An Approach to Deploy Interactive Robotic Simulators on the Web for HRI Experiments: Results in Social Robot Navigation

Nathan Tsoi, Mohamed Hussein, Olivia Fugikawa, J. D. Zhao, and Marynel Vázquez

Abstract— **Evaluation of social robot navigation inherently requires human input due to its qualitative nature. Motivated by the need to scale human evaluation, we propose a general method for deploying interactive, rich-client robotic simulations on the web. Prior approaches implement specific web-compatible simulators or provide tools to build a simulator for a specific study. Instead, our approach builds on standard Linux tools to share a graphical desktop with remote users. We leverage these tools to deploy simulators on the web that would typically be constrained to desktop computing environments. As an example implementation of our approach, we introduce the SEAN Experimental Platform (SEAN-EP). With SEAN-EP, remote users can virtually interact with a mobile robot in the Social Environment for Autonomous Navigation, without installing any software on their computer or needing specialized hardware. We validated that SEAN-EP could quickly scale the collection of human feedback and its usability through an online survey. In addition, we compared human feedback from participants that interacted with a robot using SEAN-EP with feedback obtained through a more traditional video survey. Our results suggest that human perceptions of robots may differ based on whether they interact with the robots in simulation or observe them in videos. Also, they suggest that people perceive the surveys with interactive simulations as less mentally demanding than video surveys.**

I. Introduction

Mobile robot navigation with and around people, i.e. *social navigation*, is important for robotic applications in human environments, including service robotics [1], [2], [3], healthcare [4], and education [5], [6]. Thus, social robot navigation has long been studied from a technical and experimental perspective [7], [8], [9], [10], [11]. Yet, there is no agreed-upon protocol for evaluating these systems because: (a) robots often have different capabilities, making comparisons difficult; (b) implementing robust navigation systems is hard, thus baselines do not necessarily represent the state of the art; and (c) there is a lack of standard human-driven evaluation metrics because the context of navigation tasks can significantly alter what matters to users [12], [13]. These issues have hindered advancements and make it difficult to understand the key challenges that the social robot navigation community faces today.

While simulations may not be perfect replicas of the real world, we believe that they provide a viable path towards standardizing the evaluation of social robot navigation systems in Human-Robot Interaction (HRI). Our rationale is twofold. First, simulations have long been leveraged to conduct early tests, stress tests, and verification of robotic systems [14], [15], [16]. They provide controlled environments to systematically study critical application scenarios, and can be used for benchmarking [17], [18], [19], [20]. Second, simulations can be integrated with real-time robotic software, facilitating sim-to-real transfer. For example, interfaces for the Robot Operating System (ROS) have enabled running robot stacks in a variety of simulators [21], [22], [23], [24].

Given prior work in robotics simulations, what is the key challenge that prevents us from fully leveraging simulations for evaluating systems in HRI? The problem is that the evaluation requires human input because the social aspects of robot navigation are subjective in nature. One option for gathering human input is to utilize web-based surveys and crowd workers. However, modern simulations are compute-intensive applications designed for local use in a desktop computing environment. That is, these simulators are *rich-client* applications that provide rich functionality independent of a remote server – in contrast to thin-client applications which are heavily dependent on remote processing, like browser-based web applications. This makes the modern simulators inaccessible to crowd participants who are limited to browser-based web applications.

Making a rich-client application, like a robotics simulation, available on the web is a non-trivial task. Perhaps one could think of re-implementing the software under the constraints of a web browser [25] and application-specific or web server-specific modules such as [26]. However, some re-implementations are too complex, time consuming, or even infeasible due to the lack of specific dependencies such as a programming language, physics engine, or rendering engine. Another option could be to use specific solutions that make applications such as word processor programs available in a web browser [26]. Unfortunately, these solutions do not generalize well to robotic simulators that require high performance graphics rendering via specialized hardware and libraries such as OpenGL. These challenges have often restricted human evaluation of social robot navigation via crowd-sourcing to video surveys (e.g., [27], [28]). While videos may lead to comparable results to in-person studies in some cases [29], they are passive mediums with low interactivity [30].

In this work, we propose a method of making rich-client, interactive robotic simulators accessible at scale on the web. As an example implementation, we introduce the SEAN Experimental Platform (SEAN-EP), an open-source system that allows roboticists to gather human feedback for

This work was supported by a 2019 Amazon Research Award. M. Hussein was supported by the National Science Foundation Grant No. 1910565.

Artwork provided by Xaiver Ruiz, Yale College Computing and the Arts.

The authors are with the Interactive Machines Group at Yale University, 51 Prospect Street, New Haven, CT 06511 and Rutgers University-Camden, 303 Cooper St, Camden, NJ 08102, USA. Contact: nathan.tsoi@yale.edu

