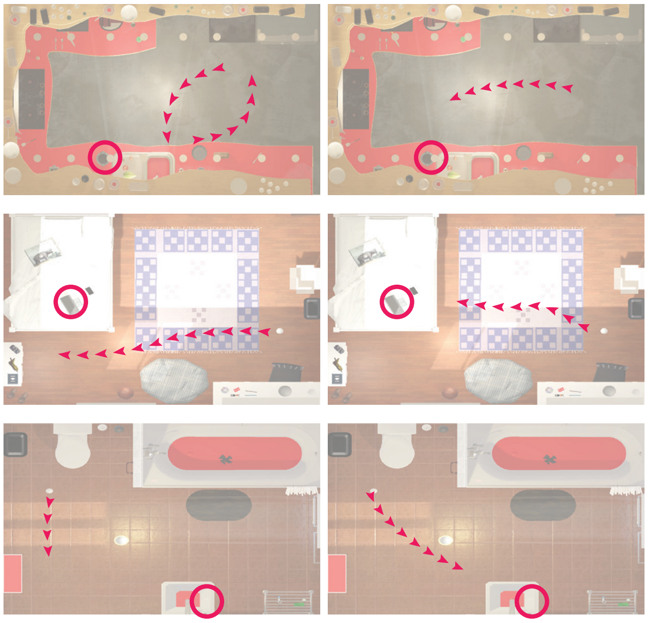

Fig. 6: **Qualitative results with visualizations of trajectories for baseline (left) and referencing gesture (right) models.** Our agent can efficiently navigate to the target with the help of gestures.

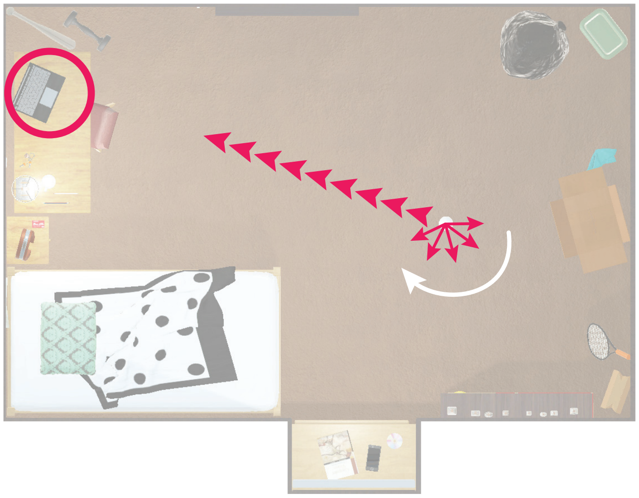

Fig. 7: **Qualitative results for the intervention gesture model.** When the agent is back to the target, it receives interventional gestures and could first rotate until it faces the target before making any forward movements, thus increasing the navigation success rate and efficiency.

interactive activities during agent task execution can improve performance. While the main setting and experimental design of our study have been used in prior works, to the best of our knowledge, our paper is the first incorporating human gestures for embodied agent learning and showing the agent can learn the semantic of gestures without supervision. We will make publicly available our simulation environment and the recorded gesture dataset for any future research for Human-AI interaction via gestures. The future directions include adding more objects, tasks, gestures, and multiple agents in the scene, *e.g.*, navigating to an object and bring it back by showing gestures in our framework and also allowing agents to make gestures to the human player such that both parties can communicate with gestures, which will also help humans to utilize even more diverse gestures to communicate with agents.

References

[1] J. Joo, F. F. Steen, and M. Turner, "Red hen lab: Dataset and tools for multimodal human communication research," *KI-Künstliche Intelligenz*, vol. 31, no. 4, pp. 357–361, 2017.

[2] J. Joo, E. P. Bucy, and C. Seidel, "Automated coding of televised leader displays: Detecting nonverbal political behavior with computer vision and deep learning.," *International Journal of Communication (19328036)*, 2019.

[3] Z. Kang, C. Indudhara, K. Mahorker, E. P. Bucy, and J. Joo, "Understanding political communication styles in televised debates via body movements," in *European Conference on Computer Vision*, pp. 788–793, Springer, 2020.

[4] P. Bremner and U. Leonards, "Iconic gestures for robot avatars, recognition and integration with speech," *Frontiers in Psychology*, vol. 7, p. 183, 2016.

[5] Y. Zhu, T. Gao, L. Fan, S. Huang, M. Edmonds, H. Liu, F. Gao, C. Zhang, S. Qi, Y. N. Wu, J. Tenenbaum, and S.-C. Zhu, "Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense," *Engineering*, vol. 6, no. 3, pp. 310–345, 2020.

[6] M. Tomasello, *Origins of human communication*. MIT press, 2010.

[7] F. Steen and M. B. Turner, "Multimodal construction grammar," *Language and the Creative Mind. Borkent, Michael, Barbara Dancygier, and Jennifer Hinnell, editors. Stanford, CA: CSLI Publications*, 2013.

