IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 July 2022, accepted 11 September 2022, date of publication 30 September 2022, date of current version 10 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3210932

RESEARCH ARTICLE

Sample-Efficient Training of Robotic Guide Using Human Path Prediction Network

HEE-SEUNG MOON AND JIWON SEO, (Member, IEEE)
School of Integrated Technology, Yonsei University, Incheon 21983, Republic of Korea

Corresponding author: Jiwon Seo (jiwon.seo@yonsei.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2018R1D1A1B07043580; in part by the Unmanned Vehicles Core Technology Research and Development Program through the National Research Foundation of Korea (NRF) and by the Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science and ICT, Republic of Korea, under Grant 2020M3C1C1A01086407.

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

ABSTRACT Training a robot that engages with people is challenging; it is expensive to directly involve people in the training process, which requires numerous data samples. This paper presents an alternative approach for resolving this problem. We propose a human path prediction network (HPPN) that generates a user's future trajectory based on sequential robot actions and human responses using a recurrent-neural-network structure. Subsequently, an evolution-strategy-based robot training method using only the virtual human movements generated using the HPPN is presented. It is demonstrated that our proposed method permits sample-efficient training of a robotic guide for visually impaired people. By collecting only 1.5 K episodes from real users, we were able to train the HPPN and generate more than 100 K virtual episodes required for training the robot. The trained robot precisely guided blindfolded participants along a target path. Furthermore, using virtual episodes, we investigated a new reward design that prioritizes human comfort during the robot's guidance without incurring additional costs. This sample-efficient training method is expected to be widely applicable to future robots that interact physically with humans.

INDEX TERMS Blind navigation, evolution strategy, human–robot interaction, recurrent neural network, robotic guide.

I. INTRODUCTION

One of the most important advantages that robots provide to human society is that they can complement the limited sensory and physical abilities of their users. The use of a robotic guide, which offers navigational aid to visually impaired users, clearly illustrates this benefit. Most visually impaired people have difficulty moving freely in unfamiliar spaces. In the absence of visual recognition, users can rely on directional guidance from a robotic guide and achieve a safe route without being disrupted by unknown obstacles. This type of assistive robot has various valuable applications, such as providing assistance to blind and elderly people with impaired visuomotor skills or guiding people from visually impoverished areas (e.g., during a disaster).

For robots engaging with humans (e.g., robotic guides), the ability to understand a user's behavior, that is, social intelligence, is essential [1], [2], [3], [4], [5], [6]. This allows robots to make highly sophisticated movements without reducing their usability and allows users to perceive more credibility from the robots. A robotic guide with such social intelligence can accurately predict a user's movement and guide the user to follow a precise path. This ensures the safety of visually impaired people even in narrow and crowded places. Furthermore, robot actions planned based on predicted human movements can lead to increased usability of a robotic guide, for example, inducing smooth user movements and avoiding excessive movements that put stress on the user's muscles.

The associate editor coordinating the review of this manuscript and approving it for publication was Angel F. García-Fernández.

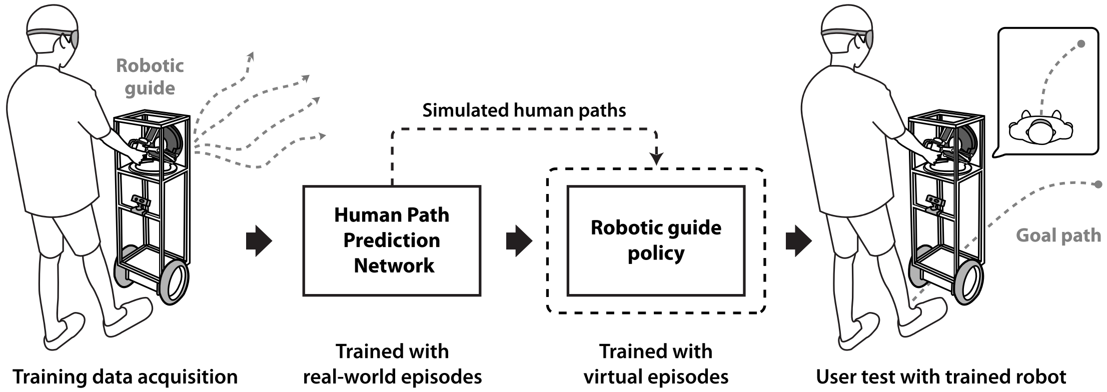
Robotic guide
Simulated human paths
Human Path Prediction Network
Robotic guide policy
Goal path
Training data acquisition
Trained with real-world episodes
Trained with virtual episodes
User test with trained robot

FIGURE 1. Procedural overview of our training method for a robotic guide. We first train a human path prediction network with a limited number of episodes, and then train the robot policy using numerous virtual episodes generated based on the human path prediction network. The trained robotic guide is validated through a user test to guide users along the given goal path.

In the past, social intelligence was usually provided to robots through handcrafted models [7]. However, recent deep learning-based computing methods have made it possible to provide robots with more advanced intelligence. There are two representative methods for training a robot's action policy: reinforcement learning (RL) and evolution strategy (ES) methods. On the one hand, RL methods develop the policy of an agent (e.g., a robot) by making the agent interact with the given task environment and gradually updating its policy (e.g., gradient descent). On the other hand, ES methods find the optimal policy through a black-box optimization process that generates numerous candidate solutions and evaluates them. By involving a neural network in an RL or ES method, recent studies have demonstrated significant results for training robots in terms of locomotion [8], [9], [10], object grasping [11], [12], and finger dexterity [13].

However, there are critical problems in applying robot-training methods in real-world situations, particularly where humans are required to engage with robots. The following two major problems are posed. First, considerable amounts of time and spatial resources are required: While recent machine learning-based methods show promise, their sample efficiency is limited by the necessity of 'trial and error' processes. Typical agent training using deep RL methods requires millions of data samples [8], [11], making it practically impossible to collect such a large number of data samples when a human user is engaged. Second, it is difficult to ensure the safety of the human partners participating in the training process. The robot's movements during the early stage of the RL or ES training period are unrefined because the robot must go through an 'exploration' process of taking various random actions and searching the sample space that can yield high cumulative rewards. Therefore, human involvement in such unpredictable robot movements can cause discomfort to users or, more seriously, lead to a dangerous situation.

In this paper, we present a training method for a robotic guide to compensate for both the above problems, as illustrated in Figure 1. In particular, we propose a human path prediction network (HPPN) that predicts the next human movement based on the robot's sequential actions and the user's response data. The trained HPPN can generate an infinite number of virtual human movements. Therefore, a robotic guide can be trained using the ES method with numerous self-simulations based on virtual human movements. Our method does not directly involve people in training robot policies (RL or ES), but requires only a relatively small amount of movement data for training the HPPN. Moreover, our robot-training process is safe because the training dataset for the HPPN can be obtained from a general human guidance situation, and all inevitable trial-and-error processes for robot training are enacted only through a virtual simulation.

Another benefit of our approach is that we can repeatedly perform robot training according to different reward designs, without additional human involvement. Using infinitely generated data based on an HPPN, various types of intelligence for a robotic guide pursuing different objectives can be acquired. For example, one robotic guide policy might prioritize only the accuracy of the guidance (e.g., how precisely the assisted user follows the given path), while another robot policy might also take the comfortable movement of the assisted user into account in addition to the accuracy by shaping the training objective (i.e., reward).

To validate the proposed method, we developed a robotic guide testbed that physically guides users to follow a given target path without using their vision. The users consistently receive kinetic guidance from the robot through a haptic device mounted on the robot. Our robot observes the user's torso movements using a depth camera and measures the responsive kinetic force from the user using the haptic device. The HPPN was trained to predict the user's next movement based on multimodal response data, and then our robot learned its optimal behavior using the ES method from virtual simulations. We trained the robot using two different reward designs: one that pursues guidance performance (speed and accuracy) only, and another that pursues comfortable user movement as well as the guidance. In an indoor setting where a motion capture system can precisely measure user trajectories, we conducted a user test using our robotic guide. Our results demonstrated that, despite the limited amount of data collection, the trained robot could precisely guide the participants along the target path. Furthermore, among the two reward designs, the robot policy trained with a reward that additionally considered human comfort led to smoother movements of the actual participants.