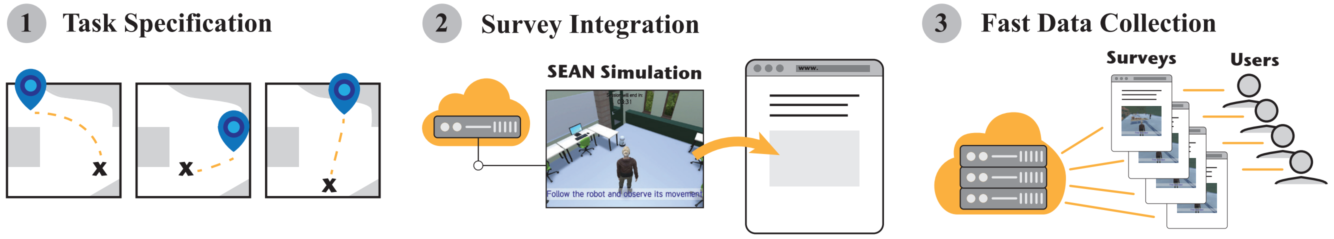


Fig. 1: With SEAN-EP, researchers can scale HRI experiments in the context of navigation via 3 steps: (1) experimenters specify navigation tasks in the simulator, (2) they integrate interactive simulations based on the tasks with online surveys, and (3) they collect data in parallel from multiple users. See the text for more details.

social robot navigation via online simulations, as illustrated in Figure 1. Though our implementation uses the Social Environment for Autonomous Navigation (SEAN) [31] as the underlying simulator, any rich-client simulator that runs in Linux could be deployed using our method.

We validated our implementation and its usability through an online study about social robot navigation. Further, we investigated whether human perceptions of robots differ based on whether they experience human-robot interactions via online simulations or watch them through videos. Interestingly, our results suggest that interactive surveys are less mentally demanding than non-interactive video surveys.

In summary, our work has four main contributions:

1) A novel approach to deploy rich-client robot simulation environments at scale using standard web technologies. This method allows one to quickly gather human feedback in HRI.

2) SEAN-EP, a specific instantiation of the proposed approach based on the Social Environment for Autonomous Navigation. SEAN-EP is open-source and available at https://github.com/yale-sean/social_sim_web.

3) Validation of our example implementation (SEAN-EP) through an online study about social robot navigation.

4) An experimental comparison of interactive simulations and videos for studying human perception of robot navigation.

II. Related Work

A. Robotics Simulation Environments

Progress has been made on developing photorealistic simulations which bridge the gap between virtual worlds and reality [32], [33]. With their high visual fidelity and responsiveness, game engines such as Unity and Unreal Engine have proved indispensable to robotic simulation of flying and mobile robots [22], [23], [34], [35]. Several of these simulation environments integrate with ROS to achieve realistic robot control and transfer results to the real world.

Within social robot navigation, crowd simulation and modeling of pedestrian behavior have improved as well [36], [37], [38]. However, there has been less work on combining crowd models with robotics simulation for robot navigation in human environments. One exception is the Social Environment for Autonomous Navigation (SEAN) [31], which we leverage in our work. SEAN builds on Unity and integrates with ROS, making it a good option in terms of photorealism and future sim-to-real transfer.

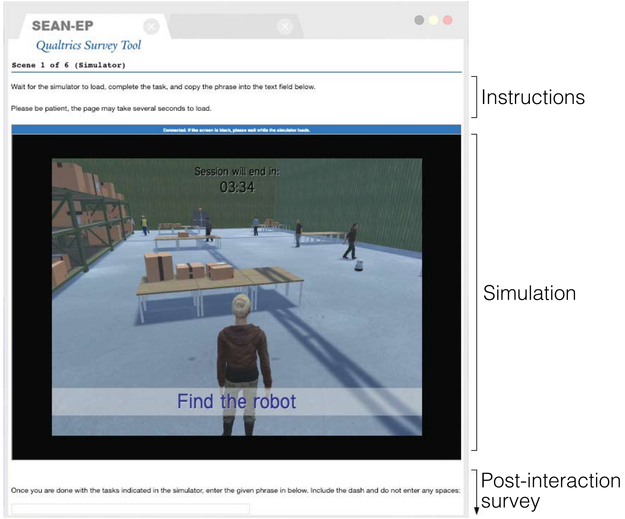
Modern robotics simulations are rich-client applications meant to run in a desktop computer or powerful gaming laptop. While some simulators may utilize frameworks that provide the option to compile to WebGL for deployment on the web, like Unity, there are many challenges and limitations to this approach. This includes lack of direct access to IP sockets from WebGL due to security implications, limitations in rendering and illumination, lack of threading in JavaScript, and limited access to hardware [39]. While specific limitations can be addressed via engineering workarounds on a case-by-case basis, this approach lacks the flexibility of our method. Our method works with any kind of rich-client simulation that runs in Linux, even if it requires interacting with other software such as ROS components.

B. Leveraging the Web in HRI

While traditional HRI experiments are conducted in person, prior work has explored faster mechanisms that leverage the accessibility of the Web. For example, [40] explored using a two-player online game to build a data corpora for HRI research. Also, [41], [42], [43] developed web-compatible simulations based on Unity and WebGL, although these systems have the limitations discussed in Sec. II-A.

Especially during early HRI system development, it has become common practice to gather human feedback via online video surveys [44], [45], [46], [27], [28]. Research has indicated that there is a moderate to high level of agreement for subjects' preferences between live and video HRI trials [29]. Our work contributes to a better understanding of experimental methods by comparing human feedback from a video survey with feedback from an interactive survey.

Our approach is inspired by internet frameworks that provide methods to remotely interact with robotic systems and simulators. For example, the Robot Management System (RMS) [47] and Robot Web Tools [48] provide software for building web-based HRI interfaces and demonstrate their approach in a variety of tasks, including remote robot operation. These tools allow web clients to interface with ROS and the Gazebo simulator by transforming ROS-specific data streams into formats compatible with a web browser. While ROS applications can be made accessible on the web via RMS, our approach is relevant to all kinds of rich-client simulators that run on Linux, not just Gazebo.



