Preprint manuscript. The final publication is available at IEEE via http://dx.doi.org/10.1109/TCDS.2017.2787062

Two Demonstrators are Better than One - A Social Robot that Learns to Imitate People with Different Interaction Styles

Phoebe Liu, Dylan F. Glas, Takayuki Kanda, *Member, IEEE*, Hiroshi Ishiguro, Senior *Member, IEEE*

***Abstract*— With recent advances in social robotics, many studies have investigated techniques for learning top-level multimodal interaction logic by imitation from a corpus of human-human interaction examples. Most such studies have taken the approach of learning equally from a variety of demonstrators, with the effect of reproducing a mixture of their average behavior. However, in many scenarios it would be desirable to reproduce specific interaction styles captured from individuals. In this study, we train one deep neural network jointly on two separate corpuses collected from demonstrators with differing interaction styles. We show that training on both corpuses together improves performance in terms of generating socially appropriate behavior even when reproducing only one of the two styles. Furthermore, the trained neural network also enables us to synthesize new interaction styles on a continuum between the two demonstrated interaction styles. We discuss plots of the hidden layer activations from the neural network, indicating the types of semantic information that appear to be learned by the system. Further, we observe that the better performance with the synthesized corpus is not merely due to the increase of the sample size, as even with the same number of training examples, training on half the data from each corpus provided better performance than training on all the data from a single corpus.**

***Index Terms*— Human-robot interaction, learning by imitation, social robotics, service robots, proactive behaviors, learning interaction style.**

I. Introduction

In recent years, we have seen an upsurge of social robots being used commercially, specifically in the space of providing entertainment [1, 2], educating children [3, 4], or providing customer service [5, 6]. As more social robots gain traction in public, one promising approach for generating human-robot interaction logic is to automatically learn natural human behaviors by imitation from real human-human interaction. The arrival of the big data era reveals the feasibility of creating natural human-robot interactions empowered by data-driven approaches [7, 8, 9, 10]. Our previous work [7, 11], in which a shopkeeper robot learns to reproduce both reactive and proactive multimodal behaviors by means of abstraction from examples of human-human interactions, illustrates the idea that repeatable interaction data can be used to automatically infer interaction strategies for generating robot behaviors.

While data-driven approaches can be an efficient method for reproducing robot behaviors, one possible downside is that individual behavior differences from person to person may end up confusing a learning-based robot regarding what behaviors it should reproduce. Traditionally, most data-driven approaches require training a unique model for each task, and since each model may require thousands of examples [12, 13], this approach may not be so scalable when it comes to learning the range of variations of social behaviors that arise from person to person. For example, in a shop scenario, if a passive shopkeeper mainly let customers browse around, while on the other hand, a proactive shopkeeper took initiative to interact with the customer – which shopkeeper should a robot learn from, or is it possible for the robot to jointly learn from both shopkeepers, even when the shopkeeper behaves differently in the same situation?

Thus, this notion of learning social interactions from two different people instead of just a single person is an attractive option for a learning-based robot. Learning from two people provides more example training data, as well as the possibility to learn the different interaction style of each person. Both of these points are important, as having more training data could potentially improve the quality of the learned robot behaviors, and the possibility of learning multiple interaction styles can better equip the robot to assume different language behavior and interaction style depending on scenario or situation at the moment [14].

In this work, we will attempt to move from the paradigm of a robot learning from just one person to jointly learning from two people who may behave differently given the same situation (i.e. passive and proactive). Since our goal is to jointly

This work was supported by JST ERATO Ishiguro Symbiotic Human-Robot Interaction Project, Grant Number JPMJER1401 and in part by JSPS KAKENHI Grant Number 25240042.

The authors are with the Advanced Telecommunications Research Institute International Hiroshi Ishiguro Laboratories (e-mail: phoebeccliu@gmail.com; dylan.f.glas@gmail.com) and Intelligent Robotics and Communication Laboratories (e-mail: kanda@atr.jp), Kyoto 619-0288, Japan Hiroshi Ishiguro belongs to the Intelligent Robotics Laboratory, Osaka University, Toyonaka, Japan. Email: ishiguro@sys.es.osaka-u.ac.jp

learn from both shopkeepers and reproduce their behavior in a robot, we propose a simple modification to our previous learning system, which is to append a *style* feature to the input of the Multilayer Perceptron (MLP). In addition, we will investigate the effect of training jointly from both shopkeepers, demonstrating that it can both improve the performance of the robot behavior and equip the robot with the ability to assume different interaction styles via the *style* feature, due to the MLP learning shared neural representations for some semantically similar interaction patterns. Lastly, we discuss that the performance improvement when jointly learning from two shopkeeper corpuses is not just because of the increase of the sample size, as even with the same number of training examples, training on half the data from each corpus provided better performance than training on all the data from a single corpus.

II. Related work

A. Learning from data for social robots

For social robots, frameworks focused on crowdsourcing have been developed to enable learning of overall interaction logic from data collected from simulated environments, such as the Robot Management System framework [15] and The Mars Escape online game [8, 16]. Remote users are asked to interact in order to complete several search and retrieval tasks in an online game. The interaction data are logged and used to generate autonomous robot behaviors to also complete the same task. In Thomaz et al's work, they developed a framework to enable other online users to administer feedback when teaching a Reinforcement Learning agent to perform tasks observed in a game [17, 18]. While our work complements these approaches by considering crowd-based data collected directly from human-human interaction using sensors in a physical environment, we are also interested in capturing individual style from humans and reproducing the differing styles in a robot.

The use of real human interaction data collected from sensors for learning interactive behaviors has been investigated in numerous works. In a study by Nagai et al., a robot was developed with infant-like ability to learn from human parental demonstrations by using a model based on visual saliency to detect likely important locations in a scene without employing any knowledge about the actions or the environment [19]. The robot JAMES was developed to serve drinks in a bar setting, in which a number of supervised (i.e. dialog management) and unsupervised learning techniques (i.e. clustering of social states) were applied to learn social interaction [20]. Admoni and Scassellati proposed a model that uses empirical data from annotated human-human interactions to generate nonverbal robot behaviors in a tutoring applications. The model can simultaneously predict the context of a newly observed set of nonverbal behaviors, and generate a set of nonverbal behaviors given a context of communication [21]. While some of these studies do support learning interactive behaviors, we are not aware of any framework designed to simultaneously learn from human demonstrators who exhibit distinctively different behavior styles, in terms of both verbal expression and nonverbal motion, given the same situation.

B. Learning from multiple sources

In robot manipulation tasks, most works have focused on learning a task-specific behavior [22, 23]. There have been some attempts to move from learning a task-specific model to jointly learning multiple robot tasks at the same time. Pinto and Gupta demonstrated how models with multi-task learning (i.e. grasp and push) tend to perform better than a task-specific model with the same amount of data [24]. They hypothesized that performance improvement is due to diversity of data and regularization in learning. Likewise, our study also considers the merit of jointly learning interactions from multiple people as a scalable solution for the robot to improve performance as well as learning different interaction styles at the same time.

There have been some attempts to acquire verbal and non-verbal dialog behaviors for a robot learned from multiple demonstrators. In Leite et al.'s study, they proposed a semi-situated learning method to crowdsource from multiple authors, resulting in a dataset consisting of a set of annotated, human-authored dialog lines that are associated with the goal that generated them [25]. Their system blends together content created by multiple authors and rated by multiple judges to be used for generating robot speech. In contrast to their work, where input from multiple authors is manually created and merged together, our work learns directly from data and aims to preserve and reproduce the individual styles of the demonstrators.

In the regime of natural language processing, work with Long Short Term Memory (LSTM) neural networks has demonstrated [26] that translation to multiple languages is possible from one source language, an approach which also showed benefits such as better training efficiency and smaller number of models. The translation task of generating honorific language from English to German was made possible by training with two sources of informal and polite German speech [27]. Simultaneously, word-graphs were constructed by using tweets collected from two different domains (i.e. politics and entertainment) to transform regular chatbot responses to the responses which mimic the speaking styles of those specific domains [28]. Similarly, we also want to jointly learn different interaction styles from two different shopkeepers, but in the problem domain of learning multimodal human-robot interactions from noisy sensor data collected in a physical environment.

III. Data Collection

A. Scenario

We chose a camera shop scenario for this study so that repeatable behaviors consistent with either a proactive or passive interaction styles could be observed. We set up a simulated camera shop environment in an 8m x 11m experiment space with three camera models on display, each at a different location (Fig. 1). For each interaction, one shopkeeper participant interacted with one customer participant. In this environment, our goal was to collect data corresponding to the following two shopkeeper behavior patterns:

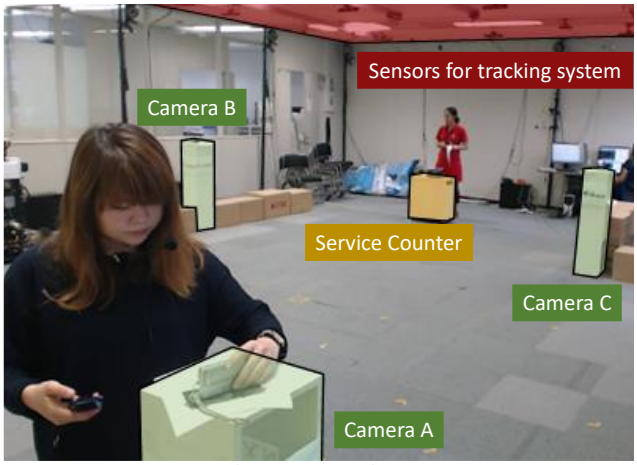


Fig. 1. Environment setup for our study, featuring three camera displays. Sensors on the ceiling were used for tracking human position, and smartphones carried by the participants were used to capture speech.