A. CONTRIBUTIONS

The contributions of this paper are summarized as follows:

- We proposed a human path prediction network (HPPN) and training method for robots engaging with human partners based on the HPPN. Our training method compensates for two critical problems: sample inefficiency and human safety.
- We developed a robotic guide testbed for blind navigation that collects multimodal human-response data and physically guides users based on the sensed human data.
- We validated our trained robotic guide through a user test with eight participants. We demonstrate that our method is effective in developing human-centered robots, achieving smoother user motion and better goal-related metrics (e.g., speed and accuracy).

II. RELATED WORK

A. ROBOTIC AIDS FOR BLIND NAVIGATION

Over the last few decades, the need for safe navigation for people with a visual impairment has been continuously raised. Since the pioneering development of robotic aids for blind navigation, such as NavBelt [14] and GuideCane [15], robotic assistive technologies have been steadily growing. NavBelt is a belt-typed robotic device equipped with ultrasonic sensors, which provides information regarding the detected obstacles around the user through acoustic feedback. Another type of robotic aid, GuideCane, is a mobile robot moving ahead of the user. Similar to the NavBelt, it detects obstacles through ultrasonic sensors and delivers the kinetic feedback to the user through an attached cane. Subsequent studies have utilized various advanced sensors, such as radio-frequency identification [16], ultra-wideband systems [17], laser scanners [1], [18], [19], and depth cameras [20], [21], [22]; and their robotic guides provide common functionalities of obstacle detection and avoidance [1], [16], [17], [18], [19], [20], [21], [23] and guidance along a given safe path [24], [25].

Recent deep neural-network-based computing methods have been applied to the robotic guides, contributing especially to robot vision. Regarding obstacle detection functionality, Poggi and Mattocia [20] developed a wearable device that detects and classifies objects ahead of blind users from depth images using the LeNet model architecture [26] based on a convolutional neural network (CNN). Niu *et al.* [27] presented another wearable device with the aim of assisting the blind users in opening a door. Using a stereo camera, the device detects the position of a doorknob and the hands of the users using a CNN-based model, and delivers audio feedback regarding which direction the users should move their hand. With regard to the function of guidance of a user along a given path, a robotic guide dog was developed by Chuang *et al.* [25], which recognizes a trail on the floor using a CNN-based architecture and determines the next robot motion along the trail.

Predicting human trajectory is another important factor to guide users in a precise and comfortable manner. Especially, for the blind navigation scenario, it is challenging to predict human trajectories accurately because human movements are easily affected by several factors, such as the user's acceptance of a navigational aid [28], situational contexts [29], and environmental factors [30]. There have been several machine learning-based approaches to analyze behavior of a user moving with a mobile robot and predict the user's next trajectory [3], [31], [32], [33]. However, how the learned user behavior model can be reflected in the navigational guidance of the robot and used to optimize the robot policy remains an open question. For an example of non-learning-based methods, Scheggi *et al.* [1] developed a robotic guide that detects the user's position using a depth camera and reduces the moving speed to suit the user's intention. One limitation of [1] is that the user's position data only applies to slowing and lifting the pace. The robotic guide in our study uses a neural-network-based model to predict human movements and reflects this prediction data to select the next action of the robot for more precise and comfortable guidance.

B. TRAINING OF AUTONOMOUS ROBOTS

With the development of artificial intelligence technology, there have been significant achievements in training robots to perform complex tasks on their own. RL algorithms have been perceived as a promising way to dealing with real-world robot manipulation tasks, such as, picking up objects [12], [34], [35], opening doors [36], [37], and locomotion [10], [38]. In addition to RL methods, recent studies have suggested that ES methods can be an effective alternative for robot training [39]. Although ES methods have an even lower sample efficiency than RL methods, ES methods are computationally efficient and have better exploration characteristics, which can provide robustness in robot training. Accordingly, a covariance matrix adaptation evolution strategy (CMA-ES) [40], which is one of the latest ES methods, has been applied to robot tasks, such as the locomotion of humanoid [41] and quadruped [42] robots.

As a problem with the RL and ES methods, struggling to collect large amounts of data from real robots, researchers have attempted to acquire training data from the virtual robot movements on a physics-based simulator [37], [38], [43], [44]. These simulator-based approaches have inherent errors between the simulation results and reality; however, researchers have succeeded in mitigating this problem by randomizing the simulation parameters [44] or adapting the parameters through a few real-world rollouts [37].

For the case of robots working with humans, the sample efficiency problem is still challenging because the collection of training data is more expensive and there is lack of physics-based simulators that fully embody the unpredictable nature of human behavior. Therefore, few studies applied RL or ES methods to train the robots that collaborate with people. Qureshi *et al.* [45] presented a humanoid robot that successfully learned when to extend its hand for a handshake with a human by applying a deep Q-network method. However, the robot had a long real-world training period of 14 days, showing the sample-inefficiency problem. Shafti *et al.* [46] also showed the possibility that the off-policy RL algorithm would be applied for robots to learn collaborative tasks with human partners in real-world environment.

A recent approach to addressing the sample efficiency problem is to model the movement of a user interacting with a robot, and then generate virtual data that can be used for robot training. Ghadirzadeh *et al.* [47] implemented a user's behavior model through Gaussian process and applied Q-learning approach to train a collaborative robot with modeled human behavior. Lathuilière *et al.* [48] also pretrained a deep Q-network on generated human movement dataset to optimize the robot's gaze control during human–robot interaction, and then went through a fine-tuning process with real-world data.

Besides, recent neural network-based generative models have been shown to be promising in synthesizing higher-dimensional data including realistic images. Using variational autoencoder (VAE) [49], Ha and Schmidhuber [50] succeeded in modeling complex environment surrounding an agent and training the agent only by simulated rollouts from the modeled environment. Their validation was limited to learning an agent that performs a video game in virtual environment; however, Thabet *et al.* [51] later showed that the approach of Ha and Schmidhuber [50] can be applied even in training real-world robots interacting with humans. Nevertheless, because the variation by Thabet *et al.* [51] focused on learning the robot's movement based on a given single-step image input, it is still unclear whether the approach in [50] would be effectively applicable to more realistic environment with time-series and multimodal human behavior data.

In this study, we present a sample-efficient learning method of the robotic guide that can provide safe navigation to real-world users, by optimizing the action policy of the robot using simulated rollouts. To achieve this goal, we propose the HPPN to model the multimodal dynamics of human behavior when following a robotic guide, including walking trajectory, kinetic force of a hand, and body posture, in a time-series manner, and combine the HPPN with the approach in [50].

III. ROBOTIC GUIDE TESTBED

Our robotic guide is designed to guide the user to walk along a goal path by continuously delivering kinetic feedback to the user. As shown in Fig. 2, we mounted an Omega.7 haptic device (Force Dimension) on a Stella B2 mobile robot (NTREX, Inc.). To use our robotic guide, the user is instructed to grasp the handle of the haptic device, and the movements of the robot can be transmitted to the user directly through the haptic device. Another issue of our robotic guide design is the acquisition of multimodal human response data according to the robot's movements. As the first measurement data, a backward-facing Xtion 2 RGB-D camera (ASUS) is mounted on our robotic guide and captures the depth image of the user's torso at a pixel resolution of 640 × 480. For the second measurement data, we record the kinetic force input of the user from the haptic device. To quantitatively measure the force exerted on the robot, we apply a spring system that returns more strongly to the origin as the handle of the haptic device moves away from the origin. Specifically, we set the handle to move only in a 2D horizontal plane, and a 2D spring system toward the origin with a spring constant of 500 N/m is implemented in the haptic device. While using the robotic guide, the multimodal human response data are collected every 250 ms, i.e., four times a second. In addition, our robotic guide is equipped with a laptop computer, which controls the mobile robot and the sensors, motion capture markers for precise tracking of robot movements, and a power bank for powering the haptic device.