SEAN-EP

Qualtrics Survey Tool

Scene 1 of 6 (Simulator)

Wait for the simulator to load, complete the task, and copy the phrase into the text field below.

Please be patient, the page may take several seconds to load.

Connected. If the screen is black, please wait while the simulator loads.

Session will end in:
03:34

Find the robot

Once you are done with the tasks indicated in the simulator, enter the given phrase in below. Include the dash and do not enter any spaces:

Fig. 2: Screenshot of Qualtrics survey with embedded SEAN simulation. Best viewed in digital form.

Lastly, RoboTurk [49] allows for rapid crowdsourcing of high-quality demonstrations for robot learning in the context of manipulation. While our method could be used in the future to gather data for learning navigation policies through teleoperation, in this work we explore giving users control of a human avatar in the simulation. This methodology aims to bring their online, virtual experience closer to real-world human-robot interactions.

III. Method

We propose a general approach to deploy the graphical user interface (GUI) of rich-client robotic simulators on the web to facilitate and scale HRI experiments. Our approach builds on standard tools for graphical desktop sharing in Linux. It does not require the adaptation of rich-client simulators to other technologies, such as WebGL.

With our approach, researchers can create interactive HRI surveys. These are surveys that, in some parts, include simulations in which the participant interacts with a robot. Figure 2 shows an example online survey used to study social robot navigation. This simulation was embedded in the survey using a particular instantiation of our approach, as later described in Sec. IV. After the simulations end, the online survey can query the participant for explicit feedback about his or her experience interacting in the virtual world.

Importantly, our method addresses parallelization challenges inherent to online studies typically run via crowdsourcing platforms. While simulation systems for in-person HRI studies are usually designed for one participant at a time, online surveys must cope with a potentially large number of people who participate in the study simultaneously. Our method provides a mechanism to scale simulations designed for a single user to many users in parallel. This is possible without changes to the underlying system.

The next sections describe in detail our method to make rich-client robotics simulations accessible on the web. We evaluate an implementation of this approach in Section V.

A. Making Interactive Simulations Accessible on the Web

We propose to make rich-client simulations available in a standard web browser by running them on a remote server, and using a Virtual Network Computing (VNC) server to share the GUI of the simulator with a remote user.

While remote users typically connect to a VNC server via a desktop VNC client running on their machine, we use a browser-based VNC client running on the host server to allow browser-based access to the simulator GUI. The browser-based VNC client renders the GUI on a web page, through which our system can accept user input for the simulator, e.g., keyboard commands.

One important consideration when exposing the GUI of a simulator on the web as described before is that users are unauthenticated and untrusted. Thus, it is important that the GUI of the simulator does not provide mechanisms to launch other processes on the remote server.

Because VNC connections are designed to be used by a single user, we propose the use of a web-based process orchestration tool to deploy and manage a large number of concurrent simulation environments. We call this tool the "Process Manager" because it controls the execution of processes associated with each simultaneous user. This tool is further described in the next section, where we explain in detail how to scale data collection on a single host.

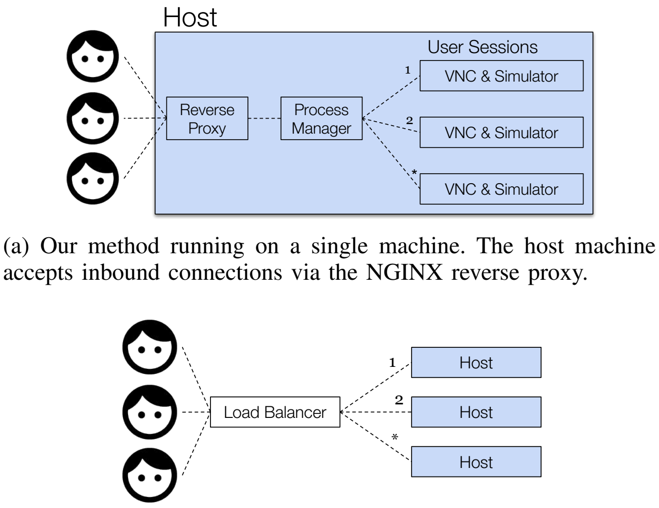
B. Scaling on a Single Host

To scale human feedback collection, we can run multiple instances of the simulator on the remote server and provide individual remote users access to one of them. Figure 3a illustrates how we achieve this goal using a reverse proxy server and a Process Manager. The Process Manager is responsible for managing *user sessions*, which include an instance of the simulator (including its GUI), all other components necessary for the simulator to run, and a VNC server and browser-based client for the given user.

The reverse proxy routes web requests that are received via Hypertext Transfer Protocol Secure (HTTPS) to specific web servers on the host machine based on the URL path of the request. The target web server may be the Process Manager, which is in charge of initiating, maintaining, and terminating user sessions, or an existing web-based VNC client within the user's session.

Requests for simulations should have a specific URL path that includes a parameter for a unique user identifier, e.g., a Mechanical Turk ID. Additionally, they should include any other parameters needed to instantiate the simulation for the user. For instance, in the example implementation described in Sec. IV, the requests include start and goal poses for a user's avatar and a robot in the simulation.