- A *proactive* shopkeeper takes initiative, either by introducing new camera features or presenting a new camera to the customer, while still answering the customer's questions.
- A *passive* shopkeeper is preoccupied with other tasks in the shop and mostly lets the customer browse around the shop, though the shopkeeper should still be helpful by answering the customer's questions.

We chose these two interaction styles because we consider them to be particularly meaningful in HRI. For example, Baraglia et al. [29] have also investigated the importance of controlling a robot's level of proactivity in collaborative tasks.

B. Sensors

We recorded the participants' speech and movement as they interacted with each other. We used a human position tracking system, consisting of 20 Microsoft Kinect 1 sensors arranged in rows on the ceiling, to capture the participants' positions and motion in the room. Particle filters were used to estimate the position and body orientation of each person in the room based on point cloud data [30].

The speech of each participant was also captured by a handheld smartphone, and the Google speech recognition API[1] was used to recognize utterances and send the text to a server via Wi-Fi. To detect the start and stop of speech activity, users were required to touch the mobile screen to indicate the beginning and end of their speech.

Location data for the shopkeeper and the customer were recorded at a rate of 20 Hz. Speech data were recorded at the start and end of each speech event, as signaled by participants tapping on their Android phones.

C. Participants

For the role of customers in our interaction, we recruited fluent English speakers as participants. They had varied levels of knowledge about cameras. We employed a total of 18 customer participants (13 male, 5 female, average age 32.8, s.d. 12.4).

TABLE I. AMOUNT OF DATA COLLECTED

Shopkeeper	Number of Trials	Shopkeeper Utterances	Customer Utterances
Passive	206	1638	1940
Proactive	199	2568	2299

Since our goal was to capture the natural interaction styles of the shopkeepers, we initially interviewed and recruited participants with various degrees of proactivity as shopkeepers and observed some trial interactions. After the trial interactions, we asked the customer participants to provide feedback on the shopkeepers in terms of how well they fit the descriptions for the target interaction styles. Thus, based on the interview and feedback results, we selected one participant (male, age 54) with a naturally outgoing personality, as the proactive shopkeeper. Comparably, we selected another participant (female, age 25), who had a quieter disposition, as the passive shopkeeper. They played the assigned roles in all interactions.

D. Procedure

The shopkeepers were encouraged to act according to their type (i.e. passive or proactive), as described above. Both shopkeepers were instructed to wait by the service counter at the start of the interaction. They were also instructed to be polite, and to give socially-appropriate acknowledgements (i.e. greetings and farewells).

To keep interactions interesting and to create variation in the interactions, customer participants were encouraged to play with the cameras and asked to role-play in different trials as advanced or novice camera users, and to ask questions that would be appropriate for their role. Some camera features were chosen to be more interesting for novice users (color, weight, etc.) and others were more advanced (High-ISO performance, sensor size, etc.), although they were not explicitly labeled as such. Customer participants were not given a specific target feature or goal for the interaction, as we were mostly interested in capturing the shopkeepers' behavior.

All participants were instructed to focus their discussion on the features listed on the camera spec sheet, ranging from 8 to 10 features for each camera, to minimize the amount of "off-topic" discussion.

Each shopkeeper interacted with 9 different customer participants. For each customer participant, we conducted 24 interactions each (12 as advanced and 12 as novice) for a total of 216 interactions. 17 interactions were removed for proactive shopkeeper trials and 10 interactions were removed for passive shopkeeper trials, due to technical failures of the data capture system or participants who did not follow instructions. Table 1 presents the amount of data collected. This data set is available online[2].

E. Observed Behavior

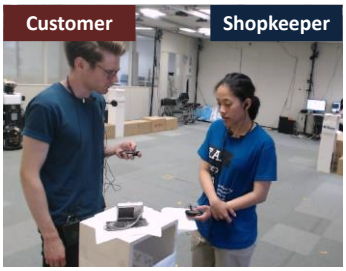
Overall, the customer participants followed our suggestions, though some customers have difficulty role-playing advanced

[1] https://www.google.com/intl/en/chrome/demos/speech.html

[2] http://www.geminoid.jp/dataset/camerashop/dataset-camerashop.htm

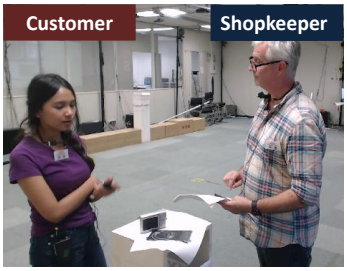
TABLE II. AN EXAMPLE INTERACTION FROM THE DATA COLLECTION (C: CUSTOMER, S: SHOPKEEPER)

Interaction with a passive shopkeeper



(Shopkeeper and Customer talking about Nikon)
C: So you would recommend me this for holiday or traveling?
S: Yes this is the perfect camera for traveling.
C: Thank you I will look at another camera before I decide.
S: Okay.
(Shopkeeper moves back to Service Counter, Customer moves to Sony)
C: Sorry excuse me.
S: How can I help you? (moves to Sony)
C: How much is this camera?
S: that one is $550.
C: this is very expensive.
S: *(Silence for 9 seconds)*
C: I will just look around…

Interaction with a proactive shopkeeper



(Shopkeeper and Customer talking about Sony)
C: I'm looking for something lightweight but good quality.
S: This Sony camera takes pictures that are almost as good as a top-end professional camera but it only has half the weight, please pick it up and have a try.
C: Yeah actually this weighs alright... how much is it?
S: This is only $550 so it's a quarter to a fifth of the price of a high-end professional camera.
C: *(Silence for 5 seconds)*
S: the picture quality is almost as good as DSLR because it's a mirrorless camera …

or novice camera users – for example, participants who had little knowledge about cameras were not easily able to think of the types of questions an advanced camera user would ask. Aside from this point, we observed a variety of behaviors captured, such as customers who spoke multiple topics in a single, long utterance and customers who only had direct questions. Since we encouraged the customers to play with different cameras, we observed that at times, a customer would be focused on a camera and would not speak or move for some time, thus creating a period of silence during the interaction.

The shopkeeper participants behaved according to their assigned roles. For the passive shopkeeper, she mainly let the customer browse around the shop and only answered questions when asked. She gave short, concise answers and did not expound on her answers. In contrast, the proactive shopkeeper had much more variation in his responses, and he often spoke in long, descriptive utterances and volunteered extra information when answering questions.

Here we describe four main differences we observed between the behaviors demonstrated by the two shopkeepers. First, the proactive shopkeeper approached the customer when he or she entered the shop, whereas the passive shopkeeper waited by the service counter. Second, the proactive shopkeeper often explained about 2 or 3 features at the same time, whereas the passive shopkeeper usually only explained one feature at a time. Third, the proactive shopkeeper often volunteered more information, either by talking about a new feature or continuing his previous explanation, after some silence had elapsed or when the customer demonstrated a "backchannel" utterance (e.g. "oh, ok"). In this situation, the passive shopkeeper would usually remain silent. Fourth, the proactive shopkeeper would ask the customer questions, such as 'what sort of pictures do you take?", whereas the passive shopkeeper rarely asked the customer questions.

Table 2 illustrates example interactions from the both the passive and proactive shopkeeper. Notice that the passive shopkeeper is quite reactive, and her responses are usually short and concise. She also moves back to the service counter or remains silent when the customer does not inquire about a camera. On the contrary, the proactive shopkeeper presents additional information about the camera, both when the customer asks a question and when the customer remains silent.

IV. PROPOSED TECHNIQUE

A. Overview

In order to reproduce proactive or passive interaction styles in a robot, we used a collection of data-driven techniques that directly learn behaviors (i.e. utterances and motion) from examples of human-human interaction from noisy sensor data. These techniques closely follow the procedure followed in our previous work [7, 11], and additional details are presented in the Appendix. The key steps of the techniques are listed here:

1. **Abstraction of training input and typical robot actions** (Sec. IV.B): Continuous streams of interaction data captured from sensors are abstracted into typical behavior patterns, and the corresponding *joint state vector* and *robot action* are defined.
2. **Learning with Multilayer Perceptron (MLP) Neural Network** (Sec. IV.C): We applied a feed-forward MLP neural network to learn to reproduce robot behaviors. An "attention" layer is applied to the neural network to learn the relative importance of various steps of interaction history as inputs to the respective robot output actions.
3. **Adding a target "interaction style" constraint** (Sec.

