2021 30th IEEE International Conference on Robot and
Human Interactive Communication (RO-MAN)
August 8-12, 2021. Vancouver, Canada (Virtual Conference)

2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN) | 978-1-6654-0492-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/RO-MAN50785.2021.9515525

Shaping Affective Robot Haru's Reactive Response

Yurii Vasylkiv[1], Zhen Ma[2], Guangliang Li[2*], Heike Brock[3], Keisuke Nakamura[3],
Irani Pourang[1] and Randy Gomez[3]

***Abstract*— We describe a method of teaching a robot its empathic behavioural response from its interaction with people. We used the input modalities such as relative spatial information, facial expressions, body gestures and speech information as perception input that triggers the robot's empathic response. First, we bootstrap the training through a pre-learning mechanism in which training is conducted by users who know the robotic system. This phase provides simulation-based training using a simple graphical user interface to simulate the input, rewards and correction feedback. In the second phase, we developed an online learning scheme for naive users to personalize their robot further, building on top of the bootstrapped model. Here, we developed a natural user interface that enables natural human-robot interaction via the suite of sensors that allows the users to provide evaluative feedback during the interaction with the robot. We evaluated the system and our results show that bootstrapping is an efficient tool to hasten the robot's learning while online learning provided some form of personalization in the real environment with naive users.**

I. Introduction

The tabletop robot Haru is a research platform to study affective engagement with a robot as a means of long-term human-robot interaction [1]. In this research, we aim to develop a new kind of robotic companion — a new form of companion species [2] that one day enables humans to build bond through empathetic interaction. In the first phase of the study, we focus on teaching the robot simplistic interactions similar to those shared with pets. We would like to achieve this by tapping the robot's empathetic character. Haru's design evolves primarily on maximizing empathy with rich multimodal channels for utmost expressiveness [3]. Since the communication of affects is integral to Haru, we provided a tool for animators to easily design animation routines in their native software platform. This expression composer studio allows designers to compose context-rich multimodal expressions of the robot and subsequently transform them into hardware-ready robotic routines [4]. The idea is for animators to curate Haru's expressive routines resulting into a well-designed repertoire of robotic routines, while robot application designers focus on developing interaction methods and make use of the pool of these empathetic routines as robot's actions.

Currently, the programming of Haru's actions is based on a tightly controlled environment relying on a rule-based approach in selecting the appropriate actions [5], [6], [7]. However, this approach is not sustainable when developing a new kind of companion species such as Haru. If Haru has to become a social creature like pets, humans should be able to train Haru's responses based on what it perceives. Hence, in this paper we propose to develop a mechanism to enable Haru to learn from the interaction of humans and automatically select the appropriate action from the pool of expressive routines in its repertoire. Machine learning approaches offer a convenient way to train robots. In particular, Reinforcement Learning (RL) has been a popular tool for agents to learn through interaction (i.e., taking actions) with the physical world [8]

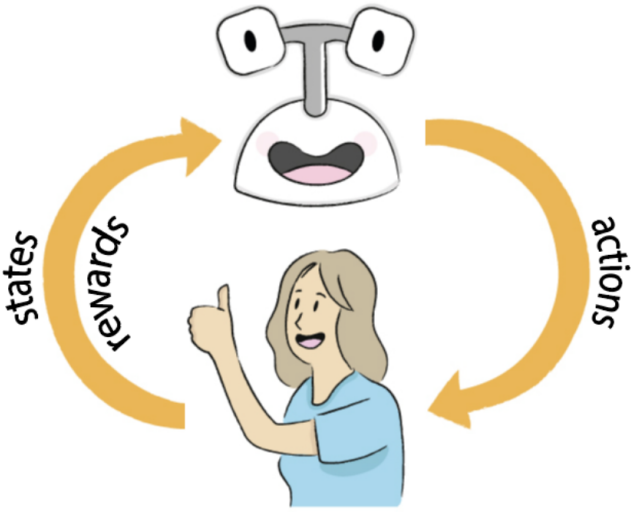

Fig. 1: Haru's learning framework

In this paper, we employ a human-in-the-loop RL approach [9], [10], [11], [12] to enable the agent's real-time learning. Using Markov Decision Process (MDPs) [13] to describe the interaction between an agent and its environment (see Figure 1), the goal of our proposed system is to learn an action-selection strategy in order to optimize some performance metrics such as user reward (feedback). For that, in our experiment, we engage human participants as trainers to shape the robot's actions. The first set of trainers are knowledgeable of the robotic system (i.e. perception and pool of actions). They are used to bootstrap learning through simulated training using an interface tool with wide-range of options to train the robot. The second set of shapers are the end users (naive users), which are the actual people who will be interacting with the robot in their day-to-day lives. Unlike the boostrapping shapers, the naive users are provided with a simple and natural interface to give positive/negative feedback to facilitate an online learning mechanism. The idea is to construct the base model through bootstrapping to hasten learning, while further personalization is provided