When the Process Manager receives a request from the reverse proxy, it evaluates if it is a new request given the URL parameters. If that is the case, then the Process Manager launches the main components that make up an interactive simulation session and quickly redirects the request to the page of the web-based VNC client that corresponds to the user. Because the VNC web page is served on the same host,



(a) Our method running on a single machine. The host machine accepts inbound connections via the NGINX reverse proxy.

(b) Our method running on multiple machines. Each host has all the components shown in Fig.3a.

Fig. 3: Proposed methods to render the GUI of rich-client simulations on the web and scale HRI data collection.

the reverse proxy gets the request that results from the URL redirection and appropriately routes it so that the user's web client can display the simulation's GUI. If an existing user requests a running simulation, the Process Manager simply redirects the request to the corresponding VNC URL.

The Process Manager is also in charge of managing maximum session duration. Sessions are allowed to run for a configurable amount of time before being automatically shut down. Once a session is closed, the resources can be re-allocated to new sessions for other users.

Handling simulation requests as described above is beneficial in 3 key ways: (1) it is easy to integrate simulations with online surveys because a single web address (with parameters) is used to handle all requests; (2) the entire connection between a remote user and the host machine is encrypted over an HTTPS connection; and (3) because the reverse proxy is routing requests rather than having users connect directly to each VNC instance, there is no need to expose many non-standard internet ports on the host.

Crucially, this method allows a single host to handle the multiple concurrent requests required of an online study where many users need access to interactive sessions to complete surveys.

C. *Scaling Across Many Machines*

Scaling of simulator sessions on a single host machine is limited by the hardware resources on the host. The ability to scale users' sessions across many machines, or scale "horizontally," removes this limitation.

Horizontal scaling can be achieved by adding a Load Balancer to our proposed system. The Load Balancer receives user requests and then acts as a "traffic cop" to evenly distribute the requests to the available host machines, as illustrated in Figure 3b. This routing process must be "session aware" to associate users to the same host machine if they perform multiple requests.

With the guarantee that a single host will receive all requests for a unique user, the system state does not need to be shared across hosts, but can be managed in the same way as described in Sec. III-B. Moreover new hosts can be added dynamically to handle more concurrent sessions by simply notifying the Load Balancer.

IV. SEAN-EP: A System to Scale Human Feedback for Social Robot Navigation

We created the SEAN Experimental Platform (SEAN-EP) in order to validate our method (Sec. III) in the context of social robot navigation. Our goal was to test our method's feasibility in a realistic usage scenario and, through this effort, verify the key tenets of scalability and usability.

SEAN-EP uses SEAN [31] as the core simulator. SEAN provides photorealistic virtual worlds, crowd simulations for social robot navigation, and integration with ROS for robot control. We modified SEAN to use the Microsoft Rocketbox avatars library [50] for this work because these avatars are higher-quality than those used in [31].

A. *System Implementation*

We implemented our method as an open-source system and deployed it to virtual hosts with dedicated NVIDIA T4 GPUs using Amazon Web Services (AWS). We aimed to maximize the performance of our system and fully utilize the hardware resources to deliver a user friendly and visually appealing simulated interaction that is free of glitches or lag. To this end, we chose TurboVNC as the VNC server, which accelerates data transfer by compressing images via libjpeg-turbo. We made TurboVNC available on the web using noVNC with websockify.[1] Notably, TurboVNC also supports VirtualGL for hardware accelerated 3D graphics.

We used the open-source NGINX server as reverse proxy and implemented the Process Manager using the popular Flask web framework. We developed a custom configuration for NGINX to properly route requests as specified by the Process Manager. Also, we configured an AWS Application Load Balancer with sticky-sessions to make it session-aware.

The Process Manager facilitates communication between the SEAN GUI and ROS. Because each session requires its own instance of ROS, we encapsulate ROS processes in a Docker container and expose a single network port for the SEAN GUI to communicate with its ROS instance.

We used ROS bag files as the main logging mechanism for human-robot interactions enabled by SEAN-EP.

B. *Navigation Tasks*

To evaluate our idea in practice and conduct an experiment about perceptions of social robot navigation (Sec. V), we designed three tasks for users to complete in SEAN simulations. First, they had to find the robot. Second, they had to follow the robot and observe its movements, which required them to stay in proximity to the robot and observe

[1] https://github.com/novnc/websockify

its interactions with other people. Third, they had to navigate to a nearby location in the environment identified with a visual landmark. This last task incentivized them to navigate around the robot to reach their destination. Overall, these tasks motivated users to both interact with the robot in the virtual world and behave in naturalistic ways.

C. User Interface

We created a new user interface in SEAN to let a user control an avatar in the simulation and make the virtual experience similar to real human-robot interactions. We had two key requirements when designing the user interface: it had to be accessible to a wide range of users; and it needed to be simple enough to be explained in a short introductory tutorial. Given these requirements, we chose to implement an interface that is similar to a third-person video game, albeit with simplified controls. The main camera of the simulation follows the user's avatar as it moves. Users can press the up and down arrow keys to raise and lower the camera, changing the field of view of the environment as needed. In addition, they can use the keyboard commands W, A, S, D to move their character forward, left, backward, and right, respectively. These keyboard commands were captured in the users' browsers and seamlessly passed to the SEAN simulation using noVNC.

D. Data Collection via Online Survey

With our system, SEAN simulations can be integrated with standard online survey platforms via HTML iframe elements. The surveys can collect any additional data from users, such as demographic data or answers to questions about their experience in the simulations. An example is provided in the evaluation presented in Sec. V, for which we integrated SEAN simulations with a Qualtrics survey (Figure 2).

