Towards Making Virtual Human-Robot Interaction a Reality

Padraig Higgins*, Gaoussou Youssouf Kebe*, Adam Berlier*,
Kasra Darvish, Don Engel, Francis Ferraro, Cynthia Matuszek*
phiggin1,gaoussou1,aberlie1,kasradarvish,donengel,ferraro,cmat@umbc.edu
University of Maryland, Baltimore County
Baltimore, Maryland

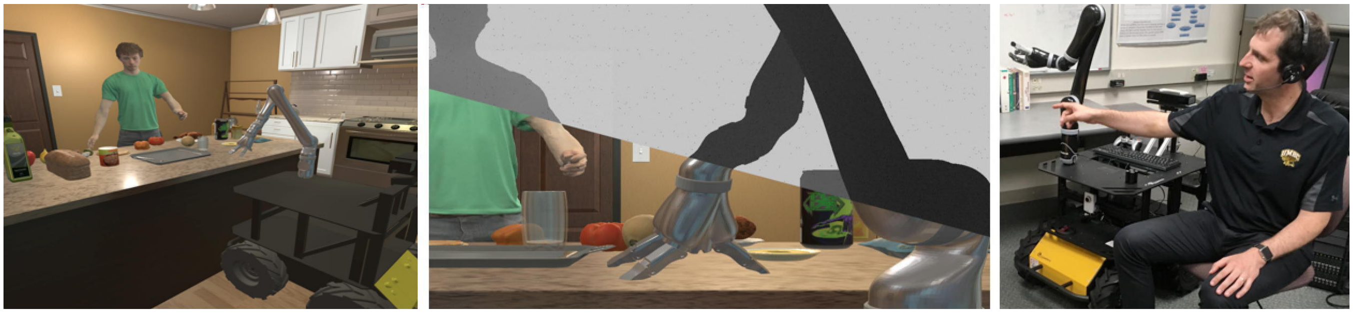

Figure 1: Our proposed Sim2Real human participant study. Left: a participant trains a robot on a grounded language task in virtual reality. Middle: the simulated robot percepts. Right: the learned model is tested on the physical robot.

ABSTRACT

For robots deployed in human-centric spaces, natural language promises an intuitive, natural interface. However, obtaining appropriate training data for grounded language in a variety of settings is a significant barrier. In this work, we describe using human-robot interactions in virtual reality to train a robot, combining fully simulated sensing and actuation with human interaction. We present the architecture of our simulator and our grounded language learning approach, then describe our intended initial experiments.

CCS CONCEPTS

• **Computer systems organization** → **Robotics**; • **Theory of computation** → *Semi-supervised learning*; • **Human-centered computing** → **Virtual reality**.

KEYWORDS

Human-Robot Interaction, Virtual Reality, Grounded Language Acquisition, Sim2Real

ACM Reference Format:
Padraig Higgins*, Gaoussou Youssouf Kebe*, Adam Berlier*,, Kasra Darvish, Don Engel, Francis Ferraro, Cynthia Matuszek. 2021. Towards Making Virtual Human-Robot Interaction a Reality. In *VAM-HRI submission 2021.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Robots deployed in dynamic, varied human settings will need to contend with a wide range of possible environments and tasks. One approach to addressing this is to allow end users to teach and instruct their robots using *grounded* language—natural language about the physical setting. In robotics, this has generally involved combining sensor data with human language [25, 26, 29, 35], and sometimes gesture and other modalities [23, 37], to create a joint model of what language refers to in the robot's frame of reference. However, this process requires extensive training data, covering a wide variety of settings, objects, tasks, and language. Relying on pre-trained models can reduce this training load, but not eliminate it, given the perceptual variations of different environments and the idiosyncrasy of human language. In addition, such data collection tends to suffer from a failure to include diverse populations, largely as a result of dependence on populations of convenience.