IV.D): In order to learn different interaction styles, this work extends the previous system by appending an extra token to the input of the neural network. The token is initialized to correspond to the respective human shopkeeper from the training examples. At runtime, it can be used to specify whether the outputted target robot action should mimic the interaction style of the proactive or passive shopkeeper.

The techniques for Steps 1 and 2 were presented in our previous study [7, 11], while Step 3 constitutes the novel contribution of this work which enables behavior generation for multiple interaction styles.

B. *Abstraction of training input and target robot action*

In order to learn effectively despite the large variation of natural human behaviors and noisy inputs from the sensor system, the sensor data needs to be abstracted into common behavior patterns (i.e. common spatial states and common spoken utterances), which are then used to discretize a continuous stream of captured sensor data into behavior events. Here we briefly describe our techniques:

- Abstraction: To find common, typical behavior patterns in the training data, we used unsupervised clustering and abstraction to identify typical utterances, stopping locations, motion paths, and spatial formations of both participants in the environment.
- Action Discretization: To discretize continuous sensor data, we identified an action whenever a participant: (1) speaks an utterance (end of speech), and/or (2) changes their moving target, and/or (3) yields their turn by allowing a period of time to elapse with no action. An interaction is discretized into a sequence of alternating customer and shopkeeper actions.
- Defining Input Features: For each action detected, the abstracted state of both participants at the time is represented as a *joint state vector*, with features consisting of their abstracted motion state and the utterance vector of the current spoken utterance.
- Incorporating History: To provide contextual information for generating robot shopkeeper actions, the *n* most recent joint state vectors are incorporated as interaction history. We chose *n* = 3 since this seemed to be a good balance for generating observed shopkeeper behaviors (e.g. presenting new features) in our scenario. This interaction history constitutes the input to our learning mechanism.
- Defining robot action: The subsequent shopkeeper action to the interaction history is mapped to a robot action, consisting of a typical utterance (e.g. ID 5) and a target spatial formation (e.g. *present Nikon*). The number of typical utterances is obtained from hierarchical clustering, further detailed in the Appendix. When executed, this would cause the robot to speak the typical utterance, "It's $68", associated with utterance ID 5, and execute a motion to attain the formation of *present Nikon*. This robot action is used as the training target for our learning mechanism.

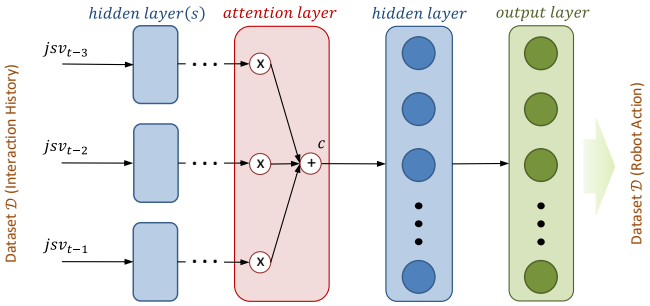


Fig. 2. Schematic of the multilayer perception neural network. Interaction history is inputted to the neural network as joint state vectors and robot action as training target for the neural network.

C. *Learning with Multilayer Perceptron Neural Network*

We are interested in automatically generating robot actions using only data observed from human-human interaction. To achieve this, we applied a multilayer perceptron neural network, which has the ability to learn the representation and the mapping between our training data and how it best relates to a robot action. It attempts to generalize class assignments from examples in a dataset $\mathcal{D}$. Our dataset is composed of $(X, robot\ action)$ interaction pattern pairs, where $X \in \mathbb{R}^{3m}$ is an interaction history consisting of the three most recent *joint state vectors*, $X = \{jsv_{t-3}, jsv_{t-2}, jsv_{t-1}\}$, and *robot action* $\in \{0,1\}^d$ is a target class assignment where d is equal to the number of possible robot actions. That is, if $robot\ action_i = 1$, observation X maps to robot action i.

Computation in the neural network is performed by artificial neurons, which are typically organized into layers. The activation value of neuron j in layer l is defined in (1) as

$$a_j^{(l)} = \sigma\left(\sum_k w_{j,k}^{(l)} \cdot a_k^{(l-1)}\right) + b_j^{(l)}) \quad (1)$$

where $b_j^{(l)}, w_{j,k}^{(l)} \in \mathbb{R}$ are parameters optimized by the network using the backpropagation algorithm, $a_k^{(l-1)}$ is the activation (output) of neuron k in layer $l-1$, and σ is a nonlinear activation function.

To learn context-dependent robot actions, we also included an attention layer in our neural network, as proposed by Raffel and Ellis [31], which has the ability to "attend" and learn which parts of the interaction history are important when predicting robot behaviors. The idea is once we have an activation value of neuron j in layer l, $a_j^{(l)}$, we can query each value asking how relevant they are to the current computation of the target class assignment. $a_j^{(l)}$ then gets a score of relevance which can be turned into a probability distribution that sums up to one via the softmax activation. We can then extract a context vector that is a weighted summation of the activation value in layer l depending on how relevant they are to a target robot action.

Fig. 2 shows the schematic of the neural network, which is identical to the architecture of our previous system [11]. It is composed of three sets of input neurons of size m from the joint state vectors in the interaction history, followed by two leaky rectified hidden layers, an attention layer, and another leaky rectified hidden layer. The output layer is a softmax with the

number of neurons equal to the number of possible robot actions, which represents the probability of a robot action given an interaction history input.

D. Adding style feature

To capture the differences in shopkeeper behavior from the human-human interactions, we propose one simple modification to our previous system, which is to introduce an extra *style* feature to the joint state vectors. The idea is that this *style* feature specifies which shopkeeper behaviors the current interaction corresponds to, which is provided to the neural network as an additional input feature.

At training time, the correct *style* feature is set based on the source of the human-human interaction data, that is, *style* is set to be <Proactive> if the training data came from the proactive shopkeeper and <Passive> if the training data came from the passive shopkeeper. This attribute is then concatenated as a *style* feature to the *joint state vector*. Next, we trained the neural network with the interaction data combined from both shopkeepers. When used in online operation, we assume that the *style* feature will be specified by a user who selects the desired level of proactivity in the robot action.

While one could envision alternative architectures to incorporate the *style* feature (e.g. directly connecting it to all hidden layers or connecting it to an output layer [32]), we consider the addition of an input feature to the neural network to target a desired interaction style to be a simple and elegant approach, as it requires no modification to the existing architecture of the neural network. Similar approaches have been applied and shown to be effective in neural translation systems – for instance, an artificial token was introduced in the input sentence of the source language to specify the required target language trained from multiple languages [26] and a *side constraint* was added to the source text to control the level of honorific for English to German translation [27].

In summary, we consider this proposed *style* feature to be a simple and elegant solution to incorporate training data representing differing interaction styles into the learning system. During training, we only need to add one additional feature to each *joint state vector* in the interaction history to preserve the observed interaction style, and during operation, we can control the *style* feature to specify a desired target shopkeeper interaction style.

V. Offline Evaluation

In this section, we evaluate our proposed system with the neural network using the additional *style* feature by training the system with the two conditions: (1) *baseline* condition: separately with only data from either the passive shopkeeper or proactive shopkeeper, (2) *proposed* condition: combining the training data from both the passive and proactive shopkeeper. Our goal is to confirm that our system enables the robot to become more socially appropriate when trained from the combined data as opposed to separate data, while still able to preserve the interaction style of the human shopkeepers.

Our baseline condition is the system trained on separate data from either the passive or proactive shopkeeper, since the previous studies [7, 11] already confirmed that the system is capable of learning socially-appropriate behaviors from a single shopkeeper. Our proposed condition is the system trained on the combined dataset from both the passive and proactive shopkeeper, as we want to evaluate whether our extended system is capable of learning jointly from shopkeepers with different interaction styles.

A. Evaluation

To evaluate the performance between the combined and separate conditions, we performed cross-validation, by first randomly selecting approximately 10% of customer-shopkeeper-customer behavior sequences from the dataset as test data, and using the other 90% of the dataset to train the neural network. In order to ensure a fair comparison between the two conditions, we chose an equal number of test examples representing the interaction with the passive and proactive shopkeeper for evaluation for both conditions. Thus, the total number of acceptable behaviors (i.e. behavior correctness) from the separate datasets is compared with the total number of acceptable behaviors from the combined dataset.

Training parameters: All conditions were trained on the same neural network architecture described in Sec. 4.3. The input dimension to the neural network is $3m$, where m is 1248, including the *style* feature. The number of neurons for each hidden layer is 800.

Since the activation value of the output layer represents the conditional probability of a robot action given an interaction history input, the number of neurons in the output layer will thus be dependent on the number of possible robot actions. This number will vary depending on which training dataset is used (for example, the passive shopkeeper used less variation in her utterances than the proactive shopkeeper). For a given dataset, the number of possible robot actions, c, is determined based on the number of utterance clusters and spatial states coming from that dataset. Thus, c was 711 for the proactive data set, 509 for the passive data set, and 912 for the combined data set.