[1]University of Manitoba, [2]Ocean University of China, [3]Honda Research Institute Japan Co., Ltd.
*Corresponding author

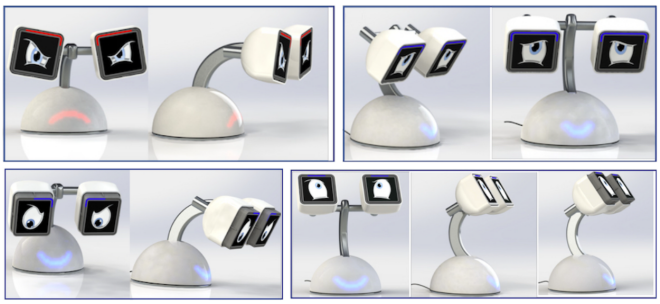

Fig. 2: Examples of a few Haru's actions (emotions/expressions) expressed through the animated emotive routines: Anger (top left), Shyness (top right), Curiosity (bottom left), Adoration (bottom right)

through the online learning as the end user continuously interacts with the robot. As a result, the robot learns to properly understand social cues and reacts to them with corresponding actions that are deemed appropriate, relevant and correct by the human shaper. Inspired by real-world methods of training social creatures such as pets, our participants create an arbitrary situation and expect to elicit a certain action (e.g. emotional reactions such as happiness, anger or sadness) from the robot. Then, they provide evaluative feedback, respectively reward, and a correction (optional, bootstrapping phase) on the robot's performance for the robot to optimize its response.

This paper is organized as follows. We present the background of the robotic platform in Section II. We discuss the bootstrapping and online learning frameworks in Section III and Section IV, respectively. The experimental setup is presented in Section V, followed by results and discussion in Section VI. Finally, we conclude the paper in Section VII.

II. Robotic Platform

A. Perception Mechanism (Human Input States)

To understand its environment, the robot needs to make sense of the input modalities through the use of perception sensors (e.g. RGB camera, depth-sensing camera and microphone array) and recognition modules. To reduce the dimensionality of the state space, we limit the input to the following perception and recognition modules.

1) Face Direction: This module uses depth information collected from the depth sensor and provides Haru with spatial awareness about the person interacting with it. In addition, the face direction modality represents a unit vector describing the orientation of the human face derived from the skeleton joints of the person. This modality is set to "AtRobot" and "Away" when the person is facing towards and away from the robot, respectively.

2) Facial Expression: The facial expression recognition module fetches single images of the internal camera data for frame-wise inference of human emotional states with a pre-trained Convolutional Neural Network (CNN). The model is derived from the MobileNets architecture [14] and is made available for public use [1]. We utilize the network as is and do not perform any further retraining or model adaptation. However, to add robustness to our prediction process, we smooth the confidence outputs with an additional moving average filter of window size $\omega_e = 5$ along the temporal domain. This modality was limited to classify five classes (labels): "Neutral", "Smile", "Surprise", "Sadness", and "Anger" (see Table I), where each represents an actual facial expression on the human's face in discrete point of time.

3) Gestural Expression: The gestural expression recognition module classifies segments of joint movement features with a CNN architecture specifically designed for the given interaction scenario. We continuously fetch the user's joint positions tracked with a depth sensor and split the data stream into segments based on the movement properties of designated landmark joints. Once a new series of joint position trajectories is created, we transform it into a set of angular and distal features [15] and re-scale the resulting feature segment to a standardized length using cubic interpolation. Next, we evaluate the resulting 'movement image' with the CNN to obtain the desired class label and confidence prediction information. The CNN constitutes of a very basic architecture that was similarly used in related application scenarios [16], [17]. Similarly to the facial expression recognition module, this modality was also limited to a small number of classes (labels): "Neutral", "Waving"(hand), "Clapping"(hands), "FaceCover"(hand(s) covering face), "Shrugging"(shoulders), "Thumb Up", and "Thumb Down". The first five are the part of the human input states (see Table I), while the last two are reserved control gestures (see Section IV-A).

4) Speech Expression (Wake Up Phrase and Intent): The hands-free acoustic speech is processed via a microphone array processing module resulting in a separated speech signal, which is then used as input through a speech recognition module [18]. Its output is further analyzed for the presence of the words or semantic information related to directly addressing the robot's name (e.g. "Hey, Haru", "Haru", "Ok, Haru", etc.) which is referred to as the wake phrase [19], [20].This explicitly informs the robot of the human interaction with it (the robot). In the event that the text is not classified as a wake up phrase, it is processed for control speech phrases (i.e. "Yes, Haru", see Section IV-A) and for check of speech sentiment (e.g. greeting, goodbye or endearment, see Table I) through a language sentiment analyzer module [21].