The ultimate goal of this work is to improve our ability to gather data in different settings and from different groups. We approach this by creating VR scenarios, in which a person can teach a robot about objects while simulated perceptual data is collected along with language and gesture. This learned model can then be brought to a physical robot, where training can be completed—the "Sim2Real" approach. In this work we describe the RIVR (Robot Interaction in Virtual Reality, or "river") simulator, which is designed to conduct human participant experiments in an environment that is both immersive for a person and technically correct for a robot. We

[*] These authors contributed equally to this research.

VAM-HRI, 2021

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

utilize the Unity game engine to build our simulated environments, and use ROS# [3] to link it with ROS [31] and Gazebo, allowing the same software and message-passing to be used on the virtual robot and its physical analog. We leverage our photogrammetry facility in order to collect realistic human models to use as avatars.

Simulation has been a valuable tool in robotic research [7, 11, 12], including in teaching robots about their environments using natural language [1, 8, 15, 21, 30]. However, these environments typically do not provide the embodied interaction between robot and human that HRI often requires. Similar to Bartneck et al. [2], we are leveraging the Unity game engine's powerful animation and interaction tools to facilitate the development of complex HRI studies. Virtual reality, meanwhile, allows for a user to be fully immersed in an environment and has shown promise when used as a tool to provide training demonstrations, for example in learning grasping policies [17, 32, 36].

Our work seeks to bring both tools together. The work most similar to ours is the SIGVerse project [16]; however, our work focuses on gathering language for grounded language learning in parallel with gathering aligned, high-quality simulated perceptual data, collected in a wide variety of human environments. RIVR's client/server architecture makes it possible for anyone who owns a commodity gaming headset to participate in data collection. In addition, we will be able to bring a VR headset and laptop to communities, rather than requiring participants to be able to visit a lab or deployment site.

2 APPROACH

In this section, we first describe the overall architecture of the simulator, including the Python API that we are developing to make it easier for other groups to adopt the simulator. We then describe our approach to learning a grounded language model using data collected from experiments run in RIVR. In the following section, we describe the first experiment for which we intend to use these platforms.

2.1 The RIVR Simulator

The overall architecture of the system described in this section is shown in Fig. 2.

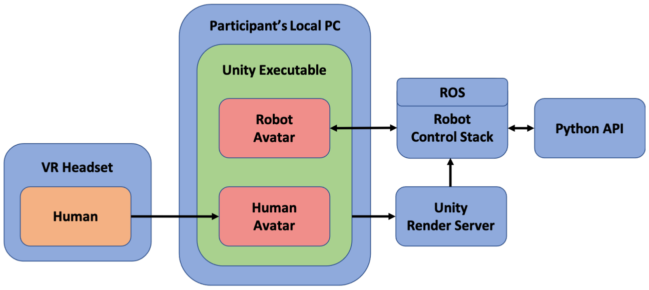

Figure 2: A high-level architecture of the RIVR system

Unity Environment and Render Server. The Unity simulation provides the virtual representation of an environment for the human and robot to interact in. Pre-built environments from the AI2Thor simulation environment serve as foundation for building experiment specific scenes [21]. There also exists a variety of pre-built assets available for Unity game development. These can be as simple as models of household objects to larger environmental scenes of landscapes. Leveraging these assets allows for the construction of environments that can be more varied than those that can be built in a laboratory setting. This foundation provides an interface for ablating and/or adding scenes and information available to the human or robot. These modifications can provide further insight into correlations in behavior and performance.

Compared to physical experiments, the virtual environment requires less cost and effort to modify experiments. Unity plugins for VR and ROS# immerse both the human and robot into the environment for representative perception and interaction with the environment and each other. The virtual reality client is the only part of the system that is run on a participant's local machine. It streams the text transcriptions, the position of the headset and controllers, and button inputs to the ROS node for the robot. Only the participants view is rendered and streamed to the participant's headset; while the Unity Render Sever models the robots more complex sensors on an adequately powered remote server. This helps lessen the computational requirements on users.