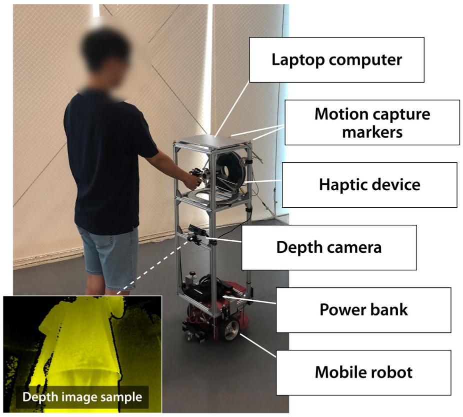

FIGURE 2. **Hardware overview of our robotic guide testbed.**

IV. ROBOT TRAINING METHOD

A. PROCEDURE

Through our robot training procedure, the participants do not directly participate in the robot's RL- or ES-based learning process, which requires numerous data samples and cannot guarantee the safety of the users from the robot's trial-and-error process. Instead, the participants are instructed to follow along various safe movements of the robotic guide, and we train the HPPN using the collected dataset. Following the approach of Ha and Schmidhuber [50], we utilized VAE [49] structure to extract low-dimensional latent feature vectors from the high-dimensional depth images. Accordingly, we train the VAE in advance using the depth images in the training dataset, and the HPPN is then trained using robot actions, human kinetic force data, and compressed depth data as input values. The HPPN can act as a simulator providing the expected human path, given only the action sequence of the robotic guide. Using the HPPN-based simulator, we discover the optimal policy parameters for determining the next action of the robotic guide that maximizes our reward formulation using the CMA-ES algorithm. To summarize, our training method is applied in the following order:

1) Obtain a dataset of the users following the robot that guides them through various paths.
2) Train the VAE using the depth images in the dataset.
3) Train the HPPN using the dataset, including the latent feature vectors extracted using the trained VAE.
4) On the simulator based on the HPPN, optimize the policy parameters to maximize the cumulative reward of the robotic guide using the CMA-ES algorithm.

B. DATA ACQUISITION

We conducted a data acquisition session for training our HPPN. Since it is difficult to invite a sufficient number of blind participants, data were obtained from non-disabled participants with blindfolded, which is also common in previous robotic guide studies [1], [18], [19]. In this session, 11 participants (2 females and 9 males) between the ages of 20 to 26 years (M = 23.64, SD = 1.67) were involved. All participants were right-handed. They were informed in advance regarding the purpose of the experiment and were free to rest when needed. During the session, blindfolded participants were instructed to move under the guidance of the robot in an open indoor environment without knowing the path of the robot in advance (Fig. 3). In the indoor environment, nine motion capture cameras were installed. Motion capture markers were attached on the robot and the participants; therefore, robot and human path data were collected with mm-level accuracy. Overall, the following time-series data were acquired: i) sets of human and robot movements, i.e., the change of the position and heading angle per timestep, ii) kinetic forces and depth images, measured from the robot sensors and iii) action commands given to the robot, i.e., goal speeds of the left and right wheels, for each timestep.

According to Article 15 (2) of the Bioethics and Safety Act and Article 13 of the Enforcement Rule of Bioethics and Safety Act in Korea, a research project "which utilizes a measurement equipment with simple physical contact that does not cause any physical change in the subject" (Korean to English translation by the authors) is exempted from the approval. The entire experimental procedure was designed to use only a haptic device and a depth camera that did not cause any physical changes in the subject.

We aimed at obtaining human data based on as much diverse robot movements as possible, without drastic changes in the robot movements causing discomfort in the participants. To do so, the robot randomly chose its own left and right wheel speeds, but instead of changing the action for each timestep, it set the action to be maintained for arbitrary timesteps. In detail, the rotation speeds of the robot wheels were set between 2.5 to 5 rad/s (0.19 to 0.38 m/s), and the robot retained the action for 4 to 20 timesteps (for 1 to 5 s) before selecting the next action. This method allowed the robot to generate a smooth and diverse guidance route. One episode, which refers to one full guidance of the robot moving with a participant, took an arbitrary time of between 15 and 30 s. During the three-hour experiment, all participants were involved in 140 episodes. In total, 1,507 episodes of data were acquired, excluding data in which the robot and human paths were beyond the motion capture range and were therefore not fully measured.

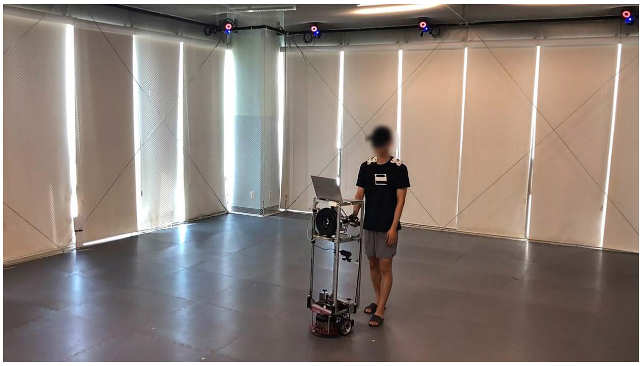

FIGURE 3. Experiment setup for training data acquisition. A blindfolded participant follows the robotic guide in an indoor environment with a motion capture system.

C. HUMAN PATH PREDICTION NETWORK

We implemented an HPPN for our robotic guide based on long short-term memory (LSTM) networks [52], an improved variant of recurrent neural networks. We extended the structure of the model described in our previous work [32], which was used to predict only the relative position of a person from a robot. Fig. 4 (a) shows an overview of our network. Our network employs the following input data: i) robot actions (two-dimensional), ii) kinetic forces applied to the haptic device (two-dimensional), and iii) latent feature vectors of depth images extracted from the pre-trained VAE (16-dimensional). We used the VAE with the structure described in our previous work [53] to compress a depth image with a pixel resolution of 640 × 480 into a 16-dimensional vector.

Structurally, our network consists of two independent parts. One part is only responsible for the movement of the robot, and the other part is responsible for the movement of the user moving with the robot. The first part (left side of Fig. 4 (a)) uses the robot's action only as an input, and outputs the next robot movement, i.e., the change of the position and heading per timestep, using two layers of the LSTM, which have eight hidden units each. Unlike the first part, which only deals with robot-related data, the second part (right side of Fig. 4 (a)) focuses on predicting human motion using both the robot action and the human response, namely the kinetic force and latent vector from the depth image. Two layers of LSTMs with 64 hidden units are used to predict