The training was performed by momentum-based mini-batch stochastic gradient descent, with a batch size of 128, a learning rate of 0.005, and a momentum coefficient of 0.9. Normalized initiation, described by [33], was used to initialize the neural network. The network was trained to minimize the cross entropy loss for 2000 epochs between the observed target robot action and the predicted robot action for the entire training set.

Evaluation Procedure: To evaluate whether our system performs better when trained from two different shopkeepers versus a single shopkeeper, we evaluated the "social appropriateness" of the predicted behaviors, rather than using exact prediction accuracy. There could be numerous robot actions that could all be equally valid for a given input. For example, when a customer asks about the camera price, responding with "$2000", "it's only $2000", and "the camera body is only $2000", could all be considered equally valid answers. This approach is similar to the procedure used in [7] for evaluating the appropriateness of robot behaviors.

To measure social appropriateness of the behaviors, we asked

TABLE III. RESULTS OF MANUALLY-CODED CROSS-VALIDATION COMPARISON. THE RESULT OF THE SYSTEM TRAINED WITH THE COMBINED DATA SHOWED A SIGNIFICANT DIFFERENCE WHEN COMPARED WITH THE DATASET THAT IS ONLY TRAINED ON THE INDIVIDUAL PASSIVE AND PROACTIVE DATASETS.

Condition	Training set	# of training examples	Test set	# of test examples	# of acceptable predicted behaviors	Behavior Correctness	*p* value	Kappa
Baseline (Separate)	Passive	2142	Passive	225	166	73.8%		
	Proactive	2223	Proactive	225	138	61.3%		
	Total			**450**	**304**	**67.6%**		**0.764**
Proposed (Combined)	Combined	4365	Passive	225	190	84.4%		
	Combined	4365	Proactive	225	176	78.2%		
	Total			**450**	**366**	**81.3%**	**<.001**	**0.718**

a human coder, naïve to the experimental conditions, to manually rate the acceptability of each prediction as "acceptable" or "unacceptable". Unacceptable behaviors included factually incorrect responses, failures to answer a question, strange behaviors like moving away to a new camera while a person was waiting for a response, and repetition of the previous shopkeeper behavior if not appropriate to do so. The evaluations were made based on transcripts of the collected dataset. Each coder was shown an interaction history of customer-shopkeeper-customer actions, where the utterances were automatically transcribed using ASR, and motion abstracted using techniques described in the Appendix.

Because the two shopkeepers behave in different ways, we also consider some behaviors to be appropriate for one shopkeeper but not for another. For example, the passive shopkeeper remains silent during "backchannel" while the proactive shopkeeper will volunteer more information. For this reason, we showed some behaviors demonstrated by the different shopkeepers from the training interaction to the coder, and asked them to keep in mind whether the predicted behavior was consistent with the interaction style of the shopkeeper. Predicted behaviors were marked as "unacceptable" if they were not consistent with the interaction style of the shopkeeper.

As the behavior appropriateness ratings require subjective judgment, we confirmed the consistency of the coder's evaluations by asking a second coder to independently rate the same data set. We checked the consistency between their evaluations by calculating Cohen's Kappa, which was 0.764 for the baseline (separate) condition, and 0.718 for the proposed (combined) condition. We consider this to be good interrater agreement. For the following analyses, when the coders had conflicting "acceptable" and "unacceptable" ratings, only the rating from the first coder was used.

B. Results of system performance in terms of behavior correctness

The results of the cross-validation comparison are shown in Table 3. The overall behavior correctness was 67.6% in the *baseline* condition, and 81.3% in the *proposed* condition.

To evaluate the statistical significance of differences between the baseline and proposed condition, a chi-squared test was performed. The chi-squared test showed significance, (χ^2 (1, N=450) = 25.737, p < .001) indicating that the *proposed* condition resulted in significantly better performance than the *baseline* condition.

C. Predicted behaviors between the two conditions

It is interesting to note that the *proposed* condition resulted in much better performance than in the *baseline* condition, as this indicates that jointly training on both the passive and proactive dataset will enhance the performance of the robot behaviors. We speculate that the system may have the capability to automatically uncover similarity between the training examples of the passive and the proactive shopkeeper, in spite of the shopkeepers' behaviors being quite different for the same given input. With such learning capability, the system would benefit from the increase in data samples provided from combining the data, and robot actions that were not well learned due to lack of repeatable training examples in the *baseline* condition would become better learned in the *proposed* condition.

One area of improvement in the *proposed* condition was in handling cases where the customer is comparing between two cameras. For example, when the customer compares the price between cameras (e.g. "it's bit pricey anything cheaper?"), the *baseline* condition did not predict introducing a cheaper camera, but instead incorrectly answers about the current camera's weight. Conversely, faced with the same situation in the *proposed* condition, the system predicts correctly by introducing a camera with a cheaper price. Upon inspection of the training data, we found that there were only 107 times when the customer compares different cameras during interactions with the passive shopkeeper and 67 times during interactions with the proactive shopkeeper. By combining the training data, the number of training examples for comparing between two cameras increases, enabling the system to better learn such behaviors.

D. Results for reproducing distinct interaction style

While we have demonstrated that the system performs better when combining the data from two shopkeepers, a second, and equally important, question is whether the system can also reap the benefit of reproducing the different interaction styles of the shopkeepers when jointly learning from two shopkeepers. Particularly, we are interested whether the predicted behaviors in the *proposed* condition will also follow the general trend of those observed in the training examples of human-human interaction.

In the human-human interactions, the proactive shopkeeper was more verbose with the customers and spoke an average of 17.61 words per turn, whereas the passive shopkeeper spoke only 6.32 words per turn. Each human utterance is mapped to a

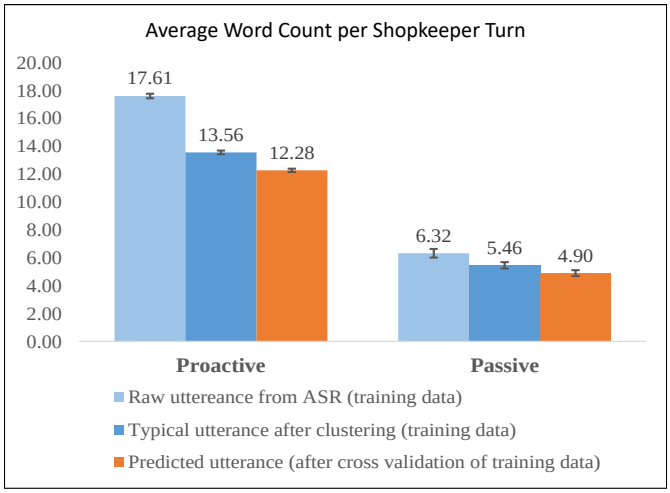


Fig. 3. Average Word Count per Shopkeeper Turn for the Proposed System. The predicted utterance follows the trend of the training data: the proactive shopkeeper is more verbose than the passive shopkeeper. The bar indicates standard error.

cluster, and a "typical utterance" from that cluster is used as the robot utterance. By its nature, the algorithm for selecting typical utterances tends to choose shorter utterances, so the average length of a typical robot utterance is shorter than the actual utterances from the training data. That is, 13.56 words for the proactive case, and 5.46 for the passive case.

Since the training data showed that the proactive shopkeeper was more verbose than the passive shopkeeper, we would also expect a more verbose robot with <Proactive> *style* than with <Passive> *style*. As shown in Fig. 3, this was indeed the case, and the average word count per predicted utterance was 12.28 with <Proactive> *style* and 4.90 with <Passive> *style*, suggesting that our *proposed* condition was able to preserve the different interaction styles of the human shopkeepers. We consider this to be a good indication that the learning system successfully captured the respective interaction styles of the shopkeepers in the training examples.

One example that demonstrates interaction style is being preserved in the *proposed* condition is when the customer says a "backchannel" utterance (e.g. "I see", "wow that's light") or remains silent. Frequently, the system predicts the robot should remain silent with <Passive> *style*, but predicts a proactive behavior with <Proactive> *style* (e.g. "Would you like to take a couple pictures with at first"). Another example is when the customer asks a question: the system tends to predict short concise answers (e.g. "this one has of 30 seconds of long exposure") with <Passive> *style* and predict longer, more detailed answers (e.g. "the shutter speed can we [sic] set up to 30 seconds if you want to go faster than that it is not possible with this camera") with <Proactive> *style*.

E. Examples of Weighted Style Feature Prediction

Having a mechanism to generate robot shopkeeper behaviors using an additional *style* feature made us think about what happens when the learning system is provided not with a discrete *style* value but with a continuous value between two target *style* features. Instead of feeding the neural network either a <Passive> or <Proactive> *style* feature, what happens if we feed it a linear combination of (1-*w*)<Passive> + *w*<Proactive>? One expectation is that the model will gradually shift from predicting a passive shopkeeper behavior to predicting a proactive shopkeeper behavior as *w* increases, allowing us to intentionally blend between styles.

In Table 4, we demonstrate several examples of the learning system predicting behaviors with different weights of *w*. In the first example (Fig. 4) where the customer enters the shop, the system predicts that the robot should wait at the service counter for weight values up to $w = 0.8$ before finally predicting that the robot should approach the customer when $w = 1.0$. The second example is a question about exposure. The predicted action is "It goes up to 30 seconds of exposure" when $w <= 0.4$, and predicts a more longer and detailed answer as *w* increases beyond 0.6. Likewise, in the next two examples, when *w* is small, the predicted action follows the typical behavior of the passive shopkeeper, and when *w* is larger, actions corresponding to the proactive shopkeeper's behavior are predicted. It is interesting to note that the system predicted two different robot actions at $w = 0.4$ and at $w = 0.8$. Perhaps the intermediate action had been observed from both shopkeepers in the training data.

As the above examples illustrate, the shift in the predicted robot action from passive to proactive style does not always happen at a single threshold *w* value, so the robot could be expected to gradually exhibit a larger fraction of proactive behaviors as *w* increases. We find the possible implications of this result interesting, since it suggests the ability to "interpolate between personalities" and fine-tune the desired interaction style of the robot during run-time. Such controllability could provide the robot with a more adaptive interaction style depending on the user's collaboration preferences for a passive or proactive robot [29].

F. Visual Analysis

The result of the system – that training a model across data from multiple shopkeepers can enhance performance of the robot action prediction – raises the question of whether the network is learning some sort of shared neural representation, in which interaction examples with the same semantic meaning are represented in similar ways regardless of differences in shopkeeper interaction style.

One way to study the representations used by the network is to visualize the activation values. The activations of the hidden layer can be seen as an alternative (learned) representation of the input observation, given that the activation of layer l depends only on learned parameters and the activation of layer $l-1$ (see Fig. 2). For our study, we used a variant of t-distributed Stochastic Neighbor Embedding (Barnes-Hut t-SNE) [34] to project high-dimensional activation values from the last hidden layer for visualization in 2D space, due to its ability to preserve neighborhoods in projections.

What the untrained MLP knows: We first consider the untrained MLP with input and class assignment from *combined* data, and initialized according to Sec. IV.C. As shown in Fig. 5(a), the projection of the last hidden layer of the neural network

Fig. 4. When w = 0.0, the robot waits by the service counter as customer enters the shop. When w = 1.0, the robot approaches the customer as she enters the shop.

TABLE IV. SEVERAL EXAMPLES OF GRADUALLY SHIFTING THE WEIGHT W FROM 0 TO 1 TO REPRODUCE PASSIVE TO PROACTIVE BEHAVIORS.

Interaction History	w	Predicted robot action	Generated robot speech
Customer enters the shop			
C: (*Enters the shop and moves to Sony*) S: (*Waiting at service counter*) C: (*moves from Sony To Nikon*)	0.0	0-Waiting	*(Waiting at service counter)*
	0.2	0-Waiting	*(Waiting at service counter)*
	0.4	0-Waiting	*(Waiting at service counter)*
	0.6	0-Waiitng	*(Waiting at service counter)*
	0.8	0-Waiting	*(Waiting at service counter)*
	1.0	144-Present Nikon	Good afternoon how can I help? (*approach customer at Nikon*)
Q&A			
(Both at Canon) C: What about this one with the ISO? S: The ISO here is 6400 it takes excellent shots in the evening but then you have full creative control with manual shutter settings. C: oh that sounds good how long does the exposure go?	0.0	271-Present Canon	It goes up to 30 seconds of exposure.
	0.2	271-Present Canon	It goes up to 30 seconds of exposure.
	0.4	271-Present Canon	It goes up to 30 seconds of exposure.
	0.6	190-Present Canon	The standard shutter exposure settings go up to 30 seconds but then it has a bulb mode so you can stay open for as long as you want.
	0.8	190-Present Canon	The standard shutter exposure settings go up to 30 seconds but then it has a bulb mode so you can stay open for as long as you want.
	1.0	190-Present Canon	The standard shutter exposure settings go up to 30 seconds but then it has a bulb mode so you can stay open for as long as you want.
Turn Yield			
(Both at Canon) C: [Silent] S: This is the Canon 5d Mark 3 it gives you full creative control so that gives you full exposure control. C: [Silent]	0.0	542-Present Canon	If you need anything else I'll be at the service counter.
	0.2	542-Present Canon	If you need anything else I'll be at the service counter.
	0.4	288-Present Canon	It has 61 auto focus points that can be combined in a variety of autofocus modes
	0.6	288-Present Canon	It has 61 auto focus points that can be combined in a variety of autofocus modes
	0.8	288-Present Canon	It has 61 auto focus points that can be combined in a variety of autofocus modes
	1.0	288-Present Canon	It has 61 auto focus points that can be combined in a variety of autofocus modes
Turn Yield			
(Both at Nikon) C: [Silent] S: How did you find the weight, it's only 120 grams. C: Yeah I think it's good, it's very light.	0.0	0-Present Nikon	[Silent]
	0.2	0-Present Nikon	[Silent]
	0.4	667-Present Nikon	It has 18 preset modes and it has for instance beach mode and snow mode.
	0.6	667-Present Nikon	It has 18 preset modes and it has for instance beach mode and snow mode.
	0.8	205-Present Nikon	It has a zoom lens as well so if you want that little bit of extra help to make the frame look a little tidier it can do that.
	1.0	205-Present Nikon	It has a zoom lens as well so if you want that little bit of extra help to make the frame look a little tidier it can do that.

before training shows the activation values. The color and shape of the projection represent the shopkeeper type (i.e. passive versus proactive) and spatial formation from the inputs. We see that the projection is clearly separated by the shopkeeper type and spatial formation, which is unsurprising since these features are explicitly represented in the *joint state vector*. This seems to indicate that the untrained MLP treats $(X, robot\ action)$ interaction pattern pairs separately by shopkeeper type and spatial formation.

Training effects: We have observed that the system performed better after training with *combined* data, thus we naturally hypothesize whether the MLP learns some shared neural representation despite the different shopkeeper type or spatial formation. To study this hypothesis, we projected the activation values of the last hidden layer *after* training the network. As shown in Fig. 5(b), we see the projected values converged despite different shopkeeper type or spatial formation. Hence, it is natural to assume that the learning process arrived at an alternative representation of the data that captures some sort of higher-level features [35], which are reflected by the projection.

Understanding higher-level features: A logical next question is understanding the meaning of the higher-level features. We try to find answers by looking at the interaction pattern pairs themselves. The human-human interactions mainly consist of several distinct interaction patterns, for

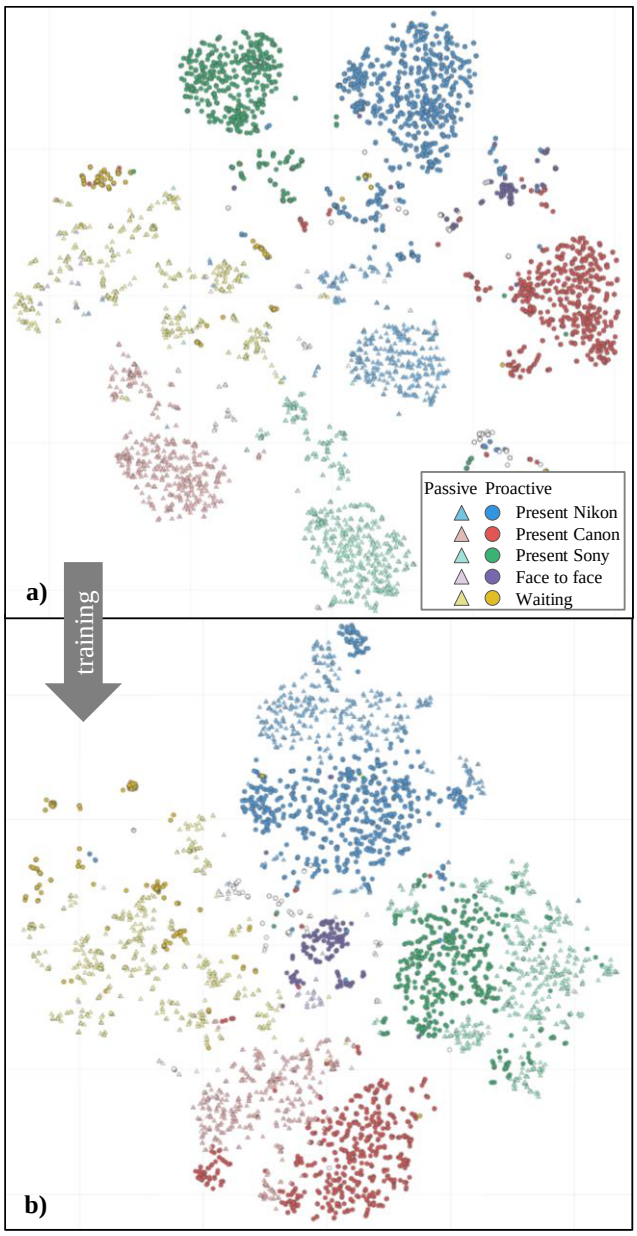


Fig. 5. Projection of the last MLP hidden layer activations for combined data (a) *before* training the neural network, (b) *after* training the neural network.

instance, question and answer. Thus, in order to understand the nature of our data and how it relates to the neural projection, we had a single coder manually annotate our combined dataset into the following three categories:

- Q&A: When customer asks a question to the shopkeeper that is defined by our scenario. In additional, we annotated which camera feature the Q&A is related to.
- Turn yield: When the customer decides to yield his turn by being silent, saying a "backchannel" utterance (e.g. "I see", "that sounds good"), or answers a shopkeeper question (e.g. "yes").
- Ending interaction: When the customer decides to end an interaction before leaving the shop or wants to browse around by himself (e.g. "thank you very much for your help" or "I'm just looking around at the moments thanks"), followed by the shopkeeper's acknowledgment.

TABLE V. SAMPLE INTERACTION PATTERNS FROM ANNOTATION FOR Q&A, TURN YIELD, AND ENDING INTERACTION. EXAMPLES FROM PASSIVE SHOPKEEPER ARE (PA) AND EXAMPLES FROM PROACTIVE SHOPKEEPER ARE (PRO). FOR BREVITY, ONLY THE MOST RECENT CUSTOMER UTTERANCE AND THE CORRESPONDING SHOPKEEPER UTTERANCE ARE SHOWN.

Q&A	
(Pa)	C: and how about the exposure? S: you can hold exposure for up to 30 seconds.
(Pro)	C: can you do long exposure shoot? S: not pass 30 seconds I'm sorry for that you would have to go through the top-end camera with different settings.
Turn Yield	
(Pa)	C: that sounds very good. S: [silent]
(Pro)	C: uh huh oh yeah S: It's a perfect camera for learning the basics of how photography works…
Ending Interaction	
(Pa)	C: ok I'll have a think about it thank you. S: no worries.
(Pro)	C: okay sounds good, thank you very much. S: no problem see you again. *(go to back service counter)*

Table 5 shows some example interaction patterns that were categorized. Interaction patterns that did not easily fit in with our scenario (e.g. "my brother has a Minolta and he swears by it how does it compare") were not annotated, since we consider them to be unrepeatable and thus will not be well-learned. Fig. 6(a) shows the same projection as Fig.5 (b), but colored according to our annotation. From the projection, we see that "Q&A" is somewhat grouped into smaller clusters, each representing one feature, "Turn yield" is not so well grouped, and "Ending interaction" is well grouped.

When we inspect the projection, it becomes apparent that the neurons capture some general, and in fact quite useful, semantic information about interaction patterns. In our scenario, we have repeated interaction pattern pairs of the customer and shopkeeper doing semantically similar tasks (e.g. Q&A about Sony's sensor size). So, even when there are natural variations in the customer utterances or differing shopkeeper responses due to interaction styles, there may be parts of the interaction history where similar actions are repeated among the customers or shopkeepers. As a result, this allows the neural network to exploit the unknown structure in the input distribution [36] in order to learn features at various levels of abstraction [35].

We first examine the data annotated with Q&A. We observed that interaction patterns are mostly grouped according to the camera feature. Fig. 6(b) shows the Q&A for "sensor size". We observed several ways the customer could ask about "sensor size", as well as several different styles of answers by the shopkeepers, which were mapped to six different robot actions. Despite the differences, the projection was close together, indicating that the neural network was able to find semantic similarities among Q&A about the same feature. This phenomenon of Q&A about the same camera feature projected near each other was also observed for other camera features, such as price and color. On the other hand, for features like exposure, the projection was further apart between the passive and proactive shopkeeper. We believe that this occurs because the shopkeepers tended to respond in similar ways to questions relating to price and color, in contrast with more complex

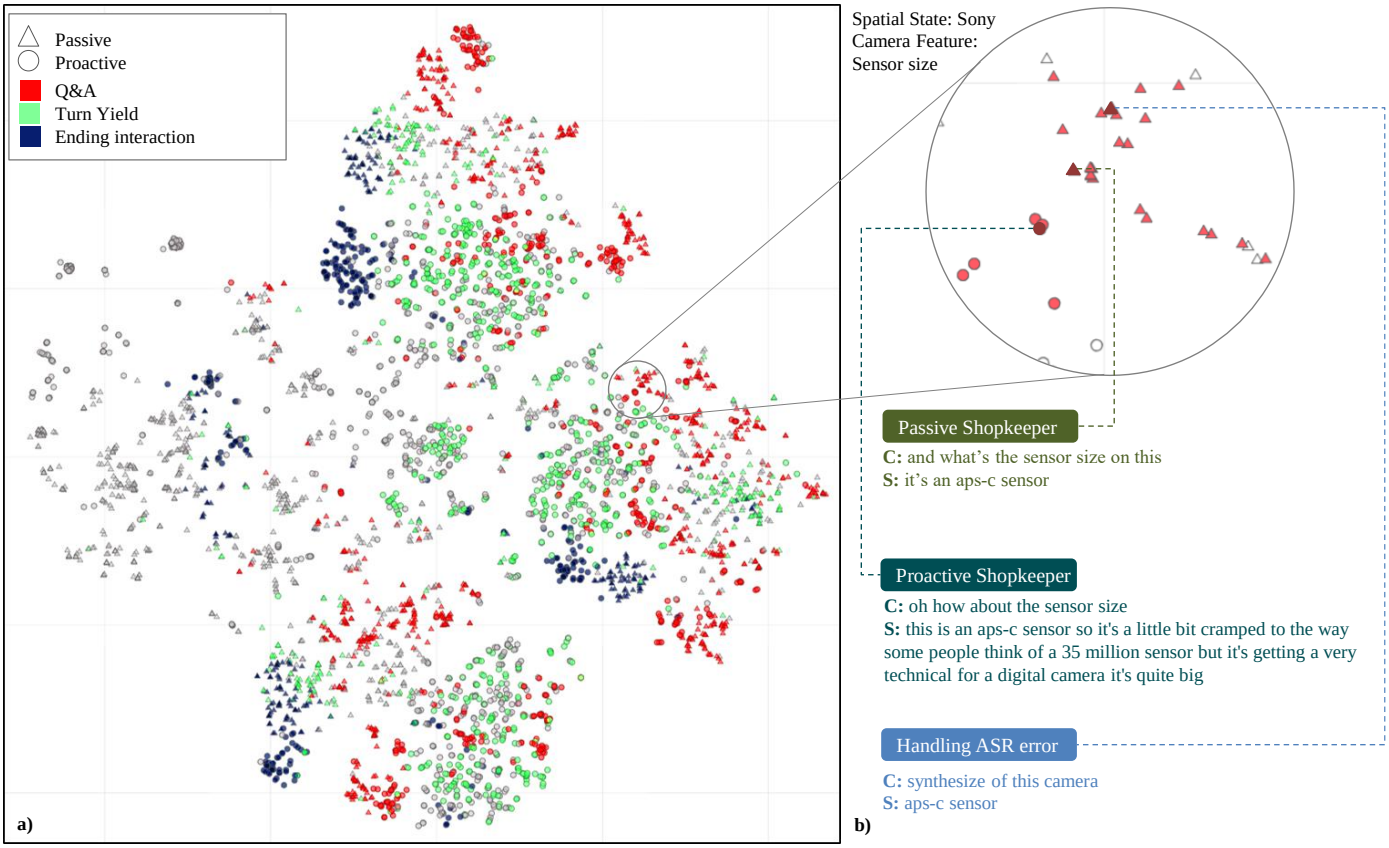


Fig. 6. Projection of the last MLP hidden layer activations for combined data with (a) annotation for Q&A, turn yield, and ending interaction. (b) A zoomed in view of Q&A about "sensor size". Here, MLP learns that questions are semantically similar, even when they are lexically different. Despite the difference in response style for passive and proactive shopkeeper, the neural network also learns that they are similar in terms that that the utterance is about answering a question about sensor size.

questions about long exposure.

In addition, we are also pleasantly surprised that despite some utterances being misrecognized by ASR (e.g. "sensor size" mistaken for "synthesize"), they were often projected correctly together with semantically-related utterances about that camera feature. This is an important observation, as it shows that the learning system has some robustness to ASR errors and is still able to predict socially-appropriate behaviors despite ASR errors, as demonstrated in [7].

For "turn yield", we observed that the projection did not show such clean semantic grouping. One reason for this might be that these interaction patterns are heavily dependent on previous interaction history, and thus few data samples that share similar context were observed, resulting in projection being scattered throughout the sample points.

For "Ending interaction", both the customer and the shopkeeper used a large variation of phrases to signal ending the current interaction. For customers, they sometimes thanked the shopkeeper or indicated they would come back later. The shopkeepers usually acknowledged the customer's response in different ways and/or moved back to the service counter. Despite these variations, the projections were close together, suggesting that the neural network learned the semantic similarity of the "Ending interaction" examples. We believe that enough repeatability of the "Ending interaction" pattern was observed during training for the neural network to uncover similarities beyond surface text.

In short, we consider this projection of the MLP to provide insightful visual feedback for understanding the learning system. Our inspections of the projection indicate that the neural network was able to learn shared neural representations for some semantically similar interaction patterns, despite differences in the observed behaviors from the shopkeepers.

