TeachMe: Three-phase learning framework for robotic motion imitation based on interactive teaching and reinforcement learning

Taewoo Kim[1] and Joo-Haeng Lee[2]

***Abstract*— Motion imitation is a fundamental communication skill for a robot; especially, as a nonverbal interaction with a human. Owing to kinematic configuration differences between the human and the robot, it is challenging to determine the appropriate mapping between the two pose domains. Moreover, technical limitations while extracting 3D motion details, such as wrist joint movements from human motion videos, results in significant challenges in motion retargeting. Explicit mapping over different motion domains indicates a considerably inefficient solution. To solve these problems, we propose a three-phase reinforcement learning scheme to enable a NAO robot to learn motions from human pose skeletons extracted from video inputs. Our learning scheme consists of three phases: (i) phase one for learning preparation, (ii) phase two for a simulation-based reinforcement learning, and (iii) phase three for a human-in-the-loop-based reinforcement learning. In phase one, embeddings of the motions of a human skeleton and robot are learned by an autoencoder. In phase two, the NAO robot learns a rough imitation skill using reinforcement learning that translates the learned embeddings. In the last phase, the robot learns motion details that were not considered in the previous phases by interactively setting rewards based on direct teaching instead of the method used in the previous phase. Especially, it is to be noted that a relatively smaller number of interactive inputs are required for motion details in phase three when compared to the large volume of training sets required for overall imitation in phase two. The experimental results demonstrate that the proposed method improves the imitation skills efficiently for hand waving and saluting motions obtained from NTU-DB.**

I. INTRODUCTION

According to social learning theory [1][2][3], humans can learn new behaviors by observing and imitating others. Through such social learning skills, humans interact and perform nonverbal social actions like hand waving or bowing. Similarly, for human-robot social nonverbal interactions, it is essential to teach a robot to observe and imitate human motions.

Imitation learning is a methodology that has long been studied to teach human motions to a robot [4][5][6][7][8]. Various robot teaching methodologies exist including point-based direct teaching [9][10][11][12], motion retargeting using dynamic modeling and optimization from human motion data [13], indirect teaching by teleoperation [14], and teaching using virtual reality-based teleoperation [15]. With these accurate dynamic modeling and end-effector-based teaching methods, a robot can perform the target task effectively.

[1]Taewoo Kim is with the Department of Computer Software and Engineering, Korea University of Science and Technology, Daejeon, Republic of Korea twkim0812@gmail.com

[2]Joo-Haeng Lee with the Human-Robot Interaction Research Group, Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea joohaeng@etri.re.kr

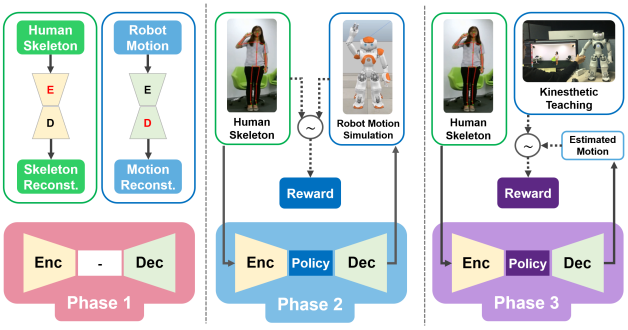

Fig. 1: Three-phase reinforcement learning framework. Human skeleton encoder and NAO motion decoder are constructed in phase one. In phase 2, simulation-based policy learning is processed and the policy for a robotic motion imitation is trained interactively in phase three.

While considering human-robot social interactions, for imitating the human motion without losing the implicit meaning or intention of a performer, accurate motion retargeting should be performed not only in the end effector but also in the other joints like the elbow. In particular, precise devices or methods are required to overcome the problems that occur owing to kinematic configuration differences while teaching elaborate motions like wrist-joint movements. Thus, in these cases, the kinesthetic teaching that guides the robot by directly manipulating its arm without any external devices is intuitive and effective especially in teaching wrist joint movement [16][17].

Recently, many studies introduce deep learning technology to solve challenging problems. A lot of training data are required for the deep learning. However collecting such a large amount of data costs a lot especially in robotic applications. In this study, to take advantages of both the direct teaching with a small set of trials and the deep learning, we attempted to perform fine tuning of the policy which was trained by a simulator with deep reinforcement learning in the previous step. We call our approach a three-phase reinforcement learning including simulation-based learning and human-in-the-loop-based[14] interactive learning methods (Figure 1). Our goal is to enable a NAO robot to generate a motion that matches the poses of a human skeleton that demonstrate hand waving and saluting motions, which are acquired from the NTU-DB [18]. Our learning scheme consists of three phases. Phase one, as a preparation step, generates embeddings of the human and NAO motions using skeleton and synthetic motion data. In phase 2, we determine the mapping policy between the two embeddings using reinforcement learning.