VR Interface. Human interactions are captured by SteamVR-supported[1] headsets. The human avatar is animated in accordance with the VR headset and controllers connected to the participant's local machine. Three-point body dynamics are derived from the localized headset and two handheld controllers. The three point dynamics are used to control gestures on the human avatar while the built-in microphone captures speech. The participant is able to perceive the environment, including the robot avatar. Without handheld controllers, the participant be unable to gesture, but still maintain capability for speech through the headset. Currently, a SteamVR-supported headset is required for human interaction in the simulation. The use of motion tracked controller and headset are used to capture the user gaze and gesture that can be animated onto the human avatar.

ROS Robot Control Stack. Robot interactions are modeled by the ROS robot control stack. A ROS server runs a Gazebo simulation for robot dynamics alongside ROS nodes for defining and controlling the robot. The Simulation Description Format is used to configure the environment scene information in Gazebo while the Unified Robot Description Format is used to represent the robot model [20]. Rosbridge is used by the control stack for the Unity client and Python API to connect to the simulator over the internet, as well as record all the sensor data from the user, robot, and any other sensors present in the scene. The ability to capture complete scene dynamics and replay these interactions in simulation allows experiments to introduce both unique humans and novel robots into previously encountered scenarios for directly comparing evaluations of their behavior.

Python API. The Python API enables training and inference algorithms to be executed in the simulation feedback loop. By defining a

[1] SteamVR is a virtual reality extension to the Steam gaming platform by Valve. It provides the headset access to the OpenVR plugin used by our unity simulation.

"User" class that contains the training/inference algorithm, the simulation will execute the algorithms at the beginning of the robots decision. The API receives observations as input and provides decisions as output. Observations include all sensor measurements available to the robot (e.g. cameras, microphones, etc.). Decisions consist of a list of object locations to interact with and an associated task assigned to each object. Tasks include object collection, moving object location or orientation, etc. Tasks are currently executed by task specific ROS controllers built for the robot. For a detailed experiment setup, see section 3.

Current system developments support a turn-based interaction between a single human and a single robot. Guards are used to transition turns between human and robot. As shown in Fig. 3, the robot will continuously act in the environment until it has completed task execution. To prevent challenges with end of utterance detection, recording of human speech begins with the press of a button on the VR controller and ends with the release of that button. The speech recording and associated test transcripts can also be used as observations by the Python API and robot. Near-term efforts are planned to modify the system to support asynchronous interactions between human and robot. This will significantly expand the types of interactions that can be supported.

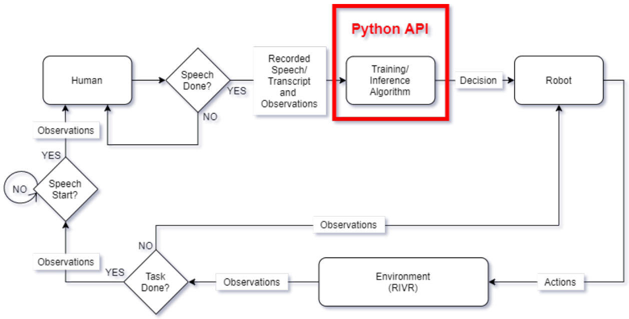

Figure 3: System data flow for an HRI language-collection experiment in RIVR

2.2 Learning a Grounded Language Model

We consider two different approaches to the language grounding problem, but essentially any other model can be deployed as long as it takes some observations including RGB-Depth images, speech, and transcriptions as input, and returns a decision or an action back. Action can be anything from picking an object to generating a sentence as a response to the user.

The first approach follows the conventional paradigm. A natural language description or a spoken natural language utterance is transcribed using Google's Speech to Text API and the language grounding model predicts which objects correspond to the textual input. While this approach has shown success [24–26, 29, 35], speech-to-text models have known biases that make them more error-prone for specific sub-populations [4, 10, 13, 19, 34]. Since the grounding model only has access to the output of the speech-to-text model, it effectively shares the same bias. For this reason, we consider a second less-conventional approach that skips the transcription step and relies on the raw speech data itself.