E. Performance

SEAN-EP provides users web access to interactive SEAN simulations with a small amount of load time. When a user requests a new SEAN simulation session, the Process Manager starts a complete ROS environment, the Unity-based simulator GUI, and a VNC server and client. Despite all these many programs, the start-up time for a user session is 18.9s on average. Transferring the simulator's GUI to the user's browser through noVNC takes on the order of milliseconds with a low Internet connection speed in the U.S. (e.g., on the order of 10 Mbps). This means that the total wait time for users to access a SEAN simulation with SEAN-EP is significantly faster than compiling SEAN Unity worlds to WebGL. The reason is that the worlds are complex, resulting in simulations that are over 1.5GB in size after the compilation. With a global average fixed broadband download speed of 77 Mbps, transferring a single WebGL environment to the browser would take over 2.5 minutes.[2]

V. SEAN-EP Evaluation

We used a Qualtrics online survey to validate the potential of our method to gather human feedback for social robot navigation. The survey included 6 interactive simulations, embedded through HTML tags, through which users could interact with a Kuri robot. The next sections detail our experimental protocol and results, with a special focus on user feedback obtained through the survey. Section VI later compares using this type of interactive survey versus a video survey to gather human feedback about robot navigation. The protocols for these studies were approved by our local Institutional Review Board.

A. Method

The Qualtrics survey was designed to gather feedback about robot navigation in two simulated indoor environments. One environment was a warehouse that included 15 virtual humans, a Kuri robot, and the user's avatar (Fig. 2). The other environment was a computer laboratory, which included one virtual human besides the robot and the user's avatar (Fig. 1). The goal of the user in the simulations was to first find the robot, then follow it for 30 seconds, and finally navigate to a destination identified by a visual landmark.

Experimental Protocol. The survey began with a demographics section. Then, the participants were asked to behave politely in the simulator and were introduced to the task with a short simulation in the lab environment. This simulation served as a tutorial to explain the commands that the participants could use to move their avatar, identify Kuri, and practice navigation tasks. After the tutorial, the participants experienced 6 simulations in randomized order: 3 in the warehouse environment and 3 in the laboratory. For each simulation, there were specific start and goal locations for all agents. In particular, the navigation goals of the robot and the human avatar were opposite to each other, so that they would easily encounter one another at some point in the virtual world. After each interactive simulation, the participants were asked a few questions about their experience, including whether they were able to identify the robot and whether it moved in the environment. At the end of the survey, the participants were asked about their overall experience.

Robot Control. The Kuri virtual robot was modeled after the real platform manufactured by Mayfield Robotics. It had a differential drive base, and used a simulated 2D LIDAR and odometry information from Unity to localize against a known map. All path planning and execution was completed by the ROS Navigation Stack, which used a global and local costmap for object avoidance and social navigation around virtual humans [8]. We opted to use the Navigation Stack because it is widely used by many robots, including TurtleBot platforms, PR2, and the Clearpath Husky. Also, it is used as a classical baseline and ground truth by more modern learning-based approaches [27], [51].

Participants. We recruited 62 participants through Prolific, a crowdsourcing platform, for this evaluation. Participation was limited to individuals 18 years or older, fluent in English

[2] Note that the complete SEAN simulation cannot be exported to WebGL due to ROS dependencies. Thus, we only report the expected time that it would take to load the Unity world after converting to WebGL.

with normal-to-corrected vision. The participants had an average age of 32 years old and 29 were female. In general, the participants were familiar with video games (M=6.1, $\sigma = 1.1$), but not as familiar with robots (M=4.0, $\sigma = 1.4$) based on answers on a 7-pt responding format (1 being the least familiar, 7 being the most familiar). They were paid $4.00 USD for completing the survey.

System Architecture. Because our interest was testing the proposed system, we ran participants under both strategies to scale simulations (Sec. III). Half of the participants experienced simulations running on a single host machine and thus were run in small batches to avoid overloading the host. The other half interacted with simulations distributed across four machines. Virtual machines were AWS g4 EC2 instances with 32 cores, 124GB of RAM, and 15.84GB of GPU memory. Given the requirements of our simulation environment, each machine was capable of running up to 30 interactive simulation sessions in parallel, based on their GPU memory and the size of our SEAN environments. In the case of scaling across many machines, we limited the number of sessions per host below the resource-constrained maximum to 10. We also added enough hosts behind the Load Balancer to accommodate the maximum number of total simultaneous participants in our study.

B. Results

We were able to successfully gather data by using a single host machine as well as multiple ones. With a single host, we ran on average 5 participants at a time collecting all 31 responses in about 8 hours. With the multiple host approach, we ran all 31 participants at the same time and collected all 31 responses in 1.5 hours. As a reference, the average participant took about 31 minutes to complete the survey.

While the multiple host approach effectively reduced the time that it took to collect data through the surveys by 81%, it required more management overhead. This included managing instances in the pool and moving collected data from the machines to a shared store for analysis.

Task Completion. While not all participants followed the instructions by the book, a large majority tried and were able to complete the given tasks, validating that our system worked as intended. In only 23 of 372 interactive trials (6.18%) the participant's simulation session timed out before they reached their goal destination. Considering all 372 interactive sessions there were only 8 interactive sessions (2.15%) in which participants did not move from the starting position. Further inspection of the data revealed that the only two sessions in which the participants failed to find the robot in the simulation corresponded to timed out sessions in which their avatar moved. These simulations were in the warehouse environment, suggesting that they tried to find the robot but the large space made it difficult for them to identify it. Because the study sessions were conducted in parallel, at scale, they help confirm the scalability of our method.