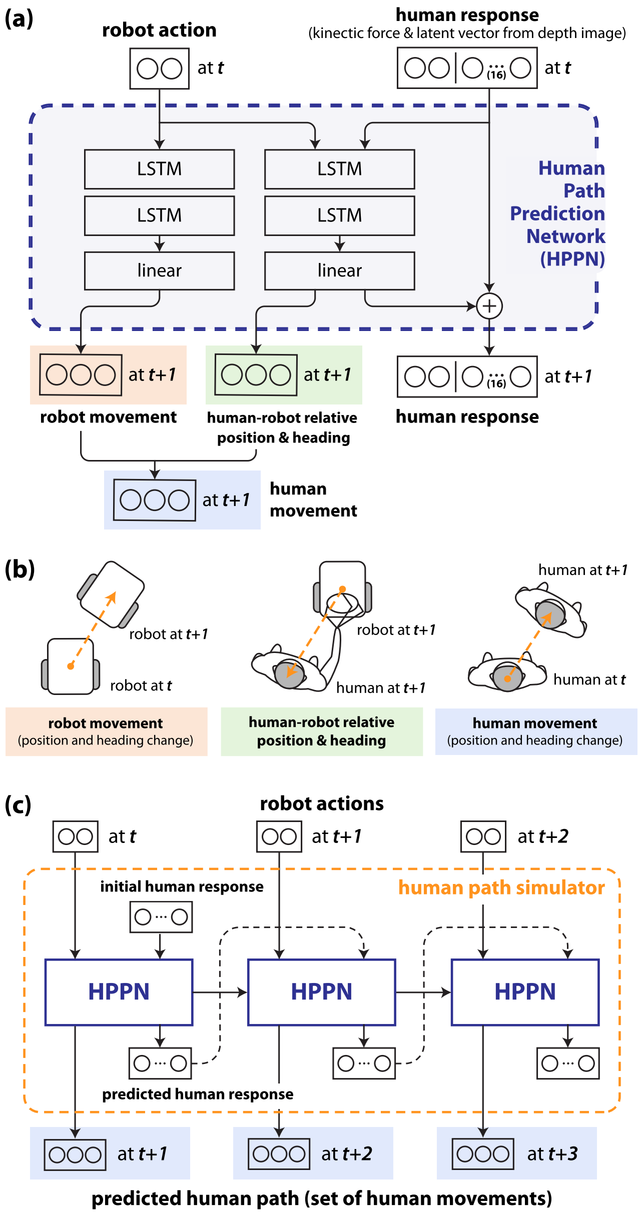

FIGURE 4. (a) Overview of our HPPN. The number of the circles in the box represents the size of each data. (b) Schematic representations robot movement, human–robot relative position and heading, and human movement. (c) The use of an HPPN as a simulator. The expected human path can be obtained by inputting the sequence of the robot actions.

the human–robot relative position and heading at the next timestep. The network also predicts the next human response to create a simulator that predicts human movements using only the robot actions, as shown in Fig. 4 (c), by applying a structure that uses the predicted human response as a new input for the next timestep. We applied a residual connection between the human response input and output, and therefore the network only predicts the amount of change, thus enabling more stable training. By obtaining robot movements in the first part and human–robot relative positions and headings in the second part, we acquire the predicted human movements as timestep progresses.

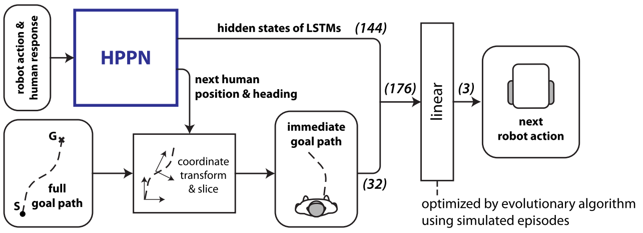
robot action & human response
HPPN
hidden states of LSTMs
(144)
next human position & heading
(176)
(3)
linear
next robot action
G
S
full goal path
coordinate transform & slice
immediate goal path
(32)
optimized by evolutionary algorithm using simulated episodes

FIGURE 5. Process by which our robotic guide determines the next action according to the input data (current action, human response and goal path) and the policy parameters optimized using the human path simulator in Fig. 4 (c). The numbers in parentheses indicate the dimensions of each data.

In detail, we used approximately 130K depth images in the training dataset to train the VAE, which was trained for 200 epochs using an Adam optimizer (learning rate of 0.001). For the HPPN, we trained the dual structures independently. By windowing the data with a size of 20 timesteps (i.e., 5 s), approximately 100K time sequential data samples were used to train the HPPN. Using the Adam optimizer at a learning rate of 0.01, the robot- and human-related LSTM structures were trained for 500 and 80 epochs, respectively.

D. TRAINING OF ROBOTIC GUIDE ON SIMULATOR

We applied the CMA-ES method to train a robot policy with the generated episodes based on the human path simulator. The CMA-ES method is an evolutionary optimization algorithm that finds a solution that maximizes the objective function of a particular problem. During every generation, the method performs episodes with a population size of the candidate solutions and develops candidate solutions in a way that yields a better objective function over a generation. We implemented our robot policy using parameters that can be optimized using the CMA-ES method and defined rewards for the robot that can be used as the objective function, as described in the following sections. Using the human path simulator, a human path when the robot moves based on the current policy with the given parameters can be predicted. Under the CMA-ES method, the policy parameters improve using the reward calculated from the predicted human path, and this process is repeated to find the optimal policy parameters.

1) ROBOT POLICY

During every timestep, our robotic guide determines the action to take in the next timestep by passing a robot's current feature vector through a single linear layer as follows:

$$(\textit{next robot action vector}) = \mathbf{W} \cdot (\textit{feature vector}) + \mathbf{b},$$

where $\mathbf{W}$ and $\mathbf{b}$ represent the weight and bias of the linear layer, and are the policy parameters optimized through the CMA-ES. The feature vector of the robot consists of two major features, as shown in Fig. 5. First, to contain the motion information of the user and the robot over time, we used the hidden state values of the LSTMs in the HPPN at the current timestep. Because the LSTMs were designed to predict the next human movement and affected by the sequential input data (robot actions and human responses), the hidden states of the LSTMs will integrate this information and are useful to reflect it in the robot's next action. Second, the information of the goal path that the user should walk along is provided to the robot to determine the next action. Instead of using a full goal path directly, only the immediate goal path for the current timestep is employed as the timestep passes. For this purpose, the coordinate transformation and slicing processes are performed at every timestep based on the predicted human position and heading obtained from the HPPN.

The feature vector we used has a size of 176, which is obtained by flattening the hidden state values of the LSTMs with a size of 144 (2 × 8 hidden units + 2 × 64 hidden units) and the immediate goal path data consisting of 16 two-dimensional points. The robot's action vector consists of three dimensions: the left motor speed, right motor speed, and degree ranging from 0 to 1 to determine whether to stop (the robot stops if the degree is over 0.5). Accordingly, our policy parameter has a size of 531, including a weight matrix with a size of 3 × 176 and a three-dimensional bias vector.

2) REWARD

In this study, we set up two reward types: i) a reward that only considers goal efficiency (referred to as *G Only*) and ii) a reward that also considers how the robot comfortably guides the user in addition to the goal efficiency (referred to as *G* + *H*). In detail, the cumulative rewards of an episode based on these two reward types were calculated as follows:

- *G Only*: This reward is determined based on the metrics related only to the goal efficiency, i.e., the completion time and accuracy. As a metric to measure the level of accuracy, the Fréchet distance, which calculates the similarity between two paths, was used to indicate how accurately the user actually moved along the goal path. The total calculation formula is as follows:

  $$\begin{aligned}(cumulative\ reward) = &-1 \times (completion\ time)\\ &- 100 \times (Fr\acute{e}chet\ distance).\end{aligned}$$

  In addition, if the completion time exceeded the maximum timestep (which we set to 100 timesteps), the episode was terminated and a penalty of −500 was given.
- *G* + *H*: In addition to the metrics used for *G Only*, we applied human motion smoothness to consider the comfort of the user. As a quantitative measure of the motion smoothness, we used the spectral arc length [54]. The calculation formula is as follows:

$$\begin{aligned}&(cumulative\ reward)\\ &\quad = -1 \times (completion\ time)\\ &\qquad - 100 \times (Fr\acute{e}chet\ distance)\\ &\qquad + 30 \times (spectral\ arc\ length\ of\ \ human\ path).\end{aligned}$$

By setting up the two different rewards as above and evaluating them through user experiments, we can indicate the following two aspects. First, our trained HPPN have the advantage of training various types of robot policies that are optimal in different reward formulations by utilizing the fact that the simulator based on the HPPN can generate countless virtual human paths. Second, it can be confirmed whether the generated episodes based on the HPPN are effective for the development of real-world human-centered robots by comparing the actual performance of the two robot policies trained using the *G* + *H* reward, which considers human convenience, and the *G Only* reward, which does not.