We use the language grounding model described in Nguyen et al. [25], although there is no speech involved in their work. The language grounding problem is treated as one of manifold alignment where the goal is to project language and vision representations into a shared manifold where instances of either type that are semantically similar are pushed to be close to each other. This is achieved using triplet loss, a supervised loss that minimizes the distance between an arbitrary anchor point a and a positive point p while maximizing the distance between anchor a and a negative point n. The loss is computed in the following way:

$$L(a, p, n) = \max\{d((f(a) - f(p)) - d(f(a) - f(n))) + \alpha, 0\}, \quad (1)$$

where f is an encoding function, d is cosine distance, and α is an arbitrarily chosen margin that we set to 0.4. We select a, p and n from either domains. Given a vision or language anchor point a, p and n are chosen such that a and p represent the same object and n represents a different object. For example if a is the RGB-D data of a mug, p could be another RGB-D data of the same mug or a language instance that describes the mug, and n could be the RGB-D data of a different object (e.g., bottle) or a language instance that does not describe the mug.

The raw inputs are featurized using off-the-shelf pretrained models. We featurize the RGB-D visual input using a ResNet152 [14] pre-trained on ImageNet. In the transcription-based approach, the textual input is featurized using a pre-trained BERT [9] model where the text embedding is obtained by averaging the concatenation of the last 4 BERT layers for every word in the description. Similarly, in the raw speech approach, the features are extracted using DeCoAR [22], a pre-trained self-supervised speech representation model. We use multi-layer perceptrons to project both the language and vision features into the shared latent space.

We plan to use the RIVR setup to both train and evaluate our language grounding models, as described below.

2.2.1 *Training.*

Our triplet-loss based grounding model needs positive and negative examples to build a meaningful latent space. While the language and vision data pairs obtained from a human describing a given object are natural (anchor, positive) pairs, negative examples are harder to obtain. Previous work has approached negative sampling by randomly picking an instance from different object classes when object labels are available or by picking a negative description that's semantically distant from the object's description in the absence of labels [25, 27]. While both approaches can achieve good results, we intend to test the hypothesis that querying a human for negative examples would result in more reliable sampling.

2.2.2 *Inference.*

Segmented RGB-D data and centroid location for every object perceived by the robot, the recorded speech and the corresponding transcription are obtained from the simulation through the Python API. For each object, the RGB-D data along with the provided language description (spoken or transcribed) are given to the language grounding models described in Section 2.2. For each model, we will use a threshold t determined through validation after training such that any object projected to the shared latent space within a radius t of the language description's projection is predicted to be relevant. The centroid locations of every relevant object are then passed through the Python API to the robot control stack.

3 PROPOSED EXPERIMENT

The overarching goal of this project is to understand how HRI-studies that involve language can be conducted in simulation, using robot sensors and immersive interactions. The goal of our first proposed experiment is to collect language in the setting described above and use it to train and test a grounded language model in simulation; once this is successful, we will transfer the learned model to physical robots. We focus on a language-based object retrieving task in which the robot learns a language model for objects from a human interlocutor, then guides the robot through the task of packing a lunch into a basket. Our initial experiment will take place in a scenario drawn from the AI2Thor [21] project, specifically a kitchen. In this scenario, a variety of food is placed on the kitchen counter (see Fig. 1). To populate this scene, we are taking advantage of the easy access to a widely diverse group of objects provided by Unity's broad asset base.