G. The amount of training data with respect to system performance

While we have demonstrated that the *proposed* condition resulted in significantly better performance than the *baseline* condition, we are also curious to what extent the improved performance is due to having more training data in the *proposed* condition than the *baseline* condition. To investigate the effect of the amount of training data on the system performance, we undersampled the training data (i.e. 2183 training examples) in the *proposed* condition such that it was equivalent to the average amount of training data in the *baseline* condition. We trained and evaluated this undersampled dataset as described by the procedure in Sec. V.A. The behavior correctness for the undersampled dataset was 74.2% (i.e. 334 acceptable behaviors out of 450 test examples) with a Kappa value of 0.782, compared with a behavior correctness of 67.6% in the *baseline* condition (i.e. 304 acceptable behaviors out of 450 test examples). We applied a chi-square test, which revealed that the ratio of the correctness in the *proposed* condition with the

undersampled dataset was significantly higher than the baseline condition, (χ^2 (1, N=450) = 4.528, $p < .05$). This suggests that even given the same amount of training data, a system trained with the undersampled data from both shopkeepers performs better than a system trained only on data of either the passive or proactive shopkeeper.

This result, showing that the system performs better when jointly training from both corpuses than from a single corpus, is echoed by similar results that have been found in studies of robot manipulation [37]. Just like they have hypothesized, we also speculate that having training examples from both shopkeepers may expose certain customer behaviors that are not usually elicited through interactions with a single shopkeeper. Both shopkeepers' actions are dependent on the customers, and interaction with one shopkeeper may reveal certain customer behaviors that are unseen when interacting with the other shopkeeper. By sharing the interaction data between the two shopkeepers, this will lead to more diversity in the customer's actions. Our data-driven learning methodology requires perception of the customer's action, abstraction of the social state, and generation of the equivalent robot actions. Therefore, at the very least, training the perception modules from a diverse set of customer actions, instead of customer actions elicited from just one shopkeeper, should be advantageous to the system's performance.

VI. Discussion

A. Analysis of acceptable versus unacceptable behaviors

To better understand the ways the system failed, we further examined the behaviors that were rated as "unacceptable" in the proposed condition (combined data set). Based on Table 3, 366 out of 450 behaviors were rated as "acceptable". We analyzed the remaining 84 behaviors and identified three categories of errors:

- Answer a question about camera feature incorrectly (34 examples, 40.5% of all errors). Example: when a customer asked "It's for my own business, what about this, does it take long exposure pictures?", the predicted behavior was "this is the best camera we have in the store" instead of predicting an answer about long exposure pictures.
- Generate an out-of-context behavior (40 errors, 47.6%). Example: when a customer wanted to takes pictures of family and friend, the system predicted to introduce an advanced camera instead.
- Continue to interact with the customer even after the customer decided to leave (10 errors, 11.9 %). Example: when a customer said "thank you for your help today", the system predicted "would you like to try taking a picture with it?"

We believe many of these errors were due to reasons like ASR failure or insufficient training examples. While we did not thoroughly investigate the specific causes behind these "unacceptable" behaviors, an in-depth analysis of errors like these was reported for a similar system [7].

B. Generalization and Scalability

We believe that this data-driven approach can be generalized to reproduce interaction styles in other scenarios. In our human-human training interactions, interaction data from the passive and proactive shopkeepers was captured and provided to the neural network using an additional *style* feature in the input. We expect that this *style* feature can also be used as a way to capture other types of interaction styles. We can imagine a museum tour guide robot that learns from two different human guides who behave differently, a person who gives a brief overview for each exhibit and a person who expatiates on the details of the exhibit. Likewise, this approach of capturing and reproducing interaction style could be applied to a receptionist robot, where we could control the *style* feature to either reproduce a more business-like interaction or a more friendly and casual receptionist. While this work demonstrates the feasibility of learning two distinct interaction styles, it will be interesting to investigate how the system can learn from multiple styles in the future, for example, by adding a new value in the *style* feature for each new distinct interaction style. Likewise, it would be interesting to combine data from multiple demonstrators to more robustly learn a single *style*. There may be some domains in which we will need new techniques in order to successfully reproduce interaction styles that are socially-appropriate. These domains might require the robot to establish a knowledge base about the user. One example might be an educational robot that teaches language. Even if the robot could successfully reproduce the style of a more authoritative or more supportive human teacher, it would still need to have knowledge about a student's progress or level of comprehension to successfully interact with the student.

Connected with the concept of scalability, we have demonstrated that jointly learning from shopkeepers with differing behavior styles can contribute better than learning from one person, as compared with only using data from that one person. Traditional approaches to learning have often focused on learning task-specific models, sometimes requiring thousands of examples to be collected to learn a unique model for each set of similar tasks. This is based on the implicit assumption that learning across a variety of tasks does not help. This work attempts to break that myth by showing that jointly learning from shopkeepers with different interaction styles is an effective approach. Once passive collection of interaction data becomes practical, we can imagine training from not just two, but many shopkeepers in a real shop scenario and reproducing the interaction style of each individual shopkeeper.

VII. Conclusion

In this work, we have demonstrated a technique for learning multimodal interaction logic by imitation from two separate corpuses of human-human interaction data representing distinct interaction styles. Through the addition of a "style" feature in the input to the predictor, we were able to adjust the robot's output behavior at runtime to resemble either of the two interaction styles, or some combination of the two. Examination of hidden layer activations indicated that the system appears to

have learned to associate semantically similar behaviors across the two interaction styles.

Through offline evaluation based on human-human interaction situations held out from the training data, we showed that training the system on both corpuses together improved the system's overall performance in predicting behaviors for both of the two interaction styles. Interestingly, we also found that combining the training data from both corpuses and downsampling it to the size of the original training data set still resulted in an improvement in behavior prediction, a finding which is similar to results which have been reported in other fields of robotics.

The task of programming social robots to be both convincingly humanlike and functionally useful presents many challenges, and learning-by-imitation techniques offer a number of benefits over other approaches, particularly in terms of scalability and robustness. However, one drawback of purely learning-based methods is that they tend to be "black box" systems that are difficult to control. The technique we have proposed suggests the possibility of providing a robot's operator with some degree of high-level control over the robot's personality or interaction style, which could constitute one step towards finding the right balance between manual controllability and learning-based robustness for interactive social robots.

APPENDIX

Here we describe the data abstraction techniques we used that enable the learning of high-level interaction logic in human-robot interaction to be achieved in an entirely data-driven way, that is, without any kind of manual annotation or cleanup of the sensor data. This follows the work presented in [7].

Abstraction of input features

Here, we describe the features used in the *joint state vector*, including the abstraction of motion (consisting of *current location*, *motion origin*, and *motion target* of both participants, and a *spatial formation*), and an *utterance vector* of the current spoken utterance.

Motion Abstraction: The purpose of the motion abstraction step is to characterize a set of stopping locations, motion trajectories, and spatial formations which can be used to describe the motion of the customer or shopkeeper as a combination of discrete state variables rather than raw position or velocity data.

To begin the analysis, we segmented all trajectories in the training data into moving and stopped trajectories, based on a velocity thresholding technique presented in [38]. We spatially clustered these trajectory segments to identify a discrete set of typical **stopping locations** and **motion trajectories** for each role (customer and shopkeeper).

For stopping locations, we used k-means clustering, identifying five stopping locations for the customer (i.e. the locations of the 3 cameras, the middle, and the door) and five for the shopkeeper (i.e. the locations of the 3 cameras, the middle, and the service counter).

For moving trajectories we used k-medoid clustering based on spatiotemporal matching using dynamic time warping.

We created rules for identifying a predetermined set of common **spatial formations** based on the distance between the interactants and their locations. The rules for spatial formations are similar to three existing HRI proxemics models: (1) *present object* [39]: both interactants were at stopping locations corresponding to the same camera, (2) *face-to-face* [40]: both interactants are within 1.5m of each other but not at a camera, and (3) *waiting* [41]: if the shopkeeper was at the service counter while the customer was not.

In addition, we also identified the current spatial target for a particular spatial formation. The *formation target* for "present object" can be either Sony, Nikon, or Canon, whereas the *formation target* for the spatial formation "face-to-face" and "waiting" is 'none'.

Utterance Vectorization: We performed utterance vectorization of the customer and shopkeeper using common text-processing techniques. Specifically, we removed stop words, applied a Porter stemmer, enumerated n-grams up to 3, and performed Latent Semantic Analysis [42] to reduce the dimensionality to 1000. To emphasize important keywords, we also used the AlchemyAPI cloud-based service [3] to automatically extract keywords from each utterance and represented the keywords separately in the vector (200 dimensions). By using this procedure, we were able to take any input utterance and represent it using a 1200-dimensional vector. Vectorization of customer and shopkeeper utterances were performed independently.

Defining Robot Actions

In our system, each observed shopkeeper action must correspond to a discrete robot action. A robot action consists of an utterance (represented by an ID number) with a corresponding target formation.