Navigation Behavior. Using ROS logs from SEAN, we checked how often the human's avatar and the robot were close to each other based on Hall's proxemic zones [52]. We set a threshold for intimate space of 0.45 meters, and found that in 195 sessions (52.4%) the robot came within this distance from the participant's avatar. In terms of personal space, in 316 sessions (84.9%) the robot came within 1.2 meters from the avatar. A benefit of simulations is that we can easily analyze proxemic behavior as shown by these results.

User Experience. At the end of the survey, the participants reported that the survey tasks required low mental demand (M=2.26, SE=0.19) and low physical demand (M=1.61, SE=0.15) on a 7-point responding format where 1 indicated the lowest demand. They did not have to work hard to accomplish the tasks (M=2.32, SE=0.18).

Some participants provided positive feedback for our system through open-ended questions. For example, one person said that *"the game was very well made and the controls are what I'm used to with my own gaming."* Another said they *"found the instructions were easy to follow."* When asked if the virtual world was confusing, participants strongly indicated it was not confusing (M=2.03, SE=0.18). However, a few participants reported confusing elements of the survey. Eight people believed the robot's motion was awkward. Additionally, 7 people thought that the control of their human avatar was unintuitive. As a reference, the participants had an average internet speed of 105.56 Mbps (SE=11.09).

Overall, these results validate the feasibility of our method to enable online, interactive HRI studies.

VI. Interactive Simulation vs. Video Feedback

We compared the interactive human feedback obtained using SEAN-EP with feedback obtained through a video survey, which is a typical approach to online HRI studies as discussed in Section II. To this end, we recruited 62 more participants through Prolific. These participants provided feedback about the robot based on videos of the simulations that happened as part of our prior study (Section V).

A. Method

Experimental Protocol. We expanded our data from Sec. V with data collected through a Qualtrics video survey. In general, the video survey followed the same format as the prior one. However, instead of having participants interact with the robot in a virtual world, each participant viewed the 6 video recordings of the simulations experienced by a participant from our validation study. After watching each video, they were asked about the observed robot.

Hypotheses. The data from Sec. V (Interactive condition) and the video survey (Video condition) were analyzed together to investigate two hypotheses:

H1. *The perception of the robot would differ between the conditions.* To test this hypothesis, we gathered ratings for the Competence and Discomfort factors of the Robotic Social Attributes Scale (ROSAS) [53] (Cronbach's α was 0.938 and 0.746, respectively). We also gathered participants' opinions on whether the robot navigated according to social norms after each simulation session or corresponding video.

H2. *The perceived workload for the survey in the Interactive condition would be lower than in the Video condition.* We measured perceived mental and physical demand along with effort at the end of the surveys based on responses to the following questions from the NASA Task Load Index [54]: "How mentally demanding were the tasks?", "How physically demanding were the tasks?", and "How hard did you have to work to accomplish what you had to do?". Responses were collected on a 7 point responding format (1 being lowest, 7 being highest).

Participants. A total of 124 participants were considered for this experiment (62 from Section V plus 62 new participants). Their average age was 34 years old and 55 of them were female. We limited participation in the same way as Sec. V. Participants were paid $4.00 USD for completing the survey.

B. Results

Human Perception of the Robot. There were 2 simulation sessions out of 372 in the Interactive condition in which the participants failed to identify the robot and 53 sessions in which they said it was not moving. Meanwhile, there were 4 sessions out of 372 in the Video condition in which the participants failed to identify the robot and 51 sessions in which they said that it was not moving. We excluded these sessions from further analysis about perceptions of the robot.

For the 288 pairs of sessions in which participants both saw the robot moving in the simulation and the video condition, we conducted a Wilcoxon signed-rank test for the paired data to check if the Competence and Discomfort factors from ROSAS (in 7-point responding format, 1 being lowest and 7 being highest) differed by condition. The test resulted in no significant differences for Discomfort. The median Discomfort was 2.33 for the Interactive condition and 2.17 for the Video condition. However, the Wilcoxon test revealed significant differences for robot Competence ($p < 0.0001$). The median Competence value was 3 for the Interactive condition and 3.83 for the Video condition. Lastly, an additional paired Wilcoxon signed-rank test indicated significant differences by Condition in terms of whether the robot navigated according to social norms. The mean rating was 3 over 7 points for the Interactive condition and 4 over 7 for the Video condition. These different results provided evidence in support of our first hypothesis (H1).

Workload. We conducted an additional paired Wilcoxon signed-rank test to evaluate potential differences in perceived workload across Conditions. The test indicated significant differences for perceived mental demand ($p < 0.01$). The median rating for mental demand in the Interactive condition was 2 points over 7, while the median for the Video condition was 3. No significant differences were found for the ratings about physical demand. The median rating was 1 – the lowest possible – for both conditions. Lastly, we found significant differences for how hard the participants had to work to complete the surveys ($p = 0.023$). The median rating for the Interactive condition was 2 over 7 points, and the median for the Video condition was 3. These results partially support H2.