Experimental process. During the first phase (training), a participant will be asked to describe the objects on the table. Their verbal responses and head pose will be captured by a VR headset. Gestures, if any, will be captured by the associated handheld controllers. During this, a continuous stream of simulated sensor data will be collected and stored in a ros bag file. During this interaction, the robot will be able to ask follow-up questions in order to improve its understanding. For example, a question like "Is there another object that corresponds to this description?" would provide a positive visual example, while "Is this object also an apple?" would help reduce uncertainty for ambiguous objects. We intend to use active learning to determine what questions to ask, drawing on existing work on active learning for robots [6], especially in language [5, 28].

Our experiment is a between-subject study, where the first group of participants provides descriptions, while the second and third groups try to instruct a robot to perform actions based on what it has learned. During the second phase (testing), participants will be asked to instruct the robot through the process of packing a lunch, using the same interface. Our current expectation is that they will provide descriptions of the food items that should be packed into the lunch basket and the robot will attempt to follow the instructions, presumably by picking up all relevant objects and placing them in the basket. However, the goals of this initial experiment include understanding how these expectations may be incorrect.

During the final phase (transfer), a third group of participants will engage in the same testing task, but in the presence of a physical robot with real sensors and real food. Our expectation is that we will need to acquire additional training data in this real-world setting, but hopefully much less than would be required without the in-simulation data that has been collected. This stage of the experiment is intended to explore the efficiency of sim2real transfer and what changes need to be made in RIVR to improve that transfer.

Metrics. After all interactions, participants will be asked to provide feedback. We will ask them about the experience, how they felt about interacting with the robot, whether they found the training process frustrating, and so on. However, because the primary goal of this work is to establish the effectiveness of robot learning in simulation and acting in reality, we will focus heavily on questions about whether the robot packed the basket as expected and whether the questions asked by the robot were sensible. The robot's performance will be quantitatively evaluated as a classification task of its correct identification of objects to which the person refers. We will also consider metrics that differentiate between the virtual and real robots' performances, including comparing the amount of training data required in a sim2real versus learning-in-reality setting.

Recruitment. As a final note, data collected for machine learning tasks is often drawn from a very limited set of people, leading to significant problems in fairness, accountability, transparency, and ethics. [18] While improving diversity in data is only one element of the general problems with diversity in AI, it is an important one. While our initial recruitment will focus on people who already own commodity virtual reality hardware, once the current pandemic is no longer a factor, we intend to aggressively recruit participants from a variety of groups. It is our hope that the accessibility and portability of RIVR will make it much easier to reach a broader set of experiment participants.

4 FUTURE WORK

Once we obtain reliable results in simulation, our most immediate goal will be to transfer the language grounding models learned in simulation to a real robotic system. Bridging the gap between simulation and reality is a challenging research problem but we expect that the realistic nature of our simulated interactions will enable a smoother transition.

We intend to replace the current turn-based interaction with an interactive dialogue system. Once the simulation supports asynchronous interactions, we plan on supporting interactions between multiple robots and multiple humans. We also plan on implementing better modeling of RGB-D sensors. Currently, Gaussian noise is added to the depth images. We plan on integrating the noise models proposed by Sweeney et al. [33] to simulate pixel dropout in depth images. We also plan on implementing additional robotic sensors such as distance, thermal, laser or LIDAR sensors that would create interesting data collection and learning opportunities. These developments would enable RIVR to support task-based domain adaptation and transferring learned models of grounded language between robots with heterogeneous sensor platforms.

For users not familiar with Unity and/or RIVR, simulation developments intend to work towards a Unity API for modifying scenes and robots. A Python API provides an interface for specifying how objects will be modified in the scene (e.g. added, ablated, change size and/or color, etc.) without the need to understand Unity development. Using local 3D-model files can provide users the ability to introduce unique objects in the scene. In addition, we plan to implement support for users to provide URDF files through the Unity API to construct novel robots and sensors in the scene. The Unity API is intended to help simplify the process of designing new experiments and increase usability across the robotics community.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. 1428204, 1531491, 1637614, 1657469, 1637937, 1940931, and 2024878.