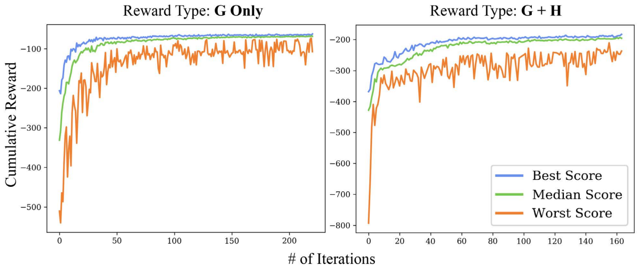

FIGURE 6. Convergence of the cumulative reward of our robot policy trained using the CMA-ES method with the *G Only* (left) and *G* + *H* (right) type rewards.

3) IMPLEMENTATION DETAILS

For the CMA-ES method, we set 32 population sizes for each generation, which means that 32 different policy parameters were tested using the virtual simulations from one generation. For each policy parameter, we simulated 16 episodes based on 16 randomly generated goal paths of 4 m in length, and collected their average cumulative rewards. Fig. 6 shows the robot policy training results for each reward type using this CMA-ES method. As a result of training applied until the objective functions of the best performer converged, it took 219 generations for *G Only*, and 163 generations for *G* + *H*. These figures indicate that 112,128 virtual episodes (219 × 32 × 16) were conducted for *G Only*, and 83,456 (163 × 32 × 16) were conducted for *G* + *H*. Considering that only 1,507 episodes were actually collected, our training method shows that the ES approach can be applied with a notably better sample efficiency.

V. USER TEST

We conducted a user test to validate our robotic guide trained with the two reward types mentioned above. Eight participants (1 female and 7 males) between 24 to 29 years of age (M = 25.75, SD = 1.48) were involved in the user test. None of them participated in the training data acquisition session, and all were right-handed. Similar to the preceding training session, each participant was blindfolded and instructed to follow the robot's guidance without knowing the path information. Using the motion capture system, the precise path data of the participants were collected with mm-level accuracy to analyze the performance of our robotic guide. Note that the motion capture system that generated the ground truth was used only for the evaluation purpose during the user test.

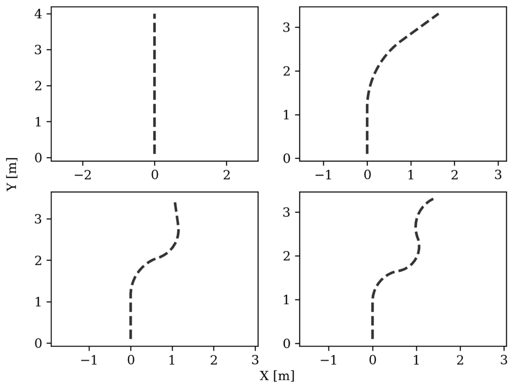

FIGURE 7. Goal paths used for the user test. All paths are 4 m long and created by adjusting the number of curvatures from zero to three. By laterally inverting the goal paths except for the straight line, seven goal paths were used for the user test.

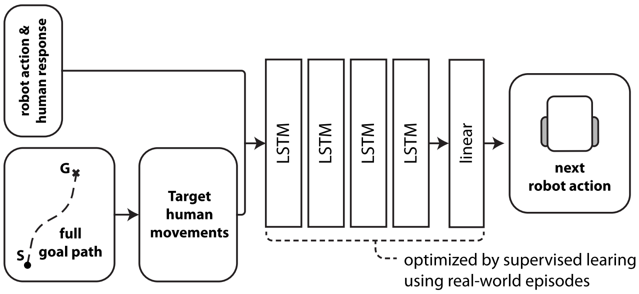
robot action & human response
G
S
full goal path
Target human movements
LSTM
LSTM
LSTM
LSTM
linear
next robot action
optimized by supervised learing using real-world episodes

FIGURE 8. Overview of the baseline policy network.

The trained robot does not utilize any information from the motion capture system when guiding the user.

While the robot was set to take randomly generated movements during the training session, the trained robot policies were applied to guide the participants to follow the goal path during this user test. To investigate the performance of the robotic guide according to the path complexity, we generated various goal paths, controlling the number of curvatures in the path. As depicted in Fig. 7, we constructed 4 m long goal paths with zero to three curvatures. By laterally inverting the three goal paths except for the straight line, seven goal paths were used for the user test.

In this user test, three robot policies were evaluated: a baseline policy, described below, and our two robot policies trained using the *G Only* and $G + H$ rewards. All participants conducted three episode trials for each of the 21 pairs of robot policy-goal path. Accordingly, each participant of this user test performed 63 episodes (7 goal paths × 3 robot policies × 3 trials) during a 1.5 h long experiment. Because the order of each robot policy-goal path pair was determined randomly, the participants were unable to know what conditions they were currently being guided under.

A. BASELINE POLICY

As a baseline policy, we developed a neural-network-based model that directly outputs the robot's next action from input data, which is the best way to train a robot with a limited amount of data and has shown effective performance in recent studies [25], [55]. As mentioned earlier, the conventional RL and ES methods suffer from the sample inefficiency problem. Thus, it is evident that the training with those methods will not be completed using the very limited number of real-world data samples used in this study. Therefore, a neural-network-based model, which can be trained using supervised learning even with the limited dataset we acquired, is implemented as our baseline policy. The performance of our training method based on the proposed HPPN was compared with that of the baseline policy.

The baseline policy network consists of four LSTMs with eight hidden units, as shown in Fig. 8. The network has the same input (current robot action, human response, and goal path) and output (next robot action) data as the robot policy we presented in Fig. 5. Because the training dataset consists of the actual distances that a person moved rather than a goal path, we delivered the goal path information in the form of target human movements that the robot should guide. Accordingly, an additional process that converts a given goal path into plausible target human movements is required for the user test. For the processing, we set the target speed of the user to 0.36 m/s, which corresponds to the top 80% of the distribution of human speeds obtained from the training dataset.

VI. RESULTS

We evaluated the performance of the robotic guide from two perspectives: goal efficiency and user comfort. In terms of the goal efficiency, the completion time and path error were used as the evaluation metrics. The completion time of the robot guidance was measured as the time taken from the start to the stop of the robot. To quantify the path error, the Fréchet distance between the goal path and the actual path of the user was measured. For the user comfort, we focused on how comfortable the user could move under the guidance of the robot. A smooth movement of a person has been identified as the result of minimized effort [56], [57]. Therefore, we utilized the spectral arc length to measure the smoothness of the human movements.

Fig. 9 shows box plots of the distributions of the above three metrics measured during all episodes of the user test. Through the paired t-test (eight participants), we verified whether one policy led to a significant difference in the guidance performance over another. In terms of the completion time, all three robot policies required a similar amount of time to guide the users. Note that, during the user test applying the same goal path length, the baseline policy consistently resulted in the same completion time. Because a fixed target user speed was utilized for the baseline policy, a fixed-length input data sequence was provided to the robot. Therefore, the robot with the baseline policy ended the guidance after the same amount of time. In terms of the path error, both the *G Only* and $G + H$ type robot policies induced significant decrease of path error compared to the

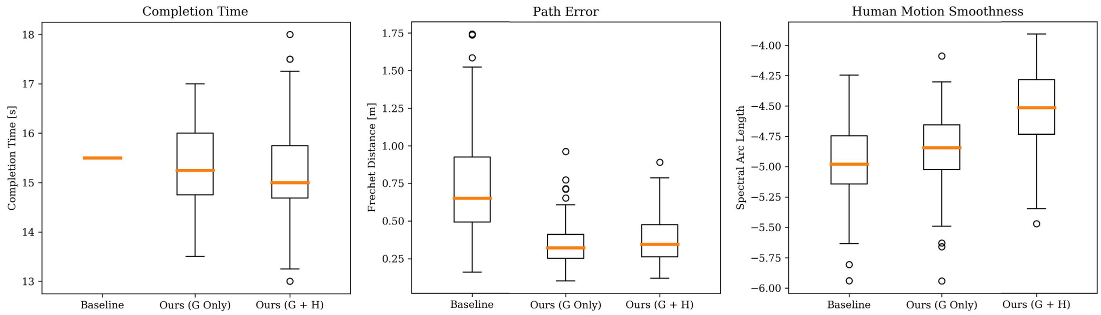

FIGURE 9. **Box plot showing the overall performance results of each robot policy.**

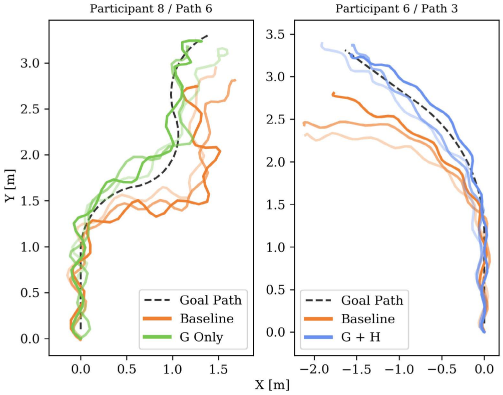

FIGURE 10. **Samples of the user's actual path following the robot with the baseline policy and the *G Only* (left) and *G* + *H* (right) type policies. User paths during different trials are distinguished by the line transparency.**

baseline policy ($t = 9.931$, $p < 0.001$ and $t = 8.478$, $p < 0.001$, respectively). No significant differences were found between the *G Only* and $G + H$ type robot policy ($t = 1.361$, $p = 0.216$). In terms of the smoothness of the human motion, the results of the $G + H$ type policy significantly outperformed those of both the *G Only* type policy ($t = 8.236$, $p < 0.001$) and the baseline policy ($t = 12.975$, $p < 0.001$), whereas the *G Only* type policy and the baseline policy did not show a significant difference ($t = 2.175$, $p = 0.066$). This indicates that the robot policy training method, which additionally considered motion smoothness, led to smoother movements of participants.

Fig. 10 shows samples of the user's actual path when following the robot using the baseline policy and that of our robot policies (*G Only* and $G+H$). It is clearly shown that the user's path is closer to the dotted goal path, when using our robot policies. The oscillatory behavior of the user is a natural response to stepping on the left and right foot and has been reported in [58]. The trajectory of the robot does not have this oscillation, which is shown in Fig. 11.

A. HOW THE ROBOT ADEQUATELY GUIDES DIFFERENT USERS

Our robotic guide was designed to reflect human multimodal response data to better guide people to a goal path at each timestep. In other words, the robot learns to recognize the user's behavioral patterns from the multimodal response data and choose the best action based on the expected path of the user. We examined episodes from the user test to see how our robotic guide adequately guides different users.

Fig. 11(a) shows the path samples of the robots and humans during episodes in which the *G Only* type robot policy guided two different participants to the same goal path. In the left sample (a-1), participants 1 and 6 showed different behavioral patterns in following the robot. Participant 1 tended to walk on the right side of the robot ahead, whereas participant 6 tended to walk on the left. Accordingly, our robotic guide showed different path movements for each participant. When guiding participant 1, the robot led to the left side of the goal path, and when guiding participant 6, the robot led to the right side of the goal path, consequently guiding the users closer to the goal path. These movements of the robot also appeared in another episode (a-2). While guiding along the same goal path, participant 2 walked along the left side of the robot, and participant 5 followed almost the same path as the robot. Accordingly, the different guidance paths of the robot were observed, moving toward the right side of the goal for participant 2, and moving as close as possible toward the goal path for participant 5. These episodes show that our trained robot effectively learned the different behavioral patterns of humans and chose adequate actions for different users.

B. HOW THE ROBOT COMFORTABLY GUIDES USERS

During this user test, the $G + H$ type robot policy showed a superior smoothness of the human motion compared to the baseline and *G Only* type policies. We further analyzed how the $G + H$ type policy, which was trained on virtual simulations, improved the motion smoothness of the actual users. Our analysis proceeded in two ways, in terms of the actual guiding path and the change in speed of the robot.

First, we compared the actual robot paths generated by the *G Only* and $G + H$ type robot policies for the same participants based on the two path samples in Fig. 11(b). In both path samples, we observed that the $G + H$ type robot gave up turning sharply and chose a less convoluted path, although a sharp turn was required to follow the sharply turning goal path. In the path samples on the left (b-1), the $G + H$ type

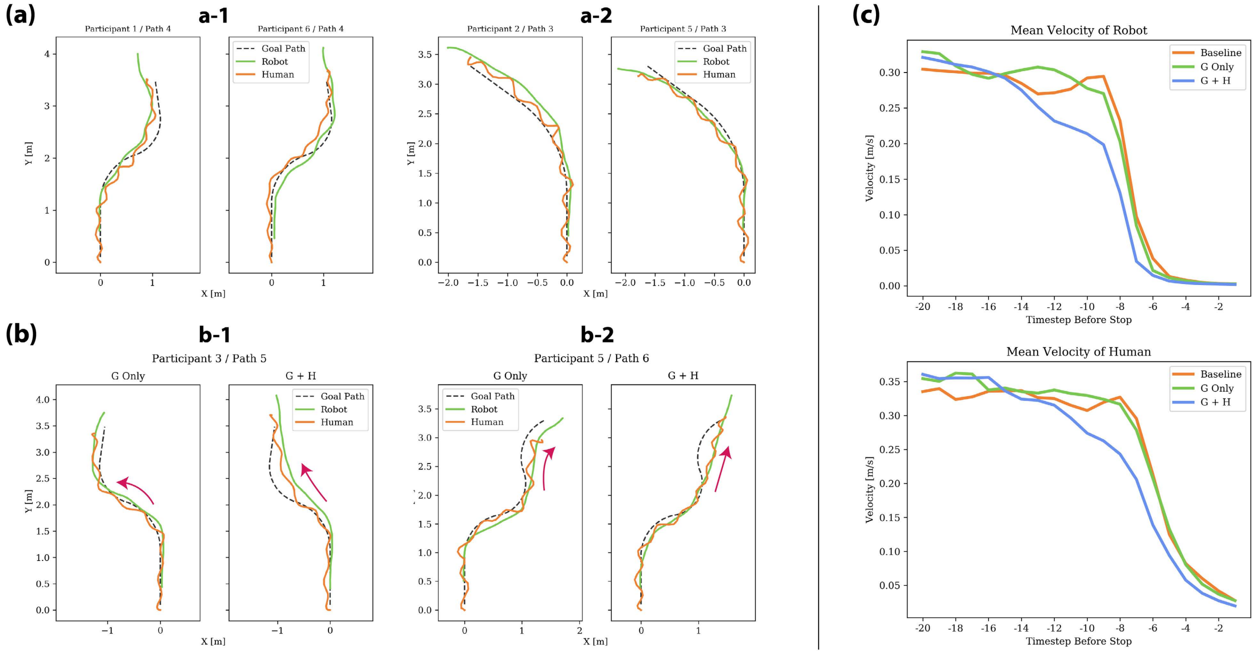

FIGURE 11. (a) Path samples of the robot using the *G Only* policy when moving differently for different participants for the same goal path. (b) Path samples showing how robots under the *G Only* and *G* + *H* type policies behave differently for the same participant and the same goal path. (c) The mean velocity changes of the robot (top) and the user (bottom) before the robot stops.

robot rotated only slightly along the goal path, whereas the *G Only* type robot rotated obviously toward the left along the goal path (as depicted by the red arrows). In the path samples on the right (b-2), the $G + H$ type robot gave up the last turn and went straight, whereas the *G Only* type robot moved along the goal path to the end (as depicted by the red arrows). These examples demonstrate that the $G + H$ type policy learned to avoid sharp rotations that can lower the smoothness of the human motion, even if a certain level of accuracy is lost. However, such movements of the $G + H$ type robot that noticeably reduce accuracy as in Fig. 11(b) were rarely observed and, as shown in Fig. 9, did not significantly reduce guidance accuracy compared to the *G Only* type robot in general.

For the second aspect, we analyzed for all three policies the mean velocities of the robot and the user for 20 timesteps (i.e., 5 s) before the robot stopped, as shown in Fig. 11(c). The analysis showed that there were no noticeable differences between the baseline and *G Only* type policy; however, the $G + H$ type robot started decelerating earlier than the other two policies, and gradually slowed down. The other two policies started the deceleration of the robot at 8 to 10 timesteps prior to stopping, whereas the $G + H$ type policy started the deceleration at 12 to 14 timesteps before it stopped. Accordingly, the speed of the user following the robot also decreased early and slowly under the $G + H$ type policy as compared to the other two policies. Consequently, this change in speed demonstrates that the $G + H$ type policy increased the smoothness of the human motion by learning its own gradual slowdown process prior to stopping.

VII. CONCLUSION

In this paper, we introduced the HPPN, a neural network model that predicts human movements with regard to sensed human response data, and the training method for the robotic guide based on episodes generated by the HPPN. Our approach is beneficial because it addresses concerns regarding human safety and sample inefficiency that arise when training robots to collaborate with humans. We collected 1,507 real-world episodes for training the HPPN, and then it was possible to generate over 100,000 virtual episodes to optimize the action policy of the robotic guide. The user test results indicated that our method is effective in training the robotic guide with increased guidance accuracy compared with the baseline method that used the same amount of real-world training data. In addition, using infinitely generated episodes, we can investigate various reward formulations to achieve a highly human-centered robot policy. For example, one of the remarkable points of our work is that, by utilizing the reward formulation that values human comfort, the trained robotic guide actually yielded improved smoothness in human motion during a real-world user test.

This study has several promising future extensions. First, it is possible to improve the guidance performance of the robotic guide by utilizing additional sensors. For example, the robotic guide in this study estimated its own location based only on its past action commands. If the robot is combined with improved localization systems, more accurate guidance can be achieved over longer distances than those in the present study. Second, future studies can investigate methods for providing personalized guidance according to different individual user characteristics. If the HPPN can predict an individual user's movement (e.g., the elderly or children) rather than general user movement by model adaptation (e.g., [4] and [6]), it is possible to optimize the robot's policy personalized for the user. Third, it is worthwhile to study various reward formulations that can improve the development of human-centered robots. Our focus in this study was on the smoothness of human movement, but there can be a variety of different metrics indicating human comfort (e.g., the amount of kinetic force applied to a user). Additionally, it is possible to train a generalized robot policy over various reward formulations using recent multi-objective RL [59], [60]. With the generalized robot policy, it is easy to investigate the reward formulation that users prefer, as there is no need to re-train the policy from scratch for each reward formulation. Finally, our sample-efficient robot-training method can be applied to other collaborative robots. It would be of great value to examine the viability of our approach in a variety of collaborative situations where robots need to interact with humans, such as in the case of industrial or surgical-assistive robots.

Recent growth in computing technology has made intelligent systems universal in our daily lives; it has become commonplace for robots or conversational agents to interact with people. In this regard, our research can contribute to the development of socially intelligent systems that can comprehend human behavior and discern user intentions.

REFERENCES

[1] S. Scheggi, M. Aggravi, and D. Prattichizzo, "Cooperative navigation for mixed human–robot teams using haptic feedback," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 4, pp. 462–473, Aug. 2017.

[2] C. Sirithunge, A. G. B. P. Jayasekara, and D. P. Chandima, "Proactive robots with the perception of nonverbal human behavior: A review," *IEEE Access*, vol. 7, pp. 77308–77327, 2019.

[3] Q. Li, Z. Zhang, Y. You, Y. Mu, and C. Feng, "Data driven models for human motion prediction in human–robot collaboration," *IEEE Access*, vol. 8, pp. 227690–227702, 2020.

[4] H.-S. Moon and J. Seo, "Optimal action-based or user prediction-based haptic guidance: Can you do even better?" in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2021, pp. 1–12.

[5] A. Antonucci, G. P. R. Papini, P. Bevilacqua, L. Palopoli, and D. Fontanelli, "Efficient prediction of human motion for real-time robotics applications with physics-inspired neural networks," *IEEE Access*, vol. 10, pp. 144–157, 2021.

[6] H.-S. Moon and J. Seo, "Fast user adaptation for human motion prediction in physical human–robot interaction," *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 120–127, Jan. 2022.

[7] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, no. 5, pp. 4282–4286, 1995.

[8] X. B. Peng, G. Berseth, and M. van de Panne, "Terrain-adaptive locomotion skills using deep reinforcement learning," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–12, Jul. 2016.

[9] R. Hafner, T. Hertweck, P. Kloeppner, M. Bloesch, M. Neunert, M. Wulfmeier, S. Tunyasuvunakool, N. Heess, and M. Riedmiller, "Towards general and autonomous learning of core skills: A case study in locomotion," in *Proc. Conf. Robot Learn.*, 2020, pp. 1084–1099. [Online]. Available: https://proceedings.mlr.press/v155/hafner21a.html

[10] M. Bloesch, J. Humplik, V. Patraucean, R. Hafner, T. Haarnoja, A. Byravan, N. Y. Siegel, S. Tunyasuvunakool, F. Casarini, N. Batchelor, F. Romano, S. Saliceti, M. Riedmiller, S. M. A. Eslami, and N. Heess, "Towards real robot learning in the wild: A case study in bipedal locomotion," in *Proc. Conf. Robot Learn.*, 2021, pp. 1502–1511.

[11] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 421–436, 2018.

[12] A. X. Lee *et al.*, "Beyond pick-and-place: Tackling robotic stacking of diverse shapes," in *Proc. Conf. Robot Learn.*, 2021, pp. 1089–1131.

[13] M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *Int. J. Robot. Res.*, vol. 39, no. 1, pp. 3–20, 2020.

[14] S. Shoval, J. Borenstein, and Y. Koren, "The NavBelt—A computerized travel aid for the blind based on mobile robotics technology," *IEEE Trans. Biomed. Eng.*, vol. 45, no. 11, pp. 1376–1386, Nov. 1998.

[15] I. Ulrich and J. Borenstein, "The GuideCane-applying mobile robot technologies to assist the visually impaired," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 31, no. 2, pp. 131–136, Mar. 2001.

[16] V. Kulyukin, C. Gharpure, J. Nicholson, and G. Osborne, "Robot-assisted wayfinding for the visually impaired in structured indoor environments," *Auton. Robot.*, vol. 21, no. 1, pp. 29–41, 2006.

[17] C.-L. Lu, Z.-Y. Liu, J.-T. Huang, C.-I. Huang, B.-H. Wang, Y. Chen, N.-H. Wu, H.-C. Wang, L. Giarré, and P.-Y. Kuo, "Assistive navigation using deep reinforcement learning guiding robot with UWB/voice beacons and semantic feedbacks for blind and visually impaired people," *Frontiers Robot. AI*, vol. 8, p. 176, Jun. 2021.

[18] A. Wachaja, P. Agarwal, M. Zink, M. R. Adame, K. Möller, and W. Burgard, "Navigating blind people with walking impairments using a smart Walker," *Auto. Robots*, vol. 41, no. 3, pp. 555–573, Mar. 2017.

[19] Z. Li and R. Hollis, "Toward a ballbot for physically leading people: A human-centered approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4827–4833.

[20] M. Poggi and S. Mattoccia, "A wearable mobility aid for the visually impaired based on embedded 3D vision and deep learning," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 208–213.

[21] H.-C. Wang, R. K. Katzschmann, S. Teng, B. Araki, L. Giarré, and D. Rus, "Enabling independent navigation for visually impaired people through a wearable vision-based feedback system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 6533–6540.

[22] S. Kayukawa, K. Higuchi, J. Guerreiro, S. Morishima, Y. Sato, K. Kitani, and C. Asakawa, "BBeep: A sonic collision avoidance system for blind travellers and nearby pedestrians," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2019, pp. 1–12.

[23] M. Bousbia-Salah, M. Bettayeb, and A. Larbi, "A navigation aid for blind people," *J. Intell. Robot. Syst.*, vol. 64, nos. 3–4, pp. 387–400, 2011.

[24] A. Ghosh, J. Penders, P. E. Jones, and H. Reed, "Experience of using a haptic interface to follow a robot without visual feedback," in *Proc. 23rd IEEE Int. Symp. Robot Hum. Interact. Commun.*, Aug. 2014, pp. 329–334.

[25] T.-K. Chuang, N.-C. Lin, J.-S. Chen, C.-H. Hung, Y.-W. Huang, C. Teng, H. Huang, L.-F. Yu, L. Giarré, and H.-C. Wang, "Deep trail-following robotic guide dog in pedestrian environments for people who are blind and visually impaired-learning from virtual and real worlds," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 5849–5855.

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[27] L. Niu, C. Qian, J.-R. Rizzo, T. Hudson, Z. Li, S. Enright, E. Sperling, K. Conti, E. Wong, and Y. Fang, "A wearable assistive technology for the visually impaired with door knob detection and real-time feedback for hand-to-handle manipulation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1500–1508.

[28] A. Abdolrahmani, W. Easley, M. Williams, S. Branham, and A. Hurst, "Embracing errors: Examining how context of use impacts blind individuals' acceptance of navigation aid errors," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2017, pp. 4158–4169.

[29] J. Guerreiro, E. Ohn-Bar, D. Ahmetovic, K. Kitani, and C. Asakawa, "How context and user behavior affect indoor navigation assistance for blind people," in *Proc. 15th Int. Web All Conf.*, Apr. 2018, pp. 1–4.

[30] H. Kacorri, E. Ohn-Bar, K. M. Kitani, and C. Asakawa, "Environmental factors in indoor navigation based on real-world trajectories of blind users," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2018, pp. 1–12.

[31] Q. Yan, J. Huang, C. Xiong, Z. Yang, and Z. Yang, "Data-driven human–robot coordination based walking state monitoring with cane-type robot," *IEEE Access*, vol. 6, pp. 8896–8908, 2018.

[32] H.-S. Moon and J. Seo, "Prediction of human trajectory following a haptic robotic guide using recurrent neural networks," in *Proc. IEEE World Haptics Conf. (WHC)*, Jul. 2019, pp. 157–162.

[33] R. Akabane and Y. Kato, "Pedestrian trajectory prediction based on transfer learning for human-following mobile robots," *IEEE Access*, vol. 9, pp. 126172–126185, 2021.

[34] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke, "Towards vision-based deep reinforcement learning for robotic motion control," 2015, *arXiv:1511.03791*.

[35] S. Bohez, T. Verbelen, E. De Coninck, B. Vankeirsbilck, P. Simoens, and B. Dhoedt, "Sensor fusion for robot control through deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2365–2370.

[36] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3389–3396.

[37] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8973–8979.

[38] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," in *Proc. Robot., Sci. Syst. XIV*, Jun. 2018, pp. 1–11. [Online]. Available: http://www.roboticsproceedings.org/rss14/p10.html

[39] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," 2017, *arXiv:1703.03864*.

[40] N. Hansen, "The CMA evolution strategy: A comparing review," in *Towards a New Evolutionary Computation*. Berlin, Germany: Springer-Verlag, 2006, pp. 75–102.

[41] N. Shafii, N. Lau, and L. P. Reis, "Learning to walk fast: Optimized hip height movement for simulated and real humanoid robots," *J. Intell. Robot. Syst.*, vol. 80, nos. 3–4, pp. 555–571, Dec. 2015.

[42] C. Gehring, S. Coros, M. Hutter, C. D. Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. Hoepflinger, and R. Siegwart, "Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot," *IEEE Robot. Autom. Mag.*, vol. 23, no. 1, pp. 34–43, Mar. 2016.

[43] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 31–36.

[44] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 23–30.

[45] A. H. Qureshi, Y. Nakamura, Y. Yoshikawa, and H. Ishiguro, "Robot gains social intelligence through multimodal deep reinforcement learning," in *Proc. IEEE-RAS 16th Int. Conf. Hum. Robots (Humanoids)*, Nov. 2016, pp. 745–751.

[46] A. Shafti, J. Tjomsland, W. Dudley, and A. A. Faisal, "Real-world human–robot collaborative reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 11161–11166.

[47] A. Ghadirzadeh, J. Bütepage, A. Maki, D. Kragic, and M. Björkman, "A sensorimotor reinforcement learning framework for physical human–robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 2682–2688.

[48] S. Lathuilière, B. Massé, P. Mesejo, and R. Horaud, "Neural network based reinforcement learning for audio–visual gaze control in human–robot interaction," *Pattern Recognit. Lett.*, vol. 118, pp. 61–71, Feb. 2019.

[49] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[50] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 2455–2467.

[51] M. Thabet, M. Patacchiola, and A. Cangelosi, "Sample-efficient deep reinforcement learning with imaginary rollouts for human–robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 5079–5085.

[52] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[53] H.-S. Moon and J. Seo, "Observation of human response to a robotic guide using a variational autoencoder," in *Proc. 3rd IEEE Int. Conf. Robot. Comput. (IRC)*, Feb. 2019, pp. 258–261.

[54] S. Balasubramanian, A. Melendez-Calderon, and E. Burdet, "A robust and sensitive metric for quantifying movement smoothness," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 8, pp. 2126–2136, Aug. 2012.

[55] A. Nguyen, D. Kanoulas, L. Muratore, D. G. Caldwell, and N. G. Tsagarakis, "Translating videos to commands for robotic manipulation with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3782–3788.

[56] S. Balasubramanian, A. Melendez-Calderon, A. Roby-Brami, and E. Burdet, "On the analysis of movement smoothness," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, pp. 1–11, 2015.

[57] C. M. Harris and D. M. Wolpert, "Signal-dependent noise determines motor planning," *Nature*, vol. 394, no. 6695, pp. 780–784, Aug. 1998.

[58] D. A. Winter, "Human balance and posture control during standing and walking," *Gait Posture*, vol. 3, no. 4, pp. 193–214, 1995.

[59] A. Abels, D. Roijers, T. Lenaerts, A. Nowé, and D. Steckelmacher, "Dynamic weights in multi-objective deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 11–20.

[60] R. Yang, X. Sun, and K. Narasimhan, "A generalized algorithm for multi-objective reinforcement learning and policy adaptation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 14636–14647.

HEE-SEUNG MOON received the B.S. degree in integrated technology from Yonsei University, Incheon, South Korea, in 2015. He is currently pursuing the Ph.D. degree with the School of Integrated Technology. His research interests include human–computer interaction, computational interaction, user behavior modeling, and deep learning. He received the Undergraduate and Graduate Fellowships from the Information and Communications Technology (ICT) Consilience Creative Program supported by the Ministry of Science and ICT, South Korea.

JIWON SEO (Member, IEEE) received the B.S. degree in mechanical engineering from the Division of Aerospace Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2002, the M.S. degree in aeronautics and astronautics, in 2004, the M.S. degree in electrical engineering, in 2008, and the Ph.D. degree in aeronautics and astronautics from Stanford University, Stanford, CA, USA, in 2010. He is currently an Associate Professor with the School of Integrated Technology, Yonsei University, Incheon, South Korea. His research interests include GNSS anti-jamming technologies, complementary PNT systems, and intelligent unmanned systems. He is a member of the International Advisory Council of the Resilient Navigation and Timing Foundation, Alexandria, VA, USA, and a member of several Advisory Committees of the Ministry of Oceans and Fisheries and the Ministry of Land, Infrastructure and Transport, South Korea.