Shopkeeper Utterance: In order to reproduce shopkeeper speech with a robot, it is necessary to define a set of discrete utterance actions. Common utterances are frequently repeated in the training data (for example, variants of answering about the color of Sony occur 61 times), but these instances often include slight differences due to speech recognition errors or individual variation. Thus, we used bottom-up hierarchical clustering based on lexical cosine similarity to group these repeated and similar utterances into clusters corresponding to discrete robot speech actions.

From each shopkeeper utterance cluster, one utterance was selected for use in behavior generation. For each utterance, we compute the cosine similarity of its term frequency vector with every other utterance in the same cluster, and we sum these similarity values. The utterance with the highest similarity sum is chosen as the typical utterance. A total of c typical utterances was extracted from the shopkeeper utterance clusters, which

[3] http://www.alchemyapi.com

can be used to generate robot speech. Notice the typical utterance can also be “none”, which means that the robot does not output an utterance.

Target Formation: We use the same abstraction rule described earlier to represent a target spatial formation for the robot (i.e. *present product*, *face-to*-face, *waiting*, or *none*). This allows the robot to precisely calculate its target position and facing direction defined by the specefic HRI model, in accordance with its estimation of the customer’s destination.

For example, if the predicted target formation is different from the robot’s current formation, the robot moves to attain the new target formation. Specifically, if the predicted formation is *face-to-face*, the robot approaches the customer; if the predicted formation is *waiting*, it returns to the service counter; if the predicted formation is *present-object*, the robot approaches the target object; and if the predicted formation is *none*, the robot stays where it is.

ACKNOWLEDGMENT

This work was supported in part by the JST ERATO Ishiguro Symbiotic Human-Robot Interaction Project, Grant Number JPMJER1401 and in part by JSPS KAKENHI Grant Number 25240042.

ETHICAL APPROVAL

This research was conducted in compliance with the standards and regulations of our company’s ethical review board, which requires every experiment we conduct to be subject to a review and approval procedure according to strict ethical guidelines.

REFERENCES

[1] M. P. Michalowski, S. Sabanovic, and H. Kozima, "A dancing robot for rhythmic social interaction," in *Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on*, 2007, pp. 89-96.

[2] H. Kozima, M. P. Michalowski, and C. Nakagawa, "Keepon," *International Journal of Social Robotics*, vol. 1, pp. 3-18, 2009.

[3] T. Kanda, T. Hirano, D. Eaton, and H. Ishiguro, "Interactive robots as social partners and peer tutors for children: A field trial," *Human-Computer Interaction*, vol. 19, pp. 61-84, 2004.

[4] C.-W. Chang, J.-H. Lee, P.-Y. Chao, C.-Y. Wang, and G.-D. Chen, "Exploring the possibility of using humanoid robots as instructional tools for teaching a second language in primary school," *Educational Technology & Society*, vol. 13, pp. 13-24, 2010.

[5] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, and M. Fiore, "Spencer: A socially aware service robot for passenger guidance and help in busy airports," in *Field and Service Robotics*, 2016, pp. 607-622.

[6] K. Severinson-Eklundh, A. Green, and H. Hüttenrauch, "Social and collaborative aspects of interaction with a service robot," *Robotics and Autonomous systems*, vol. 42, pp. 223-234, 2003.

[7] P. Liu, D. F. Glas, T. Kanda, and H. Ishiguro, "Data-Driven HRI: Learning Social Behaviors by Example From Human-Human Interaction," *IEEE Transactions on Robotics*, vol. 32, pp. 988-1008, 2016.

[8] C. Breazeal, N. DePalma, J. Orkin, S. Chernova, and M. Jung, "Crowdsourcing Human-Robot Interaction: New Methods and System Evaluation in a Public Environment," *Journal of Human-Robot Interaction*, vol. 2, pp. 82-111, 2013.

[9] W. B. Knox, S. Spaulding, and C. Breazeal, "Learning from the Wizard: Programming Social Interaction through Teleoperated Demonstrations," in *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 2016, pp. 1309-1310.

[10] J. E. Young, E. Sharlin, and T. Igarashi, "Teaching robots style: designing and evaluating style-by-demonstration for interactive robotic locomotion," *Human–Computer Interaction*, vol. 28, pp. 379-416, 2013.

[11] P. Liu, D. F. Glas, T. Kanda, and H. Ishiguro, "Learning proactive behavior for interactive social robots," *Autonomous Robots*, pp. 1-19, 2017.

[12] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, 2016, pp. 3406-3413.

[13] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning Hand-Eye Coordination for Robotic Grasping with Large-Scale Data Collection," in *International Symposium on Experimental Robotics*, 2016, pp. 173-184.

[14] C. G. Jung, *The archetypes and the collective unconscious*: Routledge, 2014.

[15] R. Toris, D. Kent, and S. Chernova, "The robot management system: A framework for conducting human-robot interaction studies through crowdsourcing," *Journal of Human-Robot Interaction*, vol. 3, pp. 25-49, 2014.

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

[17] A. L. Thomaz and C. Breazeal, "Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance," in *AAAI*, 2006, pp. 1000-1005.

[18] A. L. Thomaz and C. Breazeal, "Teachable robots: Understanding human teaching behavior to build more effective robot learners," *Artificial Intelligence*, vol. 172, pp. 716-737, 2008.

[19] Y. Nagai, C. Muhl, and K. J. Rohlfing, "Toward designing a robot that learns actions from parental demonstrations," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 3545-3550.

[20] M. E. Foster, S. Keizer, Z. Wang, and O. Lemon, "Machine learning of social states and skills for multi-party human-robot interaction," in *Proceedings of the workshop on Machine Learning for Interactive Systems (MLIS 2012)*, 2012, p. 9.

[21] H. Admoni and B. Scassellati, "Data-driven model of nonverbal behavior for socially assistive human-robot interactions," in *Proceedings of the 16th International Conference on Multimodal Interaction*, 2014, pp. 196-199.

[22] K. Harada, H. Hirukawa, F. Kanehiro, K. Fujiwara, K. Kaneko, S. Kajita, and M. Nakamura, "Dynamical balance of a humanoid robot grasping an environment," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 2004, pp. 1167-1173.

[23] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 4263-4270.

[24] L. Pinto and A. Gupta, "Learning to push by grasping: Using multiple tasks for effective learning," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, 2017, pp. 2161-2168.

[25] I. Leite, A. Pereira, A. Funkhouser, B. Li, and J. F. Lehman, "Semi-situated learning of verbal and nonverbal content for repeated human-robot interaction," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, 2016, pp. 13-20.

[26] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, and G. Corrado, "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation," *arXiv preprint arXiv:1611.04558*, 2016.

[27] R. Sennrich, B. Haddow, and A. Birch, "Controlling politeness in neural machine translation via side constraints," in *Proceedings of NAACL-HLT*, 2016, pp. 35-40.

[28] S. Banerjee, P. Biyani, and K. Tsioutsiouliklis, "Transforming Chatbot Responses to Mimic Domain-specific Linguistic Styles," *Second Workshop on Chatbots and Conversational Agent Technologies*, Sep. 2016.

[29] J. Baraglia, M. Cakmak, Y. Nagai, R. Rao, and M. Asada, "Initiative in robot assistance during collaborative task execution," in *Human-*

Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on, 2016, pp. 67-74.

[30] D. Brscic, T. Kanda, T. Ikeda, and T. Miyashita, "Person Tracking in Large Public Spaces Using 3-D Range Sensors," *Human-Machine Systems, IEEE Transactions on*, vol. 43, pp. 522-534, 2013.

[31] C. Raffel and D. P. Ellis, "Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems," *arXiv preprint arXiv:1512.08756*, 2015.

[32] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model," *SLT*, vol. 12, pp. 234-239, 2012.

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448-456.

[34] L. Van Der Maaten, "Accelerating t-SNE using tree-based algorithms," *Journal of machine learning research*, vol. 15, pp. 3221-3245, 2014.

[35] Y. Bengio, "Learning deep architectures for AI," *Foundations and trends® in Machine Learning*, vol. 2, pp. 1-127, 2009.

[36] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," *ICML Unsupervised and Transfer Learning*, vol. 27, pp. 17-36, 2012.

[37] L. Pinto and A. Gupta, "Learning to Push by Grasping: Using multiple tasks for effective learning," *arXiv preprint arXiv:1609.09025*, 2016.

[38] L. Guéguen, "Segmentation by Maximal Predictive Partitioning According to Composition Biases," in *Computational Biology*. vol. 2066, O. Gascuel and M.-F. Sagot, Eds., ed: Springer Berlin Heidelberg, 2001, pp. 32-44.

[39] F. Yamaoka, T. Kanda, H. Ishiguro, and N. Hagita, "How close? A model of proximity control for information-presenting robots," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, Amsterdam, The Netherlands, 2008, pp. 137-144.

[40] E. T. Hall, *The Hidden Dimension*. London, UK: The Bodley Head Ltd, 1966.

[41] T. Kitade, S. Satake, T. Kanda, and M. Imai, "Understanding suitable locations for waiting," in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, 2013, pp. 57-64.

[42] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, pp. 259-284, 1998.