C. Discussion

We suspect that the different results across conditions were due to the interactive nature of the simulation, which provided better opportunities for the participants to evaluate the responsiveness of the robot to their actions and to other virtual humans than the video. However, further tests are needed to validate this assumption. For example, future tests could consider human perception of the robot and the perceived workload in both the real world and the simulated replica. Another aspect to consider is user perspective. While we used a third-person perspective in SEAN-EP, studying interactions perceived from a first-person perspective could better translate to the real world. An additional consideration for future work is the display type used by the participants.

VII. Limitations & Future Work

Our approach to make interactive simulations available on the web was effective in general. It allowed users to control their virtual avatars in rich-client simulations and quickly gather data to study social robot navigation. In the future, we would like to use SEAN-EP to also allow users to control the robot, so that we can collect example behaviors for social robot navigation. We also wish to explore other types of common navigation scenarios [55], [56], [10], e.g. walking alongside a robot or passing in narrow spaces.

We observed that most survey participants tried to complete the simulated tasks in a polite and naturalistic manner as directed. There were several people however, who explored undesired actions for their avatars. About 18% of the participants pushed the robot in the simulation, and about 12% collided with a human based on annotations from our video survey. In the future it is important to explore incentives for participants to reduce these undesired behaviors.

Lastly, we evaluated our proposed approach using a single robotics simulator. Given the flexibility of our method, we would like to see it being used to make other rich-client simulators for Linux easily available on the web. This could facilitate human feedback collection in other HRI domains.

VIII. Conclusion

We introduced a flexible method to enable crowd-sourcing of human feedback using interactive rich-client simulators deployed on the web. We then demonstrated a particular instantiation of this approach, called SEAN-EP, in the context of social robot navigation.

We tested SEAN-EP with an online survey, which validated its ability to serve simulations to many users. Furthermore, we compared the results of evaluating robot navigation through interactive simulations using our method against evaluations based on video surveys. Our proposed interactive methodology resulted in different perception of the robot and lower mental demand for participants.

We are excited about the potential of conducting future online HRI experiments at scale with interactive, virtual experiences. We also look forward to leveraging SEAN-EP to scale benchmarking of social robot navigation methods.

References

[1] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte *et al.*, "Minerva: A second-generation museum tour-guide robot," in *ICRA*, 1999.

[2] V. Evers, N. Menezes, L. Merino, D. Gavrila, F. Nabais, M. Pantic, P. Alvito, and D. Karreman, "The development and real-world deployment of frog, the fun robotic outdoor guide," in *HRI*, 2014.

[3] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore *et al.*, "Spencer: A socially aware service robot for passenger guidance and help in busy airports," in *FSR*, 2016.

[4] L. V. Calderita, A. Vega, S. Barroso-Ramírez, P. Bustos, and P. Núñez, "Designing a cyber-physical system for ambient assisted living: A use-case analysis for social robot navigation in caregiving centers," *Sensors*, 2020.

[5] D. J. Ahlgren and I. M. Verner, "Socially responsible engineering education through assistive robotics projects: The robowaiter competition," *IJSR*, 2013.

[6] T. Kanda, R. Sato, N. Saiwaki, and H. Ishiguro, "A two-month field trial in an elementary school for long-term human–robot interaction," *T-RO*, 2007.

[7] R. Kirby, *Social robot navigation*. Carnegie Mellon University, 2010.

[8] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *IROS*, 2014.

[9] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation," *IJRR*, 2015.

[10] C. Mavrogiannis, A. M. Hutchinson, J. Macdonald, P. Alves-Oliveira, and R. A. Knepper, "Effects of distinct robot navigation strategies on human behavior in a crowded environment," in *HRI*, 2019.

[11] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, 2013.

[12] M. Luber, L. Spinello, J. Silva, and K. O. Arras, "Socially-aware robot navigation: A learning approach," in *IROS*. IEEE, 2012.

[13] E. Torta, R. H. Cuijpers, and J. F. Juola, "Design of a parametric model of personal space for robotic social navigation," *IJSR*, 2013.

[14] I. R. Nourbakhsh, y. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion, "Human-robot teaming for search and rescue," *IEEE PERVAS COMPUT*, 2005.

[15] H. S. Park, J. S. Kang, H. S. Choi, S. W. Meng, and S. W. Kim, "Simulation-based interface testing automation system and method for robot software components," *US Patent 8,601,436*, 2013.

[16] A. Steinfeld, O. C. Jenkins, and B. Scassellati, "The oz of wizard: simulating the human for interaction research," in *HRI*, 2009.

[17] R. Madaan, N. Gyde, S. Vemprala, M. Brown, K. Nagami, T. Taubner, E. Cristofalo, D. Scaramuzza, M. Schwager, and A. Kapoor, "Airsim drone racing lab," *arXiv preprint arXiv:2003.05654*, 2020.

[18] "AI Habitat Challenge," aihabitat.org/challenge/2020, 2021-02-28.

[19] "AWS DeepRacer," aws.amazon.com/deepracer, 2021-02-28.

[20] "Crowdbot challenge," crowdbot.eu/crowdbot-challenge, 2021-02-28.

[21] K. Takaya, T. Asai, V. Kroumov, and F. Smarandache, "Simulation environment for mobile robots testing using ros and gazebo," in *ICSTCC*, 2016.

[22] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, M. Hutter and R. Siegwart, Eds., 2018.

[23] A. Hussein, F. Garcia, and C. Olaverri-Monreal, "Ros and unity based framework for intelligent vehicles control and simulation," in *ICVES*, 2018.