B. Robot Actions

The robot's action refers to the repertoire of expressive routines that are designed and curated by the animators.

[1] https://github.com/EliotAndres/tensorflow-2-run-on-mobile-devices-ios-android-browser

Modalities	Classes
Face Direction	“AtRobot”, “AwayFromRobot”
Face Expression	“Neutral”, “Smile”, “Surprise”, “Sadness”, “Anger”
Gestural Expression	“Neutral”, “Waving”, “Clapping”, “FaceCover”, “Shrugging”
Speech Expression: Wake-Up-Phrase	“Present”, “NotPresent”
Speech Expression: Intent	“None”, “Greeting”, “Goodbye”, “Endearment”, “Embitterment”, “Directive”

TABLE I: Summary of the human input states.

Action Class	**Intensity Level**	
Sadness	1	2
Happiness	1	2
Shyness	1	2
Anger	1	2
Adoration	1	2
Thinking	1	2
Listening	1	2
Curiosity	1	2

TABLE II: Action Space. The list of the robot’s actions (emotive routine behaviors).

These are the expressions that are triggered by the perceived human states. In practice, we aim to have a large pool of routines that represents the various expressiveness of the robot. In this paper, we limit it to using eight routines as exemplary depicted in Figure 2: “Sadness”, “Happiness”, “Shyness”, “Anger”, “Adoration”, “Thinking”, “Listening”, “Curiosity”. The idea is that, as the person interacts with Haru, the manifestation of their states would elicit an emotional response (action) from the robot. Moreover, to slightly increase the action space (from 8 to 16 possible combinations) we also introduced two intensity levels (i.e. 1 and 2). Level 1 implies a low or normal level of expressivity, level 2 signifies a high or extreme expressivity and used to exaggerate the degree of a particular affect. The robot’s possible actions (actions space) are summarized in Table II.

III. Shaping by Bootstrapping

A. Simulation Interface: Interaction and Feedback

In order to hasten the learning process, we first bootstrapped the model using a simulation interface (see Figure 3) that simulates all the supported human input states and robot actions. The figure also highlights the five important steps the learning procedure is composed of:

1 **Select State:** to create/compose/design/select the human input state that Haru will react to. The five drop-down menus (Facial Expression, Gestural Expression, Speech Expression: Intent, Face Direction, Speech Expression: Wake-Up-Phrase) contain the values listed in Table I respectively.

2 **Apply State:** to apply the selected human input state

3 **Evaluate Haru’s Action:** to evaluate Haru’s action generated by the RL algorithm, that is to make a decision on whether it matches the user’s expectation or not.

4 **Provide Feedback:** to provide a reward and an optional correction. The user gives a positive reward (“Accept” checkbox selection) if Haru’s reaction is acceptable, or a negative reward (“Reject” checkbox selection) otherwise. In case of the negative reward, the user has an option to provide a correct answer (robot’s action).

5 **Track Progress:** to track the information about the human input state and robot’s action being currently learned by Haru, as well as the number of ones already learned.

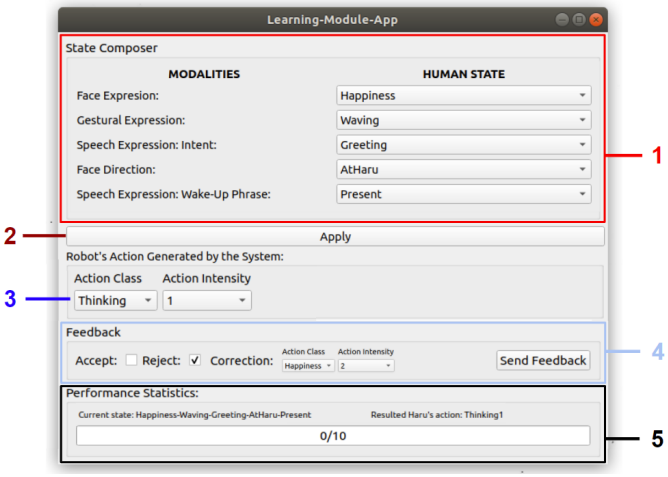

Fig. 3: GUI for bootstrapping the model.

B. Bootstrapping Learning Algorithm

We used Q-learning [22] — the most commonly used method in RL. However, different from the traditional Q-learning in RL, the rewards for learning are not provided by a pre-defined reward function, but delivered by a human-in-the-loop [9], [10], [11], [23], [24], [25]. In our setting, this means that the robot receives labels (classes) for each modality from the user (human trainer) selection in the GUI (see Figure 3), which simulates the human input state. Then, based on the perception of the human input state, from the set of actions (see Section II-B), the algorithm greedily picks a single action with the largest Q-value as it is done in

the traditional Q-learning. That action is shown to the user through the GUI. The user can then choose to accept or reject this action based on their subjective opinion. This user's response is sent back to the robot as an evaluative reward to update the learning model.