REFERENCES

[1] Peter Anderson, Ayush Shrivastava, Joanne Truong, Arjun Majumdar, Devi Parikh, Dhruv Batra, and Stefan Lee. 2020. Sim-to-Real Transfer for Vision-and-Language Navigation. 2011.03807.

[2] C. Bartneck, M. Soucy, K. Fleuret, and E. B. Sandoval. 2015. The robot engine — Making the unity 3D game engine work for HRI. In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, Kobe, Japan, 431–437. https://doi.org/10.1109/ROMAN.2015.7333561

[3] Martin Bischoff. 2019. ROS#. https://github.com/siemens/ros-sharp/releases/tag/v1.6

[4] Su Lin Blodgett and Brendan O'Connor. 2017. Racial Disparity in Natural Language Processing: A Case Study of Social Media African-American English. arXiv:1707.00061 http://arxiv.org/abs/1707.00061.

[5] Kalesha Bullard, Yannick Schroecker, and Sonia Chernova. 2019. Active Learning within Constrained Environments through Imitation of an Expert Questioner. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, Macao, China, 2045–2052. https://doi.org/10.24963/ijcai.2019/283

[6] M. Cakmak and A. L. Thomaz. 2012. Designing robot learners that ask good questions. In *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, Boston, Massachusetts, USA, 17–24. https://doi.org/10.1145/2157689.2157693

[7] S. Chernova, N. DePalma, E. Morant, and C. Breazeal. 2011. *Crowdsourcing human-robot interaction: Application from virtual to physical worlds.* In RO-MAN, 2011 IEEE, pages 21-26. IEEE.

[8] Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, L. Willems, Chitwan Saharia, T. Nguyen, and Yoshua Bengio. 2019. BabyAI: A Platform to Study the Sample Efficiency of Grounded Language Learning. ICLR.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[10] Raymond Fok, Harmanpreet Kaur, Skanda Palani, Martez E Mott, and Walter S Lasecki. 2018. Towards more robust speech interactions for deaf and hard of hearing users. , 57–67 pages.

[11] Maxwell Forbes, Michael Chung, Maya Cakmak, and Rajesh Rao. 2014. Robot programming by demonstration with crowdsourced action fixes. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 2. AAAI, Pittsburgh, Pennsylvania, 1.

[12] M. Forbes, R. P. Rao, L. Zettlemoyer, and M. Cakmak. 2015. *Robot programming by demonstration with situated spatial language understanding.* In Robotics and Automation (ICRA), 2015 IEEE International Conference on, pages 2014-2020. IEEE.

[13] Mahault Garnerin, Solange Rossato, and Laurent Besacier. 2019. Gender representation in French Broadcast Corpora and its impact on ASR performance. In *Proceedings of the 1st International Workshop on AI for Smart TV Content Production, Access and Delivery*. ACM, Nice, France, 3–9.

[14] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Caesars Palace, Las Vegas, NV, 770–778. https://doi.org/10.1109/CVPR.2016.90

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

[16] Tetsunari Inamura and Yoshiaki Mizuchi. 2020. SIGVerse: A cloud-based VR platform for research on social and embodied human-robot interaction. arXiv:2005.00825 [cs.RO] arXiv.

[17] Astrid Jackson, Brandon D. Northcutt, and Gita Sukthankar. 2018. The Benefits of Teaching Robots Using VR Demonstrations. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* (Chicago, IL, USA) *(HRI '18)*. Association for Computing Machinery, New York, NY, USA, 129–130. https://doi.org/10.1145/3173386.3176980

[18] Eun Seo Jo and Timnit Gebru. 2020. Lessons from Archives: Strategies for Collecting Sociocultural Data in Machine Learning. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Barcelona, Spain) *(FAT* '20)*. Association for Computing Machinery, New York, NY, USA, 306–316. https://doi.org/10.1145/3351095.3372829

[19] Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R. Rickford, Dan Jurafsky, and Sharad Goel. 2020. Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences* 117, 14 (2020), 7684–7689. https://doi.org/10.1073/pnas.1915768117 arXiv:https://www.pnas.org/content/117/14/7684.full.pdf

[20] Nate Koenig. 2012. SDFormat Specification. http://sdformat.org/spec

[21] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. 2017. AI2-THOR: An Interactive 3D Environment for Visual AI.