[24] B. Talbot, D. Hall, H. Zhang, S. R. Bista, R. Smith, F. Dayoub, and N. Sünderhauf, "Benchbot: Evaluating robotics research in photorealistic 3d simulation and on real robots," *CoRR*, 2020.

[25] A. Taivalsaari, T. Mikkonen, D. Ingalls, and K. Palacz, "Web browser as an application platform," in *SEAA*, 2008.

[26] B. Chen, H.-P. Hsu, and Y.-L. Huang, "Bringing desktop applications to the web," *IT Prof*, 2016.

[27] A. Pokle, R. Martín-Martín, P. Goebel, V. Chow, H. M. Ewald, J. Yang, Z. Wang, A. Sadeghian, D. Sadigh, S. Savarese *et al.*, "Deep local trajectory replanning and control for robot navigation," in *ICRA*, 2019.

[28] S. B. Banisetty and T. Williams, "Implicit communication through social distancing: Can social navigation communicate social norms?" in *HRI LBRs*, 2021.

[29] S. N. Woods, M. L. Walters, K. L. Koay, and K. Dautenhahn, "Methodological issues in hri: A comparison of live and video-based methods in robot to human approach direction trials," in *RO-MAN*, 2006.

[30] Q. Xu, J. Ng, O. Tan, Z. Huang, B. Tay, and T. Park, "Methodological issues in scenario-based evaluation of human–robot interaction," *IJSR*, 2015.

[31] N. Tsoi, M. Hussein, J. Espinoza, X. Ruiz, and M. Vázquez, "Sean: Social environment for autonomous navigation," in *HAI*, 2020.

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

[33] "Nvidia isaac sim," developer.nvidia.com/isaac-sim, 2021-02-28.

[34] A. Konrad, "Simulation of mobile robots with unity and ros: A case-study and a comparison with gazebo," Master's thesis, Department of Engineering Science, University West, 2019.

[35] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "FlightGoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," in *IROS*, 2019.

[36] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially compliant navigation through raw depth inputs with generative adversarial imitation learning," in *ICRA*, 2018.

[37] S. Curtis, A. Best, and D. Manocha, "Menge: A modular framework for simulating crowd movement," *Collective Dynamics*, 2016.

[38] A. Aroor, S. L. Epstein, and R. Korpan, "Mengeros: A crowd simulation tool for autonomous robot navigation," in *AI-HRI*, 2017.

[39] H. Bakri, C. Allison, A. Miller, and I. Oliver, "Virtual worlds and the 3d web–time for convergence?" in *iLRN*, 2016.

[40] S. Chernova, J. Orkin, and C. Breazeal, "Crowdsourcing hri through online multiplayer games," in *AAAI Fall Symposium Series*, 2010.

[41] P. Robinette, A. R. Wagner, and A. M. Howard, "Assessment of robot guidance modalities conveying instructions to humans in emergency situations," in *RO-MAN*, 2014.

[42] M. Nayyar, Z. Zoloty, C. McFarland, and A. R. Wagner, "Exploring the effect of explanations during robot-guided emergency evacuation," in *ICSR*, 2020.

[43] J. Xu and A. Howard, "How much do you trust your self-driving car? exploring human-robot trust in high-risk scenarios," in *SMC*, 2020.

[44] J. A. Złotowski, A. Weiss, and M. Tscheligi, "Navigating in public space: participants' evaluation of a robot's approach behavior," in *HRI*, 2012.

[45] L. Takayama, D. Dooley, and W. Ju, "Expressing thought: improving robot readability with animation principles," in *HRI*, 2011.

[46] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *HRI*, 2013.

[47] R. Toris, D. Kent, and S. Chernova, "The robot management system: A framework for conducting human-robot interaction studies through crowdsourcing," *JHRI*, 2014.

[48] R. Toris, J. Kammerl, D. V. Lu, J. Lee, O. C. Jenkins, S. Osentoski, M. Wills, and S. Chernova, "Robot web tools: Efficient messaging for cloud robotics," in *IROS*, 2015.

[49] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *CORL*, 2018.

[50] M. Gonzalez-Franco, E. Ofek, Y. Pan, A. Antley, A. Steed, B. Spanlang, A. Maselli, D. Banakou, N. Pelechano, S. Orts-Escolano *et al.*, "The rocketbox library and the utility of freely available rigged avatars for procedural animation of virtual humans and embodiment," *Frontiers in Virtual Reality*, 2020.

[51] K. Chen, J. P. de Vicente, G. Sepulveda, F. Xia, A. Soto, M. Vàzquez, and S. Savarese, "A behavioral approach to visual navigation with graph localization networks," in *RSS*, 2019.

[52] E. T. Hall, *The hidden dimension*. Doubleday, 1966.

[53] C. M. Carpinella, A. B. Wyman, M. A. Perez, and S. J. Stroessner, "The robotic social attributes scale (rosas) development and validation," in *HRI*, 2017.

[54] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*, 1988.

[55] E. Repiso, A. Garrell, and A. Sanfeliu, "Adaptive side-by-side social robot navigation to approach and interact with people," *IJSR*, 2019.

[56] G. Ferrer, A. G. Zulueta, F. H. Cotarelo, and A. Sanfeliu, "Robot social-aware navigation framework to accompany people walking side-by-side," *Autonomous robots*, 2017.