In addition, besides the evaluative feedback, we also provided a channel for a user to provide a correction (a correct action). The correction is used when the user disagrees with the selected action. In that case, they may choose to correct it, which is directly used to update the optimal action for that particular human state, as shown in Figure 4.

To be specific, at time t, the agent first detects the current human input state s_t that a user (human trainer) creates using the GUI. The algorithm then picks the action that currently has the largest Q-value, or randomly for the actions with equal Q-values (e.g. all Q-values are set as zeros at the beginning, etc.) for the current human input state s_t. The selected action is sent to the user via GUI who will evaluate whether the selection is correct (most optimal) for the current human input state s_t. If they accept it, a positive value of "+2" is sent back to the algorithm; if they reject and do not provide any correction, the algorithm receives a negative value of "-2". This user's response is received by the algorithm as a human reward R_h that is used to update the corresponding Q-values.

$$R_h = \begin{cases} +2 & \text{accept suggested action} \\ -2 & \text{reject suggested action (no correction)} \end{cases} \quad (1)$$

The value of "2" in Equation 1 was experimentally found to be sufficient in order to change particular Q-values in a single iteration, significant enough, to affect the algorithm's future action selections.

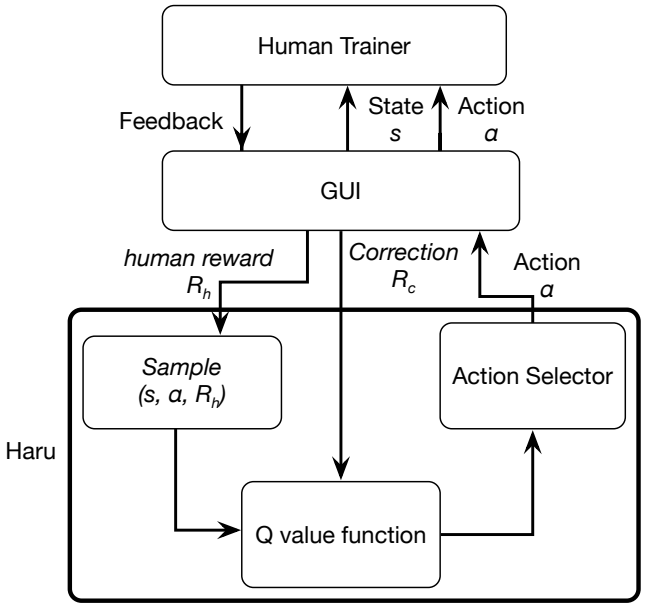

Fig. 4: The scheme of the learning algorithm for bootstrapping.

The received human reward R_h together with the human input state s and selected action a will be used as a sample to update the Q-value function as:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(R_h + \gamma \max_a Q(s', a) - Q(s_t, a_t)) \quad (2)$$

where α is the learning rate [26], γ is the discount factor [22]. If the human trainer rejects the selected action but chooses to correct it, the algorithm will receive a negative value of "-1" for that selected action and a positive value of "+1" for the action of correction. Both are directly used to update the Q-values for the two actions in the current human input state s_t:

$$R_c = \begin{cases} +1 & \text{action of correction} \\ -1 & \text{selected action} \end{cases} \quad (3)$$

The value of "1" in Equation 3 was experimentally found to be sufficient in order to change particular Q-values in a single iteration, significant enough, to affect the algorithm's future action selections. Q-value function calculated with regard to the given correction is denoted as:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha R_c \quad (4)$$

where R_c is the reward for the action of correction. Next, at time $t+1$, the perception system will detect a new human input state s_{t+1} and another action with the largest Q-value will be selected for execution:

$$a \leftarrow \arg\max_a Q(s_{t+1}, a_i), \quad (5)$$

New iterations of action selection, feedback propagation and model update repeat selected by the algorithm action is accepted. The same procedure is applied for all created user's human input states.

IV. Online Shaping Mechanism

A. Natural User Interface: Interaction and Feedback

In order to deploy our learning system in a real environment with Haru and naive users, it is important to provide a natural user interface [27]. Naive users are generally not technology savvy and are not familiar with the specifics of the system. To allow the user to provide feedback for the robot in an intuitive way, we engaged our perception system (see Section II-A) to establish a natural human-robot interaction. Here, the idea is that the system recognizes and interprets the human physical interaction (facial expression, body gesture, speech, etc.) into understandable human input states in real-time. On top of the perception system, we developed a natural interface for the user feedback (see Figure 5). In particular, we built a control gesture (i.e. "Thumbs Up", "Thumbs Down") and control speech phrases (e.g. "no Haru", "yes Haru"). These control gestures and control speech phrases are defined as reserved control commands. They are recognized by the perception system and when encountered are automatically translated only exclusively into

corresponding positive or negative rewards (see Section IV-B). Correction in this setting was not provided as it would complicate the interaction.