[22] Shaoshi Ling, Yuzong Liu, Julian Salazar, and Katrin Kirchhoff. 2020. Deep Contextualized Acoustic Representations For Semi-Supervised Speech Recognition. In *ICASSP*. IEEE, Barcelona, Spain, 6429–6433.

[23] Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. 2014. Learning from Unscripted Deictic Gesture and Language for Human-Robot Interactions.

[24] Cynthia Matuszek, Dieter Fox, and Karl Koscher. 2010. Following Directions Using Statistical Machine Translation. In *HRI*. IEEE, Lausanne, Switzerland, 251–258.

[25] Andre T. Nguyen, Luke E. Richards, Gaoussou Youssouf Kebe, Edward Raff, Kasra Darvish, Frank Ferraro, and Cynthia Matuszek. 2020. Practical Cross-modal Manifold Alignment for Grounded Language. arXiv:2009.05147 [cs.CV]

[26] Thao Nguyen, Nakul Gopalan, Roma Patel, Matthew Corsaro, Ellie Pavlick, and Stefanie Tellex. 2020. Robot Object Retrieval with Contextual Natural Language Queries. In *Proceedings of Robotics: Science and Systems*. RSS, Corvalis, Oregon, USA. https://doi.org/10.15607/RSS.2020.XVI.080

[27] Nisha Pillai, Francis Ferraro, and Cynthia Matuszek. 2018. Optimal Semantic Distance for Negative Example Selection in Grounded Language Acquisition. Robotics: Science and Systems Workshop on Models and Representations for Natural Human-Robot Communication.

[28] Nisha Pillai, Edward Raff, Francis Ferraro, and Cynthia Matuszek. 2021. Sampling Approach Matters: Active Learning for Robotic Language Acquisition. Proc. of the IEEE International Conference on Big Data (BigData), machine learning session.

[29] L. E. Richards, K. Darvish, and C. Matuszek. 2020. Learning Object Attributes with Category-Free Grounded Language from Deep Featurization. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Las Vegas, USA, 8400–8407. https://doi.org/10.1109/IROS45743.2020.9340824

[30] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, New York, USA, 10737–10746. https://arxiv.org/abs/1912.01734

[31] Stanford Artificial Intelligence Laboratory et al. 2018. *Robotic Operating System*. ROS. https://www.ros.org

[32] Francesca Stramandinoli, Kin Gwn Lore, Jeffrey R Peters, Paul C O'Neill, Binu M Nair, Richa Varma, Julian C Ryde, Jay T Miller, and Kishore K Reddy. 2018. Robot Learning from Human Demonstration in Virtual Reality. Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI).

[33] C. Sweeney, G. Izatt, and R. Tedrake. 2019. A Supervised Approach to Predicting Noise in Depth Images. In *2019 International Conference on Robotics and Automation (ICRA)*. In Robotics and Automation (ICRA), Montreal, QC, Canada, 796–802. https://doi.org/10.1109/ICRA.2019.8793820

[34] Rachael Tatman. 2017. Gender and Dialect Bias in YouTube's Automatic Captions. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*. Association for Computational Linguistics, Valencia, Spain, 53–59. https://doi.org/10.18653/v1/W17-1606

[35] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI'11)*. AAAI Press, San Francisco, California, 1507–1514.

[36] David Whitney, Eric Rosen, and Stefanie Tellex. 2018. Learning from Crowdsourced Virtual Reality Demonstrations. Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI).

[37] Tom Williams. 2018. A framework for robot-generated mixed-reality deixis. Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI).