Head Pose for Object Deixis in VR-Based Human-Robot Interaction

Padraig Higgins,[1] Ryan Barron,[1] Cynthia Matuszek[1]

Abstract— **Modern robotics heavily relies on machine learning and has a growing need for training data. Advances and commercialization of virtual reality (VR) present an opportunity to use VR as a tool to gather such data for human-robot interactions. We present the Robot Interaction in VR simulator, which allows human participants to interact with simulated robots and environments in real-time. We are particularly interested in spoken interactions between the human and robot, which can be combined with the robot's sensory data for language grounding. To demonstrate the utility of the simulator, we describe a study which investigates whether a user's head pose can serve as a proxy for gaze in a VR object selection task. Participants were asked to describe a series of known objects, providing approximate labels for the focus of attention. We demonstrate that using a concept of gaze derived from head pose can be used to effectively narrow the set of objects that are the target of participants' attention and linguistic descriptions.**

I. INTRODUCTION

Preparing robots for human environments requires training on realistic data. In this paper, we present RIVR (Robot Interaction in Virtual Reality), a simulator that allows acquiring a training corpus of human-robot interactions. Our system allows for a robot and its sensors to be represented in a scene where they interact with a human avatar animated by the actions of a human using a commodity virtual reality system. Our goal is to enable simulation which includes real-time human-robot interaction and captures realistic output from the virtual robot's simulated sensors of the VR environment. The human participant is represented using an avatar and rigged to the VR controllers and headset using inverse kinematics. This is a new direction relative to prior work, which does not include a human in the scene [1].

Machine learning is commonly used as a method of training robots, but it requires large amounts of data to properly weight neural connections between the machine learning model's layers [2]. Our previous work has focused on teaching robots about their environments by having humans describe objects with natural language [3]. Collecting human-robot interaction data to train these models can take significant time. One method for optimizing data collection in robotic learning is to perform learning in simulation, which increases the radius of possible participants, minimizes travel, and reduces machine maintenance. Models trained in simulation can then be transferred to physical robots and further trained. While this simulation-to-reality, or *sim2real*, pipeline has been widely explored, the majority of sim2real work in human-robot interaction has focused on the human side, e.g., training people to use robot systems [4], [5] or exploring human comfort levels with different scenarios [6]–[8]. In this work we focus on robot learning from interactions with people in a simulated environment.

Meanwhile, in the context of robots learning from human language, gathering unconstrained human descriptions of objects in the environment requires significant effort to properly align the descriptions to the visual percepts of the specific objects being described. To allow for more self-supervised interactions, we investigate whether we can use attributes of the interaction to label the data points as they are collected. Particularly, we evaluate the RIVR HRI simulation by investigating whether a VR participant's head pose can serve as a proxy for gaze to label objects as they are described. We consider head pose as it is commonly available in all commodity VR hardware, as opposed to using eye tracking, which only some platforms support.

In our experiments, we collect data from participants who are describing objects and tasks in a virtual environment (fig. 2), with the goal of using head pose to determine what objects are being described at each time step. Head pose was determined from the position and orientation of the headset, and from that the object that most closely matched the direction of gaze is calculated. We evaluate two different approaches: a baseline, where the target object is selected to minimize the angular distance from an object position to the participant's gaze direction; and the approach introduced in BayesGaze [9], which utilizes Bayesian inference.

The contributions of our work are threefold. First, we present a system that integrates a VR environment with a realistic, controllable virtual robot platform. This system was developed with language-based human-robot learning in mind, with the specific goal of gathering robotic sensor data in a setting with sufficient realism and immersion to allow for natural human speech and behavior. Second, it brings virtual reality into an application area where it may potentially support a new class of research studies in human-robot interaction. This includes offering a platform that supports fast, safe, and inexpensive research in robotics, a field that traditionally has a significant barrier to entry. Third, we present an evaluation of the use of head pose as a proxy for gaze to allow for data labeling.

II. RELATED WORK

Robots deployed in dynamic human settings will need to contend with a wide range of environments and tasks. One approach to addressing this is to allow end users to

*This material is based in part upon work supported by the National Science Foundation under grant numbers 1940931, 2024878, and 2145642.

[1]Authors are affiliated with the University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD, 21250, USA. `(phiggins1, ryanb4, cmat@umbc.edu)`

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 [10]–[14], and sometimes gesture and other modalities [15]–[17], to create a joint model of what language refers to in the robot's frame of reference. However, this process requires extensive training data. Relying on pretrained models can reduce this training load but not eliminate it, given the perceptual variations of different environments and the idiosyncrasy of language.

The 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 transfer" approach. We utilize the Unity game engine to build our simulated environments and use ROS# [18] to link it with ROS [19] and Gazebo, allowing the same software and message-passing to be used on the virtual robot and its physical analog.

Simulation has been a valuable tool in robotic research [20]–[22], including in teaching robots about their environments using natural language [23]–[27]. However, these environments typically do not provide the embodied interaction between robot and human that HRI often requires. Similar to [6], [28], [29], 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 [30]–[32]. Human communication consists of more than just spoken language, and includes modalities such as gaze and gesture, which provide useful information for grounding [33]. Motion tracked VR headsets and controllers allow RIVR to capture of these modalities.

Robots that have a method to express gaze are perceived more favorably and perform better during interactions with humans [34]–[36]. It has also been shown to be a useful tool to measure a person's engagement [37] during an interaction and as a measure of the person's perception of the robot [38]. Gaze has been used as a tool to improve robotic manipulation and handoff tasks, where gaze provides insight to the human participant's intent [39], [40]. It has also been used to establish and maintain a common ground during interactions [41], [42]. The performance of eye tracking in VR has been compared to eye tracking in the real world under ideal circumstances [43]. The accuracy did not differ when gazing as static targets, and only showed small differences at targets at varied distances, but did show larger differences when tracking moving targets, and showed that the precision in VR was much worse when focusing on static targets. This work only investigated the performance of eye tracking with the head in a fixed position.

Work has shown that head pose can be used as a method of control for user interfaces in both virtual [44] and augmented [45] reality. There has been work that has shown that head pose by itself cannot replace eye gaze [46], [47], since a person generally will not move their head to focus on objects that are close together, instead relying on eye movements [48]. However, as not all VR headsets support eye tracking, and as gaze tracking in human-robot interactions is not always viable, we hope to compare how well head pose works as a proxy for gaze in VR on the specific task of object selection when a person is teaching a robot. The use of bidirectional gaze in interactions with a virtual agent can improve task performance, and it has been shown that using head pose as a proxy for gaze in a system utilizing bidirectional gaze performs similarly to using full eye tracking [49].

III. APPROACH

To enable the remote collection of data, our simulator design uses a distributed approach. The researcher, test participant, and control server are all at different locations. This makes the system particularly sensitive to home internet connections of the simulation users. High latency, low bandwidth, and random packet loss are more likely than when similar experiments are performed in a lab. To mitigate this, the simulation is split between the client and a render server, lowering the bandwidth and processing power requirements for the client by offloading them to the render server, which can be run with a reliable high-speed connection to a server running ROS. All that is required is a VR headset and a laptop that can run the simulation.

As we hope to be able to use RIVR to collect the large amount of data required to train grounded language models, we seek to minimize the amount of annotation and data labeling that is required, matching descriptions to objects or places. Head pose is a potential tool that may be able to provide these annotations during the data collection process.

A. System Components

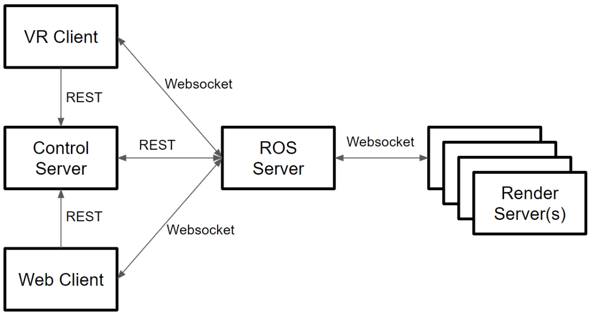


Fig. 1: A diagram showing the major software components of the system and their connections. The control server manages the running simulation, and hosts web-based and a VR simulator client; the ROS server launches a rosbridge client and manages interaction with the simulated robot and sensors; and the render servers model the simulated sensor data from which the robot learns.

The system is implemented as a set of independent components that connect to each other over the Internet as shown in Fig. 1. This design allows researchers and test participants to run portions of the simulation locally, thereby avoiding round-trip latency affecting their local display. This trade-off requires greater network bandwidth between the participant and the researcher, but significantly reduces the computational requirements for the participant's local system.

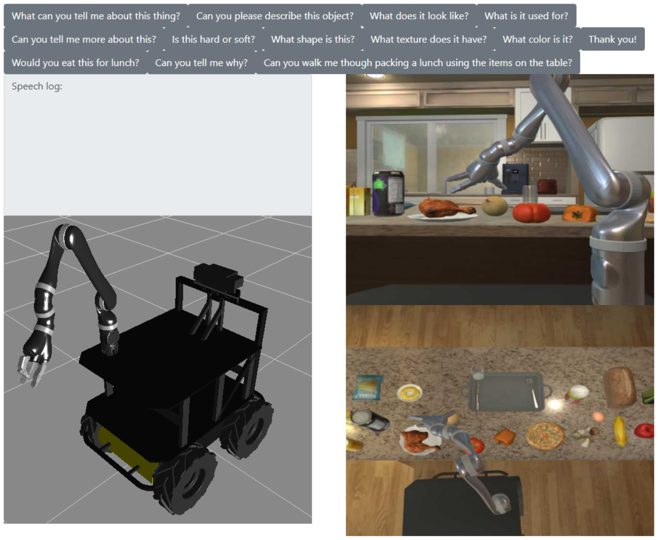

What can you tell me about this thing?
Can you please describe this object?
What does it look like?
What is it used for?
Can you tell me more about this?
Is this hard or soft?
What shape is this?
What texture does it have?
What color is it?
Thank you!
Would you eat this for lunch?
Can you tell me why?
Can you walk me though packing a lunch using the items on the table?
Speech log:

Fig. 2: A screenshot of the Wizard-of-Oz interface.

1) Control Server: The Control Server is used by the rest of the system components to track current simulation status, as well starting and stopping the server-side processes needed to run a simulation. The control server has a web front-end that allows customization of simulation runs. It also hosts a second web application that implements a web-based simulator client that supports robot teleoperation (see Fig. 2 for an example). Scripted tasks can be triggered through the use of buttons and keyboard shortcuts in the web interface, and data can be streamed live from the participant's virtual reality client to the web client.

2) ROS Server: ROS is installed on a server along with a control client. The control client loads simulation parameters from the control server and acts as a launcher for ROS. The ROS server launches an instance of rosbridge to allow clients to connect to the simulator over the network, and records simulation results to a rosbag file. When multiple simultaneous simulations are run, each simulation requires its own ROS server instance, with each instance running a separate websocket on a different port.

3) Virtual Reality Client: The virtual reality client runs on the participant's system. It renders the virtual reality environment to the participant's headset and captures the microphone audio from the headset. In order to minimize latency when objects in the scene are manipulated by the test participant, the simulator is also responsible for calculating the physics of loose objects in the scene, whether manipulated by the robot or the test participant. By calculating the motion of these objects on the participant's computer, we avoid an unnecessary round-trip over the network and the motion is much smoother. The state of the robot is streamed into this client simulator over the rosbridge websocket and visualized using a model of the robot imported into the scene using the ROS# Unified Robot Description Format (URDF) importer tool. The scene state including the pose of all the objects, the robot and user are all sent over the same websocket to the render server described in the next section.

4) Render Server: Certain sensors may be better modelled by the Unity game engine than by the Gazebo simulator, such as depth cameras. Unity also provides high quality shaders that are better able to model real-world materials than are readily available in Gazebo. However, when the simulated robot has a large number of sensors to simulate, the computational expense may become too great for the client's system to run while rendering to VR. Thus, we support using multiple instances of the Unity simulator as remote render servers, each responsible for a subset of the sensors in the scene, freeing the client instance to only render frames to the headset. The render servers generate camera views and depth sensor results that are transmitted over the ROS websocket in real-time to the other ROS nodes.

B. *Simulation Components*

The Unity simulation has two main components: (1) the environment, in which the human user and robot interact; and (2) the robot's exteroceptive sensors.

1) Environments: One of the key values of using Unity as the rendering engine for virtual reality (as opposed to Gazebo or other robotics-focused tools) is the ability to create and modify scenes easily. In the process of developing the simulator and this initial user study, we developed three different environments. Some assets were acquired from the Unity asset store, while others were generated from URDF files to model the robot. As shown in the pilot study presented here, we are able to load arbitrary Unity scenes, such as the kitchen scene from the AI2Thor project [23].

2) Human Avatars: One of the goals of this work was to have a model of one or more human participants in the scene with the simulated robot. There are a number of common sensors that robots may use to interact with a human: microphones may capture speech, cameras may capture gesture information, and depth sensors may capture body pose. We designed a system to record human performance and generate sample sensor inputs based on this recording without committing to a particular sensor selection or arrangement at the time of recording. While our current approach to motion capture is limited to the tracked points available to an off-the-shelf commercial headset, we are able to run a number of experiments with the current design.

We take advantage of MakeHuman [50], a freely available software package that can be used to build a diverse set of fully rigged human avatars that can be imported into Unity for use in the simulator. This avatar is animated from the poses of the headset and controllers using the Final IK Unity package [51] to give the robot a realistic view of the human.

3) Sensors: By using the Unity game engine and Gazebo in parallel, we have the choice to implement each sensor in either engine. We have chosen to implement sensors that interact with the world using Gazebo, and sensors that interact with the test participant using Unity. The sensors include an RGB+D camera capturing visual information of objects and human gestures in the scene, a microphone for capturing raw speech audio, and various robotic platform sensors. Our sample experiment uses a Husky robotics platform with a custom structure supporting a Kinova Jaco arm. The joint encoders in both the Husky robot's wheels and the joints of the Jaco arm are modelled using manufacturer-provided models in Gazebo. The Husky's Kinect RGB+D camera is modeled in Unity. As sensor outputs from both systems are made available over ROS topics, the robot's actions are fully decoupled from the simulator and able to support either source of data. This allows the same software used to control a physical robot to run the simulated hardware.

IV. Gaze Experiments

In order to evaluate the design and capability of our system, we investigated the use of head pose as a source of labels for linking the natural language descriptions of objects to the robots' visual perception of them.

A. Data Collection

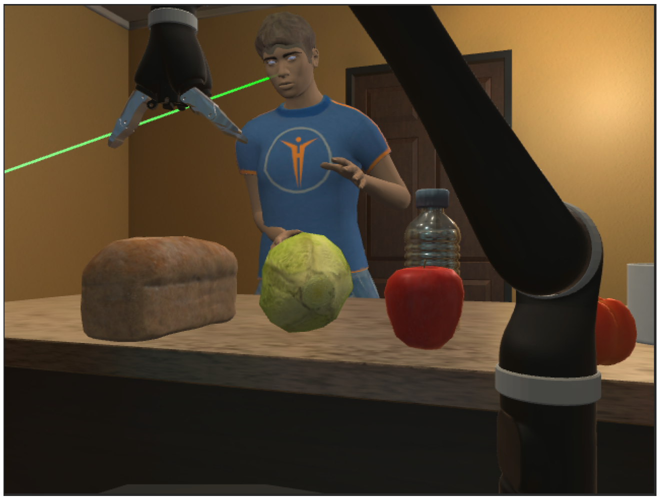


Fig. 3: The robot's perspective when prompting the user for a description of the bread. The gaze direction (not visible to participants) is overlaid on the image from the virtual reality headset's tracking to demonstrate the gaze-to-object proximity from the robot's sensors.

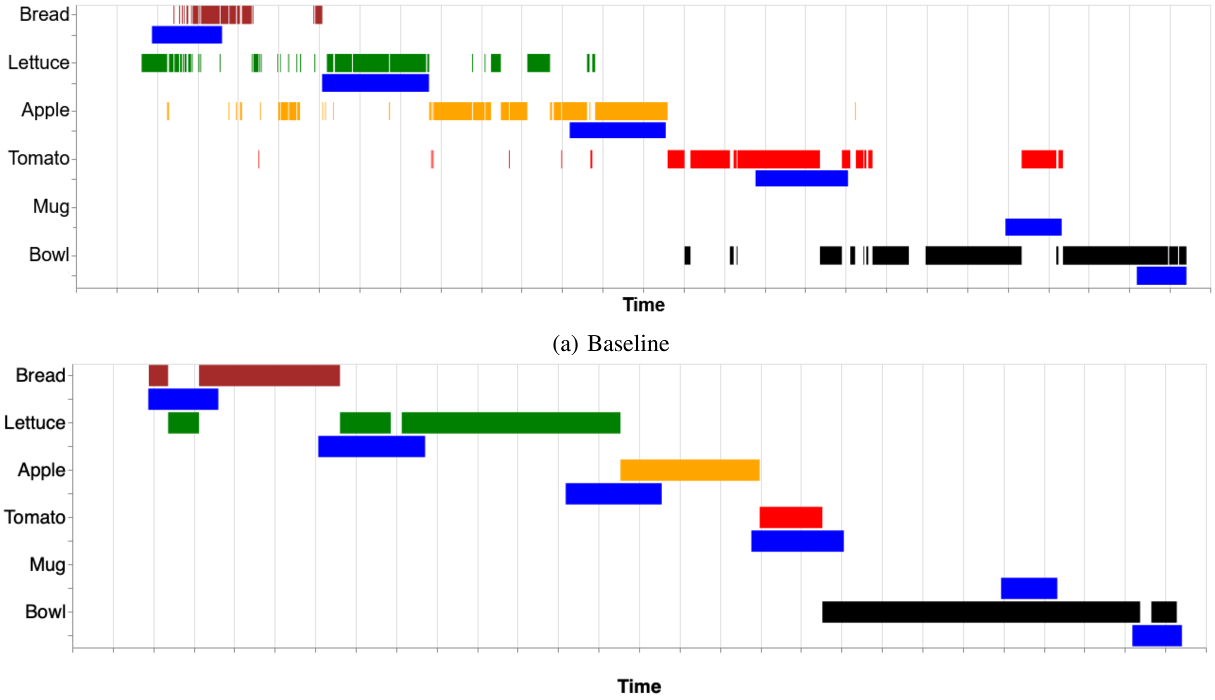
Participants were brought into a lab, and the simulation was explained. Users were told how the robot would prompt a response by asking for a description of objects. They were instructed to respond as though they were describing objects to someone that had never seen them before. Each participant went through the interaction once and was asked to describe the same six objects, out of a total of ten objects located on a kitchen island. All the objects were in the same locations for each participant. People described objects in a variety of ways, including physical attributes, ideal usages, and origins. There were fifteen participants in the study between the ages of 21 and 36, with a mean age of 25. Nine were male, six were female and one did not respond. Eight identified as Asian and seven as white. After hearing the instructions with an opportunity for questions, participants were given the virtual reality headset and taught how to adjust it.

After a familiarization period with the headset, the user was given two controllers. The right controller's trigger was used to signal the start and stop a data collection instance. The data collected on head pose was recorded by Unity and transformed into the robot's frame of reference in ROS. The objects described by the user were captured as point cloud data from a simulated Kinect. The point clouds were segmented and clustered in order to detect where objects on the kitchen countertop were positioned. Following the final description, the users were thanked by the robot and then filled out a post-study survey. Anecdotally, users felt that the simulator provided a fairly realistic environment, but that some objects were more realistically rendered than others; occasional difficulties with the VR display (e.g., blurry vision) were also reported.

B. Gaze Calculation

During experiments the entire interaction was captured in ROS, including the audio from the user, simulated RGB and depth from the robot's perspective, point clouds, the position and orientation of all the interactive objects in the scene, and all the ROS transform messages. The point clouds were segmented to get point clouds for each object on the table that the robot can see. The position and orientation of all interactive objects and headset were captured in the Unity coordinate system. The origin of the odometry frame of the robot in ROS matches the origin of the global coordinate system in Unity, but Unity uses the z axis as forward/backward, x as left/right and y as up/down. A ROS message was published that contains the raw audio that was recorded while the user is describing objects. In a few cases, the participants did not realize they should provide descriptions for the object the robot was indicating; in these cases, the audio recording was used to manually annotate which object they were describing.

1) Baseline: For a baseline we assumed that the target of the gaze was the object that had the lowest angular distance between vectors from the head position along the gaze direction and the vector from the head to the object. Once the head pose was in the same frame of reference as the point clouds, the position and orientation of the headset was used to determine the gaze direction by using a point one meter in front of the headset position in the direction it is facing. The vector defined by these two points is assumed to be the gaze vector. Each point in the point cloud was assigned a label by checking for the closest Unity object. The cosine similarity between the gaze vector and the vector between the head and the object was computed. Figure 5 shows this distance for each object on the countertop. The



(a) Baseline

(b) BayesGaze

Fig. 4: For each object described by the user, the solid blue block indicates the ground truth of what is being described, starting at the timestamp when the user began their description of the object. Concurrent with the blue blocks are various colors, signifying the objects in which the user was looking as observed by head pose tracking. The breaks in the colors show how the user in this instance was looking from object to object even while describing a single object. The *y*-axis is the objects that are being described, and the *x*-axis is time. This graph does not include 'distraction' objects not described by participants, which included a water bottle, drill, hammer, and first aid kit.

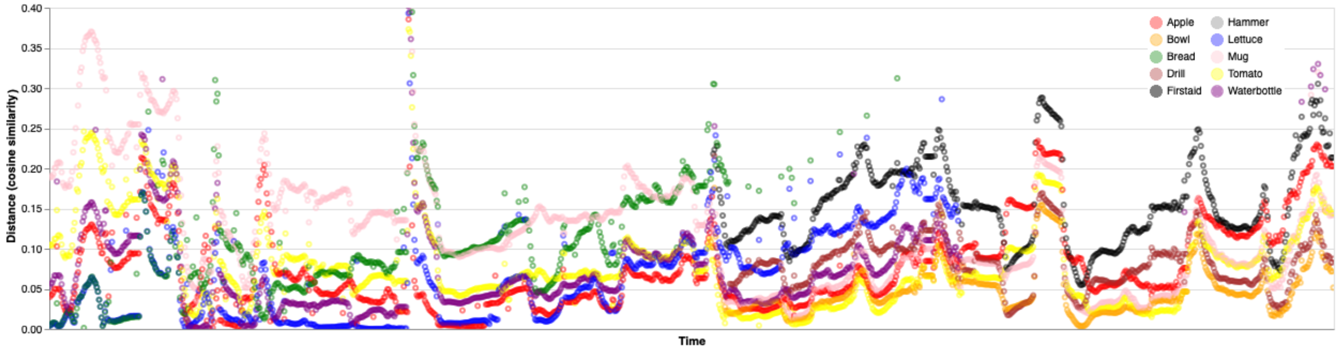


Fig. 5: The distance of the calculated gaze from each object over time, measured as cosine distance between the head pose vector and the vector between the participant's head and the objects. Different colors represent different objects; when the gaze coincides with an object, the distance drops to zero.

object containing the point with the smallest distance was considered to be what the participant was labeling. Figure 4 shows the ground truth of the object the robot is indicating (blue bars) versus the object which the gaze vector intersects (other colors) for both approaches.

2) BayesGaze: We compared this simple baseline with the BayesGaze [9] approach, originally implemented for eye tracking and targets on a 2D display. BayesGaze is an improvement on dwell-based target selection method where each target candidate accumulates "time" or "interest" from the gaze until a potential target reaches a threshold θ. It uses Bayesian inference where the prior uses a Dirichlet distribution and a Gaussian density function to determine the probability of observing a gaze given a certain target. The distance measure used was the L2 Euclidean norm of the distance between an object and the closest point on the gaze ray. In this work we focus on a 3D environment, so instead a cosine distance measure between the gaze vector and the head–object vector was used. For this approach, the centroid of each of the objects was used as their position.

This approach relies on the parameters k, the pseudocount of the Dirichlet prior; σ, which is the variance of the Gaussian density function; and θ, the target threshold. k was set to 1, and the parameters theta and sigma were determined using a grid search where σ ranges from 0.01 to 0.10 by steps of 0.01 radians, and θ ranges from 0.05 seconds to 1.00 seconds by steps of 0.05 seconds. This method was run over the data from all fifteen participants and values of σ and θ were empirically chosen ($\sigma = 0.8$, $\theta = 0.4$).

C. Results

TABLE I: Top 1, 2, 3 and MRR results for all participants

Participant	Top 1	Top 2	Top 3	MRR
1	0.54	0.83	0.88	0.73
2	0.32	0.57	0.60	0.55
3	0.64	0.86	0.89	0.78
4	0.73	0.87	0.91	0.83
5	0.72	0.87	0.99	0.84
6	0.43	0.82	0.96	0.68
7	0.50	0.76	0.92	0.70
8	0.53	0.73	0.87	0.70
9	0.58	0.85	0.92	0.76
10	0.35	0.68	0.74	0.59
11	0.50	0.64	0.85	0.67
12	0.52	0.77	0.85	0.71
13	0.52	0.78	0.83	0.70
14	0.43	0.65	0.80	0.63
15	0.58	0.84	0.98	0.76
Mean	0.51 ± 0.12	0.76 ± 0.09	0.86 ± 0.09	0.70 ± 0.08

(a) Baseline

Participant	Top 1	Top 2	Top 3	MRR
1	0.78	0.86	0.88	0.85
2	0.34	0.51	0.58	0.53
3	0.56	0.83	0.92	0.74
4	0.65	0.77	0.87	0.77
5	0.57	0.83	0.98	0.75
6	0.74	0.78	0.91	0.83
7	0.63	0.66	0.77	0.73
8	0.64	0.72	0.76	0.75
9	0.83	0.86	0.96	0.89
10	0.40	0.49	0.68	0.57
11	0.62	0.75	0.84	0.75
12	0.69	0.70	0.82	0.78
13	0.58	0.69	0.81	0.72
14	0.52	0.58	0.68	0.65
15	0.78	0.87	0.97	0.86
Mean	0.61 ± 0.14	0.72 ± 0.12	0.82 ± 0.12	0.74 ± 0.10

(b) BayesGaze

Top-N accuracy means one of the top N predictions match the target, and the mean reciprocal rank $\mathrm{MRR} = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{rank_i}$.

In order to determine how accurately these approaches identified the target object, we calculated the mean reciprocal rank of the distance calculation. Here we predict the rank of all objects based on their distance from the gaze vector, and the interest for each object from the BayesGaze approach and then the inverse rank of the desired objects in all queries are averaged. For example, if the model predicts that the person is looking at the correct object according to ground truth, MRR = $\frac{1}{1} = 1$, a perfect score. This metric is suitable for capturing the intuition that some incorrect predictions are 'closer' than others.

Table Ia and Table Ib show how often the correct object was in the top 1, top 2, top 3, and the mean reciprocal rank (MRR) for all the participants for both approaches. For the baseline the mean percentage of time in which the object was correctly identified (top 1) was 51%, top 3 was 86%, and the mean MRR was 0.70, and for BayesGaze the mean percentage of time in which the object was correctly identified (top 1) was 61%, top 3 was 82%, and the mean MRR was 0.74. The BayesGaze approach performs significantly better matching the target to the object being described, at the expense of top 2 and 3 scores. As can be seen in IIa and IIb, both approaches work better for larger objects and objects that are not clustered tightly together, with BayesGaze performing significantly better in top 1 accuracy, and slightly better overall.

These results are consistent with previous physical gaze-tracking work, which show that looking at objects separated by over 20° results in a viewer moving their head, while when viewing objects closer together than 20°, viewers move their eyes while keeping their head steady [48]. This does demonstrate that the approach described in this work is faithful to the real-world gaze tracking we are attempting to simulate. One of the confounding issues is the water bottle. Since the body of the bottle is transparent, there is a tendency to estimate that the user is looking through it to objects behind it. (Our simulator does not produce depth points for transparent objects, which is consistent with the real-world behavior of the depth sensors we intend to transfer learned models to; as a result, the point cloud of the water bottle contains few points and consists of the cap.)

The results of the study are promising in that they show the extraction of object labels in real time based on a head-pose as a proxy for gaze is feasible. Anecdotally, we discovered that participants looked at some objects more often than others, likely accounted for by varying object size and visibility; understanding this effect is one target of future work. One hypothesis is that certain objects require more visual or cognitive processing, e.g., based on the complexity of the object in shape, origin, or usage. Another possibility is the vibrancy of the colors in the objects that drive the vision to the object as a matter of immediate interest and attention captivation. Overall, while work remains, this approach to simulating gaze tracking in simulation shows promise as one of several modalities for human-robot interaction and particularly object selection.

V. FUTURE DIRECTIONS

This work forms a framework for conducting human-robot interaction experiments in multiple modalities of virtual reality. Much of the future work will be in the form of performing experiments using the simulator, rather than working on the simulator itself. However, there are still a number of interesting directions to explore which may enable additional types of research. Having multiple users

TABLE II: Top 1, 2, 3 and MRR results for all objects

Object	Top 1	Top 2	Top 3	MRR
Lettuce	0.98	0.99	0.99	0.99
Tomato	0.77	0.88	0.94	0.84
Bowl	0.60	0.85	0.92	0.75
Bread	0.44	0.96	0.98	0.71
Apple	0.30	0.77	0.98	0.60
Mug	0.002	0.06	0.33	0.12
Mean	0.51 ± 0.35	0.75 ± 0.35	0.86 ± 0.26	0.67 ± 0.30

(a) Baseline

Object	Top 1	Top 2	Top 3	MRR
Lettuce	0.99	1.00	1.00	0.99
Bread	0.98	1.00	1.00	0.99
Bowl	0.80	0.92	0.99	0.88
Tomato	0.75	0.85	0.91	0.82
Apple	0.20	0.48	0.76	0.43
Mug	0.00	0.03	0.27	0.09
Mean	0.62 ± 0.42	0.71 ± 0.39	0.82 ± 0.29	0.70 ± 0.36

(b) BayesGaze

and multiple robots able to simultaneously interact with each other could open another avenue for investigation. Our system architecture supports multiple simultaneous users.

Going forward, we plan to extend language grounding directly from speech, with the data collected from simulation used as input to a language grounding model that can then be evaluated both in simulation and again in the real world with a physical robot. The results of these two evaluations can be compared to investigate the effect of the sim2real gap. Alongside this, we can also compare how the participants perceived the interactions in both the simulated and real-world environments to investigate the effect of virtual reality on the interactions. Future studies will be inclusive of varied populations, such that each will have a representation in the robot's learning. A well represented population is made more possible through the simulation, as participants will not have to physically be in the lab. The nature of these interactions will still allow participants to interact with objects in the scenes through the controllers as other participants do in the physical interactions.

VI. CONCLUSIONS

Having developed a system to model human robot interactions which combines virtual reality with existing robotics technologies and validating it by performing a small user study, we demonstrate the immediate utility of this framework and show the potential of using similar systems to generate training data. Further, the application of head pose as a proxy for object attention as described in the study has the ability to speed of the annotation of data, speeding transition time from data collection to model training. Both the simulated environment and data labeling together allow for increasingly larger scale data collection operations to be undertaken. By releasing this system, we hope to extend the capabilities of the research community and enable new avenues of inquiry with more flexible, extensible tooling that runs on common hardware.

REFERENCES

[1] 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 preprint, Tech. Rep., 2017.

[2] G. R. Schleder, A. C. M. Padilha, C. M. Acosta, M. Costa, and A. Fazzio, "From DFT to machine learning: recent approaches to materials science–a review," *Journal of Physics: Materials*, vol. 2, no. 3, p. 032001, 5 2019. [Online]. Available: https://doi.org/10.1088/2515-7639/ab084b

[3] G. Y. Kebe, L. E. Richards, E. Raff, F. Ferraro, and C. Matuszek, "Bridging the gap: Using deep acoustic representations to learn grounded language from percepts and raw speech," in *Proc. of the AAAI Conference on Artificial Intelligence (AAAI)*, February 2022.

[4] E. Matsas and G.-C. Vosniakos, "Design of a virtual reality training system for human–robot collaboration in manufacturing tasks," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 11, no. 2, pp. 139–153, 2017.

[5] F. G. Pratticò and F. Lamberti, "Towards the adoption of virtual reality training systems for the self-tuition of industrial robot operators: A case study at kuka," *Computers in Industry*, vol. 129, p. 103446, 2021.

[6] M. Mara, K. Meyer, M. Heiml, H. Pichler, R. Haring, B. Krenn, S. Gross, B. Reiterer, and T. Layer-Wagner, "Cobot studio vr: A virtual reality game environment for transdisciplinary research on interpretability and trust in human-robot collaboration," in *Proceedings of the 4th International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 2021.

[7] V. Villani, B. Capelli, and L. Sabattini, "Use of virtual reality for the evaluation of human-robot interaction systems in complex scenarios," in *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2018, pp. 422–427.

[8] V. Weistroffer, A. Paljic, P. Fuchs, O. Hugues, J. Chodacki, P. Ligot, and A. Morais, "Assessing the acceptability of human-robot copresence on assembly lines: A comparison between actual situations and their virtual reality counterparts," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, 2014, pp. 377–384.

[9] Z. Li, M. Zhao, Y. Wang, S. Rashidian, F. Baig, R. Liu, W. Liu, M. Beaudouin-Lafon, B. Ellison, F. Wang, I. Ramakrishnan, and X. Bi, "Bayesgaze: A bayesian approach to eye-gaze based target selection," in *Graphics Interface 2021*, 2021. [Online]. Available: https://openreview.net/forum?id=r6Z8apiZQt

[10] L. E. Richards, K. Darvish, and C. Matuszek, "Learning object attributes with category-free grounded language from deep featurization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, USA: IEEE, 2020, pp. 8400–8407.

[11] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, ser. AAAI'11. San Francisco, California: AAAI Press, 2011, p. 1507–1514.

[12] T. Nguyen, N. Gopalan, R. Patel, M. Corsaro, E. Pavlick, and S. Tellex, "Robot Object Retrieval with Contextual Natural Language Queries," in *Proceedings of Robotics: Science and Systems*. Corvalis, Oregon, USA: RSS, July 2020.

[13] A. T. Nguyen, L. E. Richards, G. Y. Kebe, E. Raff, K. Darvish, F. Ferraro, and C. Matuszek, "Practical cross-modal manifold alignment for grounded language," 2020.

[14] J. Thomason, M. Shridhar, Y. Bisk, C. Paxton, and L. Zettlemoyer, "Language grounding with 3d objects," in *Conference on Robot Learning*. PMLR, 2022, pp. 1691–1701.

[15] T. Williams, M. Bussing, S. Cabrol, E. Boyle, and N. Tran, "Mixed reality deictic gesture for multi-modal robot communication," in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2019, pp. 191–201.

[16] D. Kontogiorgos, "Multimodal language grounding for improved human-robot collaboration: exploring spatial semantic representations in the shared space of attention," in *Proceedings of the 19th ACM*

International Conference on Multimodal Interaction, 2017, pp. 660–664.

[17] C. Matuszek, L. Bo, L. Zettlemoyer, and D. Fox, "Learning from unscripted deictic gesture and language for human-robot interactions," March 2014.

[18] M. Bischoff, "ROS#," Dec. 2019. [Online]. Available: https://github.com/siemens/ros-sharp/releases/tag/v1.6

[19] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system," ROS, 2018. [Online]. Available: https://www.ros.org

[20] M. Forbes, M. J.-y. Chung, M. Cakmak, and R. P. Rao, "Robot programming by demonstration with crowdsourced action fixes," in *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

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

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

[23] 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," 2017.

[24] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [Online]. Available: https://arxiv.org/abs/1912.01734

[25] 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 preprint, Tech. Rep., 2017.

[26] M. Chevalier-Boisvert, D. Bahdanau, S. Lahlou, L. Willems, C. Saharia, T. Nguyen, and Y. Bengio, "Babyai: A platform to study the sample efficiency of grounded language learning," New Orleans, Louisiana, 2019, iCLR.

[27] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, and S. Lee, "Sim-to-real transfer for vision-and-language navigation," 2020, 2011.03807.

[28] C. Bartneck, M. Soucy, K. Fleuret, and E. B. Sandoval, "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)*. Kobe, Japan: IEEE, 2015, pp. 431–437.

[29] T. Inamura and Y. Mizuchi, "Sigverse: A cloud-based vr platform for research on social and embodied human-robot interaction," 2020.

[30] F. Stramandinoli, K. G. Lore, J. R. Peters, P. C. O'Neill, B. M. Nair, R. Varma, J. C. Ryde, J. T. Miller, and K. K. Reddy, "Robot learning from human demonstration in virtual reality," *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 2018.

[31] D. Whitney, E. Rosen, and S. Tellex, "Learning from crowdsourced virtual reality demonstrations," *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 2018.

[32] A. Jackson, B. D. Northcutt, and G. Sukthankar, "The benefits of teaching robots using vr demonstrations," in *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 129–130. [Online]. Available: https://doi.org/10.1145/3173386.3176980

[33] E. Rosen, D. Whitney, M. Fishman, D. Ullman, and S. Tellex, "Mixed reality as a bidirectional communication interface for human-robot interaction," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 11 431–11 438.

[34] A. Moon, D. M. Troniak, B. Gleeson, M. K. Pan, M. Zheng, B. A. Blumer, K. MacLean, and E. A. Croft, "Meet me where i'm gazing: How shared attention gaze affects human-robot handover timing," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 334–341. [Online]. Available: https://doi.org/10.1145/2559636.2559656

[35] A. Pereira, C. Oertel, L. Fermoselle, J. Mendelson, and J. Gustafson, *Effects of Different Interaction Contexts When Evaluating Gaze Models in HRI*. New York, NY, USA: Association for Computing Machinery, 2020, p. 131–139. [Online]. Available: https://doi.org/10.1145/3319502.3374810

[36] H. Admoni, C. Datsikas, and B. Scassellati, "Speech and gaze conflicts in collaborative human-robot interactions," in *Proceedings of Annual Meeting of the Cognitive Science Society (CogSci '14')*, vol. 36, 2014, pp. 104 – 109.

[37] P. Baxter, J. Kennedy, A.-L. Vollmer, J. de Greeff, and T. Belpaeme, "Tracking gaze over time in hri as a proxy for engagement and attribution of social agency," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 126–127. [Online]. Available: https://doi.org/10.1145/2559636.2559829

[38] S. Ivaldi, S. Lefort, J. Peters, M. Chetouani, J. Provasi, and Z. Elisabetta, "Towards engagement models that consider individual factors in hri: On the relation of extroversion and negative attitude towards robots to gaze and speech during a human–robot assembly task," *International Journal of Social Robotics*, vol. 9, 01 2017.

[39] R. M. Aronson, N. AlMutlak, and H. Admoni, "Inferring goals with gaze during teleoperated manipulation," in *Proceedings of (IROS) IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[40] R. M. Aronson and H. Admoni, "Eye gaze for assistive manipulation," in *Proceedings of Companion of the ACM/IEEE International Conference on Human-Robot Interaction (HRI '20)*, 2020, pp. 552 – 554.

[41] G. Mehlmann, M. Häring, K. Janowski, T. Baur, P. Gebhard, and E. André, "Exploring a model of gaze for grounding in multimodal hri," in *Proceedings of the 16th International Conference on Multimodal Interaction*, 2014, pp. 247–254.

[42] Y. Nakano, G. Reinstein, T. Stocky, and J. Cassell, "Towards a model of face-to-face grounding," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, Jul. 2003, pp. 553–561. [Online]. Available: https://aclanthology.org/P03-1070

[43] S. Pastel, C.-H. Chen, L. Martin, M. Naujoks, and K. Petri, "Comparison of gaze accuracy and precision in real-world and virtual reality," *Virtual Reality*, vol. 25, 03 2021.

[44] C. George, D. Buschek, A. Ngao, and M. Khamis, "Gazeroomlock: Using gaze and head-pose to improve the usability and observation resistance of 3d passwords in virtual reality," in *Augmented Reality, Virtual Reality, and Computer Graphics: 7th International Conference, AVR 2020, Lecce, Italy, September 7–10, 2020, Proceedings, Part I*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 61–81.

[45] A. Esteves, D. Verweij, L. Suraiya, R. Islam, Y. Lee, and I. Oakley, "Smoothmoves: Smooth pursuits head movements for augmented reality," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 167–178. [Online]. Available: https://doi.org/10.1145/3126594.3126616

[46] O. Palinko, F. Rea, G. Sandini, and A. Sciutti, "Robot reading human gaze: Why eye tracking is better than head tracking for human-robot collaboration," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 5048–5054.

[47] J. Kennedy, P. Baxter, and T. Belpaeme, "Head pose estimation is an inadequate replacement for eye gaze in child-robot interaction," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, ser. HRI'15 Extended Abstracts. New York, NY, USA: Association for Computing Machinery, 2015, p. 35–36. [Online]. Available: https://doi.org/10.1145/2701973.2701988

[48] S. O. Ba and J.-M. Odobez, "Recognizing visual focus of attention from head pose in natural meetings," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 16–33, 2009.

[49] S. Andrist, M. Gleicher, and B. Mutlu, "Looking coordinated: Bidirectional gaze mechanisms for collaborative interaction with virtual characters," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 2571–2582. [Online]. Available: https://doi.org/10.1145/3025453.3026033

[50] MakeHuman Community, "Makehuman." [Online]. Available: http://http://www.makehumancommunity.org/

[51] RootMotion, "Final IK." [Online]. Available: http://www.root-motion.com/final-ik.html