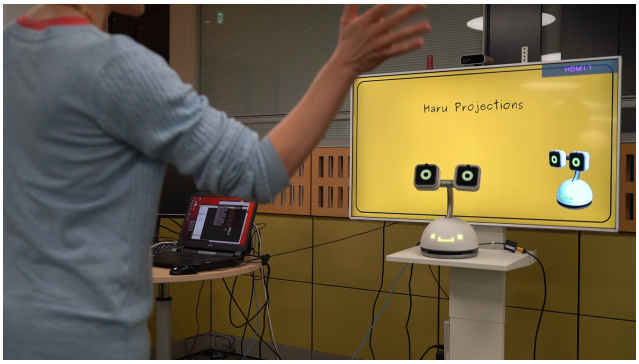

Fig. 5: Online Learning: Natural user interface

B. Online Learning

The algorithm in our online learning is similar to the one used for the bootstrapping, except for the correction that was deliberately left out. In this setting, the GUI is replaced with the natural user interface as it is shown in Figure 6. In addition, the Q-value function is initialized with the bootstrapped model from Section III.

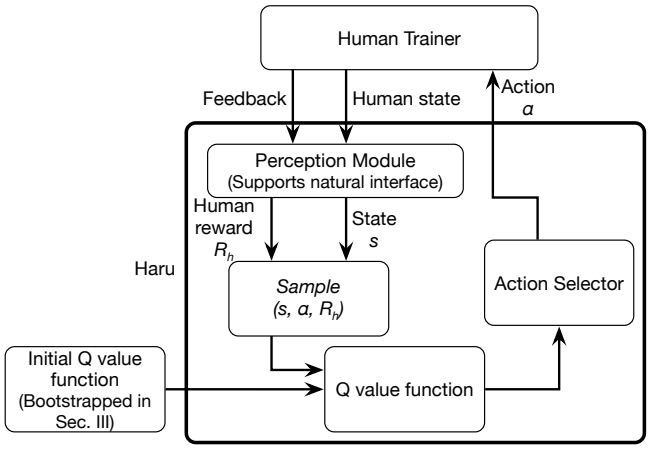

Fig. 6: Online learning algorithm from human evaluative feedback provided by real users (human-trainers).

V. Experiments

In order to test the practicality and effectiveness of our proposed system, we designed two experiments.

In the first experiment we focused on hastening the learning process by bootstrapping of the initial learning model. For this part we selected experienced users knowledgeable of Haru who we denoted as experts. The purpose of this experiment was to quickly obtain an base model which contain preferences of generic population. Normally, such a procedure requires a high volume of iterations and data. Therefore, to enable that, this stage was done in a simulated environment with the provided GUI and keyboard-mouse control (see Figure 3). The participants here were asked to imagine and design 10 unique human input states and teach Haru to react to those with particular actions individually preferred by them. The selection of the desired human input states and delivery of the feedback (reward and correction) were all explained in Section III-A.

The second experiment focused on studying the performance of our learning system in a real environment with Haru. Participants in this stage were naive users. This means the users had no concept of the human states or robot actions, and were not familiar with the workings of the system. The task for participants was mainly identical to the first experiment but was framed differently — create 10 unique scenarios (interactions) and shape Haru's behaviour according to individual preferences by giving a positive reward for those behaviors they approve of, and a negative reward for those behaviors they do not. More specifically, for each scenario participants were asked to interact with Haru using body gestures, facial expressions, face direction and speech. They were further instructed to observe Haru's behaviour as a response to that interaction, and subjectively decide whether it was appropriate or correct to the current situation. The feedback in this setting was given via the natural user interface, namely using one of the two control gestures or control speech phrases from Section IV-A.

The second experiment was conducted in in-lab settings as depicted in Figure 5. Since the number of the robot's actions and recognized human input states were very limited, before the experiment, participants were shortly introduced to the robot's actions. This briefing was made so that they would know what to expect from Haru, and to familiarize them with the range of available human input states to give an idea of what kind of physical interactions would actually be registered by the robot.

One of the main goals of our research was to see how the bootstrapping can improve online learning for social interactions. In order to measure the effect of the bootstrapping on the learning, two conditions of a between-subject study design for our second experiment were composed. The control group was decided to teach Haru "from scratch" with no initial model loaded by the algorithm. For the experimental group, on the other hand, the algorithm loaded the bootstrapped model from the previous experiment as a starting point.