[8] F. F. Steen, A. Hougaard, J. Joo, I. Olza, C. P. Cánovas, *et al.*, "Toward an infrastructure for data-driven multimodal communication research," *Linguistics Vanguard*, vol. 4, no. 1, 2018.

[9] L. Fan, S. Qiu, Z. Zheng, T. Gao, S.-C. Zhu, and Y. Zhu, "Learning triadic belief dynamics in nonverbal communication from videos," in *CVPR*, 2021.

[10] H. Joo, T. Simon, M. Cikara, and Y. Sheikh, "Towards social artificial intelligence: Nonverbal social signal prediction in a triadic interaction," in *CVPR*, pp. 10873–10883, 2019.

[11] D. L. Chen and R. J. Mooney, "Learning to interpret natural language navigation instructions from observations," in *AAAI*, 2011.

[12] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. Salakhutdinov, "Gated-attention architectures for task-oriented language grounding," *arXiv:1706.07230*, 2017.

[13] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *CVPR*, 2017.

[14] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *CVPR Workshops*, 2018.

[15] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *CVPR*, 2018.

[16] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *CVPR*, 2019.

[17] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "Touchdown: Natural language navigation and spatial reasoning in visual street environments," in *CVPR*, 2019.

[18] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell, "Speaker-follower models for vision-and-language navigation," in *NeurIPS*, 2018.

[19] F. Zhu, Y. Zhu, X. Chang, and X. Liang, "Vision-language navigation with self-supervised auxiliary reasoning tasks," in *CVPR*, 2020.

[20] M. Tomasello and M. J. Farrar, "Joint attention and early language," *Child development*, pp. 1454–1463, 1986.

[21] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv:1712.05474*, 2017.

[22] M. Hasanuzzaman, V. Ampornaramveth, T. Zhang, M. Bhuiyan, Y. Shirai, and H. Ueno, "Real-time vision-based gesture recognition for human robot interaction," in *ROBIO*, 2004.

[23] K. Nickel and R. Stiefelhagen, "Visual recognition of pointing gestures for human–robot interaction," *Image and Vision Computing*, vol. 25, no. 12, pp. 1875–1884, 2007.

[24] B. S. Ertuğrul, C. Gurpinar, H. Kivrak, and H. Kose, "Gesture recognition for humanoid assisted interactive sign language tutoring," in *SIU*, 2013.

[25] C. Nuzzi, S. Pasinetti, M. Lancini, F. Docchio, and G. Sansoni, "Deep learning-based hand gesture recognition for collaborative robots," *IEEE Instrumentation & Measurement Magazine*, vol. 22, no. 2, pp. 44–51, 2019.

[26] J. Chang, J. Xiao, J. Chai, and Z. Zhou, "An improved faster r-cnn algorithm for gesture recognition in human-robot interaction," in *Chinese Automation Congress*, 2019.

[27] K. Jiang, S. Stacy, C. Wei, A. Chan, F. Rossano, Y. Zhu, and T. Gao, "Individual vs. joint perception: a pragmatic model of pointing as communicative smithian helping," in *CogSci*, 2021.

[28] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," in *ICLR*, 2019.

[29] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to parse natural language commands to a robot control system," in *Experimental Robotics*, 2013.

[30] D. Misra, A. Bennett, V. Blukis, E. Niklasson, M. Shatkhin, and Y. Artzi, "Mapping instructions to actions in 3d environments with visual goal prediction," *arXiv:1809.00786*, 2018.

[31] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *CVPR*, 2015.

[32] Y. Niu, H. Zhang, M. Zhang, J. Zhang, Z. Lu, and J.-R. Wen, "Recursive visual attention in visual dialog," in *CVPR*, 2019.

[33] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov, "Towards ai-complete question answering: A set of prerequisite toy tasks," *arXiv:1502.05698*, 2015.

[34] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge," *TPAMI*, vol. 39, no. 4, pp. 652–663, 2016.

[35] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi, "Ok-vqa: A visual question answering benchmark requiring external knowledge," in *CVPR*, 2019.

[36] A. Das, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Neural modular control for embodied question answering," *arXiv:1810.11181*, 2018.

[37] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, "Vision-and-dialog navigation," in *Conference on Robot Learning*, 2020.

[38] W. Hao, C. Li, X. Li, L. Carin, and J. Gao, "Towards learning a generic agent for vision-and-language navigation via pre-training," in *CVPR*, 2020.

[39] M. Shridhar, X. Yuan, M.-A. Côté, Y. Bisk, A. Trischler, and M. Hausknecht, "Alfworld: Aligning text and embodied environments for interactive learning," *arXiv preprint arXiv:2010.03768*, 2020.

[40] H. Yu, H. Zhang, and W. Xu, "Interactive grounded language acquisition and generalization in a 2d world," *arXiv:1802.01433*, 2018.

[41] M. Chevalier-Boisvert, D. Bahdanau, S. Lahlou, L. Willems, C. Saharia, T. H. Nguyen, and Y. Bengio, "Babyai: A platform to study the sample efficiency of grounded language learning," in *ICLR*, 2018.

[42] T. Cao, J. Wang, Y. Zhang, and S. Manivasagam, "Babyai++: Towards grounded-language learning beyond memorization," *arXiv:2004.07200*, 2020.

[43] K. M. Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. M. Czarnecki, M. Jaderberg, D. Teplyashin, *et al.*, "Grounded language learning in a simulated 3d world," *arXiv:1706.06551*, 2017.

[44] X. Xie, H. Liu, Z. Zhang, Y. Qiu, F. Gao, S. Qi, Y. Zhu, and S.-C. Zhu, "Vrgym: A virtual testbed for physical and interactive ai," in *Proceedings of the ACM Turing Celebration Conference*, 2019.

[45] C. Gan, Y. Zhang, J. Wu, B. Gong, and J. B. Tenenbaum, "Look, listen, and act: Towards audio-visual embodied navigation," in *ICRA*, IEEE, 2020.

[46] C. Chen, U. Jain, C. Schissler, S. V. A. Gari, Z. Al-Halah, V. K. Ithapu, P. Robinson, and K. Grauman, "Soundspaces: Audio-visual navigation in 3d environments," in *ECCV*, pp. 17–36, Springer, 2020.

[47] S. Brodeur, E. Perez, A. Anand, F. Golemo, L. Celotti, F. Strub, J. Rouat, H. Larochelle, and A. Courville, "Home: A household multimodal environment," *arXiv:1711.11017*, 2017.

[48] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian, "Building generalizable agents with a realistic and rich 3d environment," *arXiv:1801.02209*, 2018.

[49] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," in *CVPR*, pp. 10740–10749, 2020.

[50] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, "Virtualhome: Simulating household activities via programs," in *CVPR*, pp. 8494–8502, 2018.

[51] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, "Minos: Multimodal indoor simulator for navigation in complex environments," *arXiv:1712.03931*, 2017.

[52] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *CVPR*, 2018.

[53] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, *et al.*, "Habitat: A platform for embodied ai research," in *CVPR*, 2019.

[54] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv:1709.06158*, 2017.

[55] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaśkowski, "Vizdoom: A doom-based ai research platform for visual reinforcement learning," in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, 2016.

[56] C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik, *et al.*, "Deepmind lab," *arXiv:1612.03801*, 2016.

[57] C. Yan, D. Misra, A. Bennnett, A. Walsman, Y. Bisk, and Y. Artzi, "Chalet: Cornell house agent learning environment," *arXiv:1801.07357*, 2018.

[58] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *ICRA*, 2017.

[59] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. 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.

[60] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson, "Visual representations for semantic target driven navigation," in *ICRA*, 2019.

[61] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[62] M. A. Goodrich and A. C. Schultz, *Human-robot interaction: a survey*. Now Publishers Inc, 2008.

[63] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, *et al.*, "Guidelines for human-ai interaction," in *CHI*, 2019.

[64] S. Kiesler, A. Powers, S. R. Fussell, and C. Torrey, "Anthropomorphic interactions with a robot and robot–like agent," *Social Cognition*, vol. 26, no. 2, pp. 169–181, 2008.

[65] J. Y. Chai, Q. Gao, L. She, S. Yang, S. Saba-Sadiya, and G. Xu, "Language to action: Towards interactive task learning with physical agents.," in *IJCAI*, 2018.

[66] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi, "Iqa: Visual question answering in interactive environments," in *CVPR*, 2018.

[67] K. Nguyen and H. Daumé III, "Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning," *arXiv:1909.01871*, 2019.

[68] S. Chen, H. Ma, C. Yang, and M. Fu, "Hand gesture based robot control system using leap motion," in *ICIRA*, 2015.

[69] Q. Gao, J. Liu, Z. Ju, Y. Li, T. Zhang, and L. Zhang, "Static hand gesture recognition with parallel cnns for space human-robot interaction," in *ICIRA*, 2017.

[70] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Skeleton-based dynamic hand gesture recognition," in *CVPR*, 2016.

[71] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand pointnet: 3d hand pose estimation using point sets," in *CVPR*, 2018.

[72] S. Baek, K. I. Kim, and T.-K. Kim, "Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering," in *CVPR*, 2019.

[73] S. Yang, J. Liu, S. Lu, M. H. Er, and A. C. Kot, "Collaborative learning of gesture recognition and 3d hand pose estimation with multi-order feature analysis," in *ECCV*, 2020.

[74] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *CVPR*, 2018.

[75] L. Weihs, J. Salvador, K. Kotar, U. Jain, K.-H. Zeng, R. Mottaghi, and A. Kembhavi, "Allenact: A framework for embodied ai research," *arXiv:2008.12760*, 2020.

[76] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv:2006.13171*, 2020.

[77] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv:1807.06757*, 2018.

[78] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[79] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[80] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv:1406.1078*, 2014.