A total of 38 participants were recruited for the two experiments (N=38). To bootstrap the learning model we recruited twelve participants (n=12). To test the effectiveness of the bootstrapping for the second experiment we recruited thirteen participants per condition (n=26).

VI. Results & Discussion

To measure the effect of bootstrapping on online learning for social interactions, we compared the average learning time between the control and experimental groups expressed through the number of times participants rejected the robot's actions selected by the algorithm, as well as the algorithm's

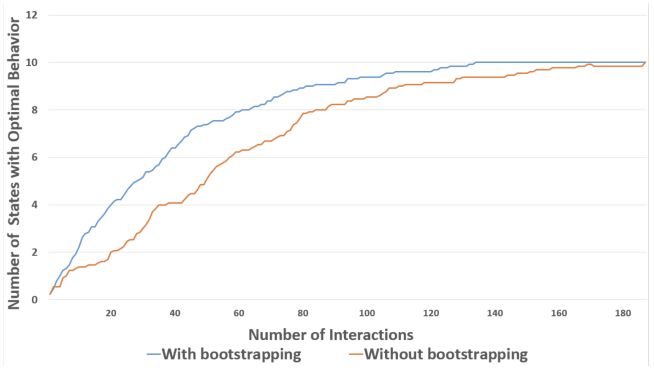

Fig. 7: Learning curves in the online learning in terms of average number of states with learned optimal behavior over all participants in the two conditions.

learning curve. In addition, we also describe the process of personalization that inevitable occurs as different actors want to choose different actions for the same human input states.

A. Number of Rejections

Table III and Table IV show the average total number of rejections for each naive user participant for both the control and experimental groups respectively. According to the conducted comparison (Mann-Whitney U test), the experimental group rejected Haru's actions significantly fewer times. As a consequence, they required significantly fewer iterations to finish the training for each of the ten human input states, and complete the total experiment (U=27.5, p¡.05).

B. Learning Curve

We measured the average algorithm's learning time for all the participants from both the control and experimental groups. The learning time is expressed through the number of learned human input states with the accepted actions over the number of iterations that were required to achieve that (see Figure 7). We can observe that for the experimental condition (i.e. with bootstrapping), even for training only 10 human input states the algorithm learned faster compared to the control group (i.e. without bootstrapping).

The learning pace in the experimental group was not consistent across participants and human input states as it was in the control group, and this can be explained. The expert participants from the first experiment created 120 human input states in total. Since we only forbade reusing the human input states within the single experimental session, there were 84 unique human input states across all 12 participants, while the rest were repeated multiple times overlapping each other with the same or different actions. A similar situation occurred with the naive users from both control and experimental groups, where each group out of 130 created 89 and 84 unique human input states. This suggests that the majority of the human input states were different. However, because the bootstrapped model created by the expert users (see Section III) was loaded as initial model for the experimental group, the human input states created by participants of the experimental group did not only overlap with those of the other participants from the experimental group, but also with those in the bootstrapped model. For that reason, the general trend was that some human input states resulted in faster learning of the correct action if the action matched with that accepted by the general population (expert users); some resulted in longer learning if the correct actions differed as it would take more iterations to inflate the Q-value of the desired action so that it exceeded the rest. According to our results, the participants of the experimental group created only 29 human input states that overlapped with those in the bootstrapped model (24%), for which 19 actions differed from those accepted as correct by the expert users while 36 accepted actions matched. Therefore, again, there is enough evidence to suggest that even a minimum bootstrapping greatly improves the algorithm efficiency, where only 25% overlap was already sufficient to significantly reduce the number of rejections by the participants of the experimental group.

C. Personalization

Throughout the experiment all the participants (regardless of the group) were essentially personalizing the Q-table, that is personalizing the algorithm's action selection for the human input states of their choice. However, for the experimental group this personalization can be observed the most vividly. Figure 8 shows a visualized heat map of the Q-values for all of Haru's actions in fragments of human input states. Each of the fragments here denotes a Q-table made of block squares of different shades indicating Q-values (i.e. the darker the shade is, the greater the Q-value is). The first (top left) fragment illustrates the Q-table for the bootstrapped model generated by the expert users in the first experiment (Section V). The rest of the fragments demonstrate the Q-tables of participants #1, #3, #4, #6 and #13 from the experimental group. The last rows of each of the fragments illustrate specific human input states that had been already learned before (during the bootstrapping) and now is being re-trained (personalized further) by different participants (naive users).

All participants trained Haru to learn an action that was different from the initial model according to their own preferences, where participants #1, #4 and #6 shared their preferences and participants #3 and #13 had a different action in mind. In concrete, the preferred action from the initial model was mostly "Sadness" and "Thinking" (marked through dark coloring). Participants #1, #4 and #6 kept a similar pattern, but assigned more weight to "Thinking". Conversely, participants #3 and #13 shifted their preference towards "Listening" and "Curiosity". Both "Sadness and "Thinking" in their case were deemed irrelevant (as marked by light coloring).

In addition, all participants from the experimental group also trained Haru for creating newly encountered (those that were not encountered during the boostrapping) states (labeled with yellow check signs in Figure 8).

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	Mean
Rejections	159	84	108	90	98	133	100	177	96	147	89	102	97	113.8
Mean	15.9	8.4 10.8	9.0	9.8	13.3	10.0	17.7	9.6	14.7	8.9	10.2	9.7		
SD	17.84	7.53	9.81	9.30	8.24	2.75	8.65	12.33	9.94	5.58	7.69	8.55	5.40	

TABLE III: Total number of rejections for 10 states to learn the optimal behavior for all participants in the online learning without bootstrapped model.

No	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	Mean
Rejections	41	23	84	59	4	47	20	39	61	111	124	64	122	71.46
Mean	4.1	2.3	8.4	5.9	0.4	4.7	2	3.9	6.1	11.1	12.4	6.4	12.2	
SD	3.07	1.57	7.11	4.51	1.26	5.33	1.63	3.35	7.69	8.43	10.07	8.28	12.04	

TABLE IV: Total number of rejections for 10 states to learn the optimal behavior for all participants in the online learning with bootstrapped model.

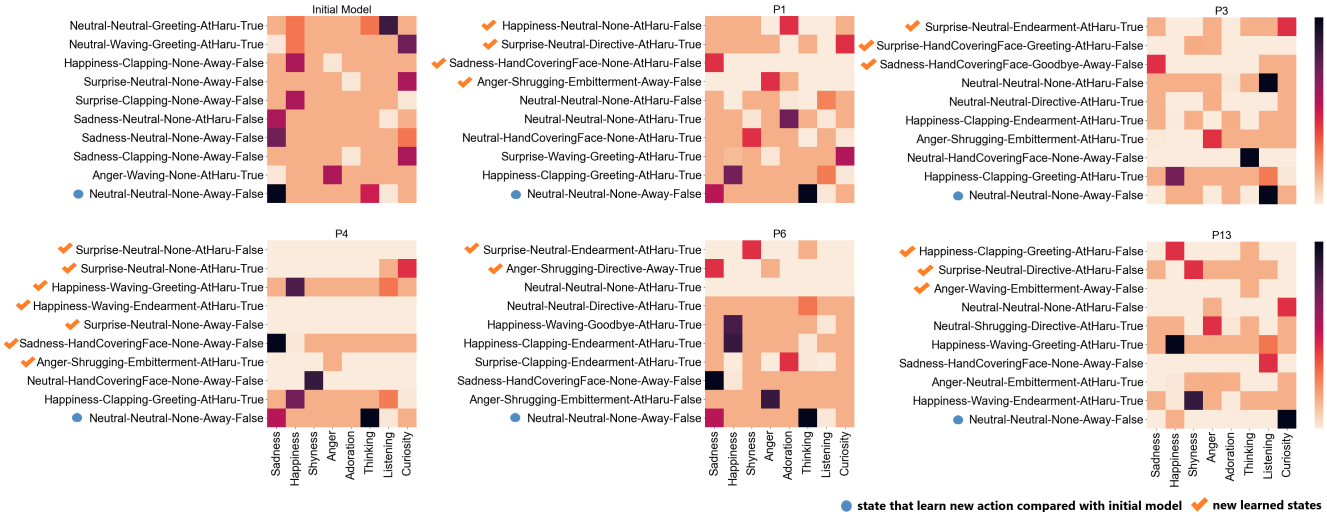

Fig. 8: Visualized heat map of final Q values for all actions in a snippet of states trained in the simulating environment and the ten states trained by participant 1, 3, 4, 6 and 13 in the online learning with bootstrapped model.

VII. Conclusion

In this paper, we have shown a method to train an agent to select optimal choices of behaviors from it's behavioral repertoire through its interaction with humans. First, the training was bootstrapped by experts using a specialized GUI to interact with the agent. The bootstrapping accelerated the training process and the corresponding bootstrapped model was then deployed to the robot with naive users, representing the actual end-users of the robot. Through the perception module, we provided a natural interface for the naive users to further personalize the robot with ease. Currently, the system is simple and prone to errors and sparsity of rewards. These dimensions are not sufficiently discussed and not evaluated in this paper. Also our approach currently does not allow a fine-grained modification of the actions using such discrete human feedback. In the future we plan to investigate these questions more closely. In addition, we plan to explore novel training methods and increase the number of robot actions and input modalities.

References

[1] "Haru: An experimental social robot from honda research." http://spectrum.ieee.org/automaton/robotics/home-robots/haru-an-experimental-social-robot-from-honda-research. accessed: 2020-02-02.

[2] D. J. Haraway, *The companion species manifesto: Dogs, people, and significant otherness*, vol. 1. Prickly Paradigm Press Chicago, 2003.

[3] R. Gomez, D. Szapiro, K. Galindo, and K. Nakamura, "Haru: Hardware design of an experimental tabletop robot assistant," in *Proceedings of the 2018 ACM/IEEE international conference on human-robot interaction*, pp. 233–240, 2018.

[4] R. Gomez, D. Szapiro, L. Merino, and K. Nakamura, "A holistic approach in designing tabletop robot's expressivity," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1970–1976, IEEE, 2020.

[5] H. Brock, J. P. Chulani, L. Merino, D. Szapiro, and R. Gomez, "Developing a lightweight rock-paper-scissors framework for human-robot collaborative gaming," *IEEE Access*, vol. 8, pp. 202958–202968, 2020.

[6] E. Nichols, L. Gao, and R. Gomez, "Collaborative storytelling with large-scale neural language models," in *Motion, Interaction and Games*, pp. 1–10, 2020.

[7] R. Gomez, D. Szapiro, L. Merino, H. Brock, K. Nakamura, and S. Sabanovic, "Emoji to robomoji: Exploring affective telepresence through

haru," in *International Conference on Social Robotics*, pp. 652–663, Springer, 2020.
[8] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," 2011.
[9] G. Li, R. Gomez, K. Nakamura, and B. He, "Human-centered reinforcement learning: a survey," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 4, pp. 337–349, 2019.
[10] A. L. Thomaz and C. Breazeal, "Teachable robots: understanding human teaching behavior to build more effective robot learners," *Artificial Intelligence*, vol. 172, no. 6, pp. 716–737, 2008.
[11] W. B. Knox and P. Stone, "Interactively shaping agents via human reinforcement: the TAMER framework," in *Proceedings of the 5th International Conference on Knowledge Capture*, pp. 9–16, ACM, 2009.
[12] G. L. H. B. K. N. I. P. Yurii Vasylkiv, Zhen Ma and R. Gomez, "Automating behavior selection for affective telepresence robot," in *International Conference on Robotics and Automation*, Springer, 2021.
[13] C. C. White, "A survey of solution techniques for the partially observed markov decision process," *Annals of Operations Research*, vol. 32, no. 1, pp. 215–230, 1991.
[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
[15] I. Farag and H. Brock, "Learning motion disfluencies for automatic sign language segmentation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7360–7364, IEEE, 2019.
[16] H. Brock, S. Sabanovic, K. Nakamura, and R. Gomez, "Robust real-time hand gestural recognition for non-verbal communication with tabletop robot haru," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 891–898, IEEE, 2020.
[17] H. Brock, I. Farag, and K. Nakadai, "Recognition of non-manual content in continuous japanese sign language," *Sensors*, vol. 20, no. 19, p. 5621, 2020.
[18] K. Nakamura and R. Gomez, "Improving separation of overlapped speech for meeting conversations using uncalibrated microphone array," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 55–62, IEEE, 2017.
[19] F. Ge and Y. Yan, "Deep neural network based wake-up-word speech recognition with two-stage detection," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2761–2765, IEEE, 2017.
[20] A. H. Michaely, X. Zhang, G. Simko, C. Parada, and P. Aleksic, "Keyword spotting for google assistant using contextual speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 272–278, IEEE, 2017.
[21] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Eighth International Conference on Weblogs and Social Media (ICWSM)*, pp. –, 2014.
[22] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
[23] J. Lin, Z. Ma, R. Gomez, K. Nakamura, B. He, and G. Li, "A review on interactive reinforcement learning from human social feedback," *IEEE Access*, vol. 8, pp. 120757–120765, 2020.
[24] G. Li, H. Hung, S. Whiteson, and W. B. Knox, "Learning from human reward benefits from socio-competitive feedback," in *4th International Conference on Development and Learning and on Epigenetic Robotics*, pp. 93–100, IEEE, 2014.
[25] G. Li, S. Whiteson, W. B. Knox, and H. Hung, "Social interaction for efficient agent learning from human reward," *Autonomous Agents and Multi-Agent Systems*, vol. 32, no. 1, pp. 1–25, 2018.
[26] E. Even-Dar, Y. Mansour, and P. Bartlett, "Learning rates for q-learning.," *Journal of machine learning Research*, vol. 5, no. 1, 2003.
[27] W. Liu, "Natural user interface-next mainstream product user interface," in *2010 IEEE 11th International Conference on Computer-Aided Industrial Design & Conceptual Design 1*, vol. 1, pp. 203–205, IEEE, 2010.