In the last phase, kinesthetic teaching is performed frame-by-frame on the optimal policy learned in the phase 2. We were able to learn the detailed motion, which was not considered in the previous phase, such as the wrist joint movement, and resolve the difficulties caused by kinematic configuration differences. Further, through experiments, we demonstrated that the imitation skill could be improved in relatively less learning time. Table 1. lists the usage of simulation and interactive learning approaches for each learning phase.

TABLE I: Characteristics of each phases: learning target, the usage of simulation and human guidance for each learning phase. See Fig. 3 for more tails

Phase	Learning Target	Simulation Usage	Human Guidance
I	Encoder ρ_s Decoder ψ_r	synthetic motion generation for ψ_r	Indirect (synthetic motion design)
II	RL policy π_ξ	robot motion simulation for π_ξ	Indirect (reward function design)
III	RL policy π_ϕ	-	Direct (interactive teaching)

In summary, our main contributions are as follows:

- We modeled the human motion mimicry as a mapping problem between the human skeleton and the robot motion, proposed a user-guidance based three-phase framework and showed the experimental results. To our knowledge, this is the first framework for human motion mimicry, which combines learning from synthetic motion with interactive fine tuning.
- We generated a unified motion decoder for robots which embraces all of our motion classes and verified that our policy can successfully be trained with this integrated robot motion decoder.
- We experimentally showed that the quantitatively trained policy from the previous phase can be effectively improved by a small set of interactive teachings.

II. RELATED WORK

In this section, we will review the relevant studies and discuss the differences with respect to our problem.

A. Encoder-Decoder based Architecture

The encoder-decoder network architecture has been gaining popularity in several recent studies including robotic tasks [19][20][21], human motion prediction [22][23], machine translation [24], and image captioning [25]. While considering robotic tasks, numerous recent studies solved the assigned specific problems such as throwing a ball to hit the target object [19] and robotic pick and place operation and assembly [15], and they usually utilized a spatial autoencoder [20], which encodes spatial information of the target object from the input image. Owing to recent advancements in human pose estimation, several skeleton-based approaches were adopted in human motion prediction. *Liangyan et al.* [22] attempted to regenerate the human motion from

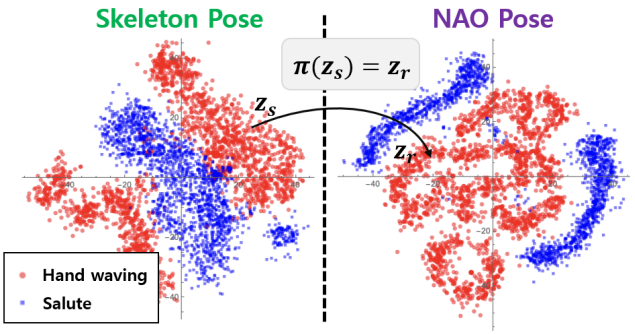

Fig. 2: Low-dimensional state representations of the poses of the skeleton and NAO. Red dot represents the hand-waving action and blue dot shows the saluting action

skeleton-based seed motion on the pepper robot using a generative adversarial network [26] and evaluated their method using the Human3.6M dataset [25]. *Brian et al.* proposed a transfer learning method for a robot manipulation skills using shared gaussian process latent variable model (GP-LVM) which enables multiple observational spaces share a common latent space [27]. Through this method, they were able to transfer a knowledge to other robots having similar kinematic configurations by reusing the latent representation as a prior information. There is another encoder-decoder method that is called a manifold alignment [28]. To learn human motion skills, human demonstration and robot motion are clustered by gaussian mixture model (GMM) and K-means algorithm, then their alignment (or mapping) is performed by local procrustes analysis (LPA) [29]. In our study, we attempted the real-time imitation of human motion by the NAO robot using frame-by-frame skeleton data and robotic motion encoded by a variational autoencoder (VAE) [30]. The motion retargeting between the poses of the skeleton and the robot was achieved by reinforcement learning; moreover, the NTU-DB [18] was used for training and evaluation.

B. Reinforcement Learning

From games [31][32][33] and robotics [34][35] to animation [36], reinforcement learning has gained popularity in various research fields in recent years. *Ghadirzadeh et al.* [19] performed the task of throwing a ball and grasping it by using a Personal Robot 2 robot with VAE and reinforcement learning. To achieve the goal task, they used a reinforcement learning-based approach to create a mapping between the low-dimensional state representations of the input image and the robotic motion trajectory that was generated from a synthetic motion generator. However, because the method only considers the success or failure of the goal task and does not consider detailed motion, this approach demonstrates limitations while applying to a robotic imitation for social interaction. Thus, based on these VAE and reinforcement learning-based methods, we added a kinesthetic teaching-based learning procedure. Further, instead of the trajectory, our framework runs on frame-by-frame data using a proximal policy optimization (PPO) algorithm [37], which is a recent reinforcement learning algorithm.

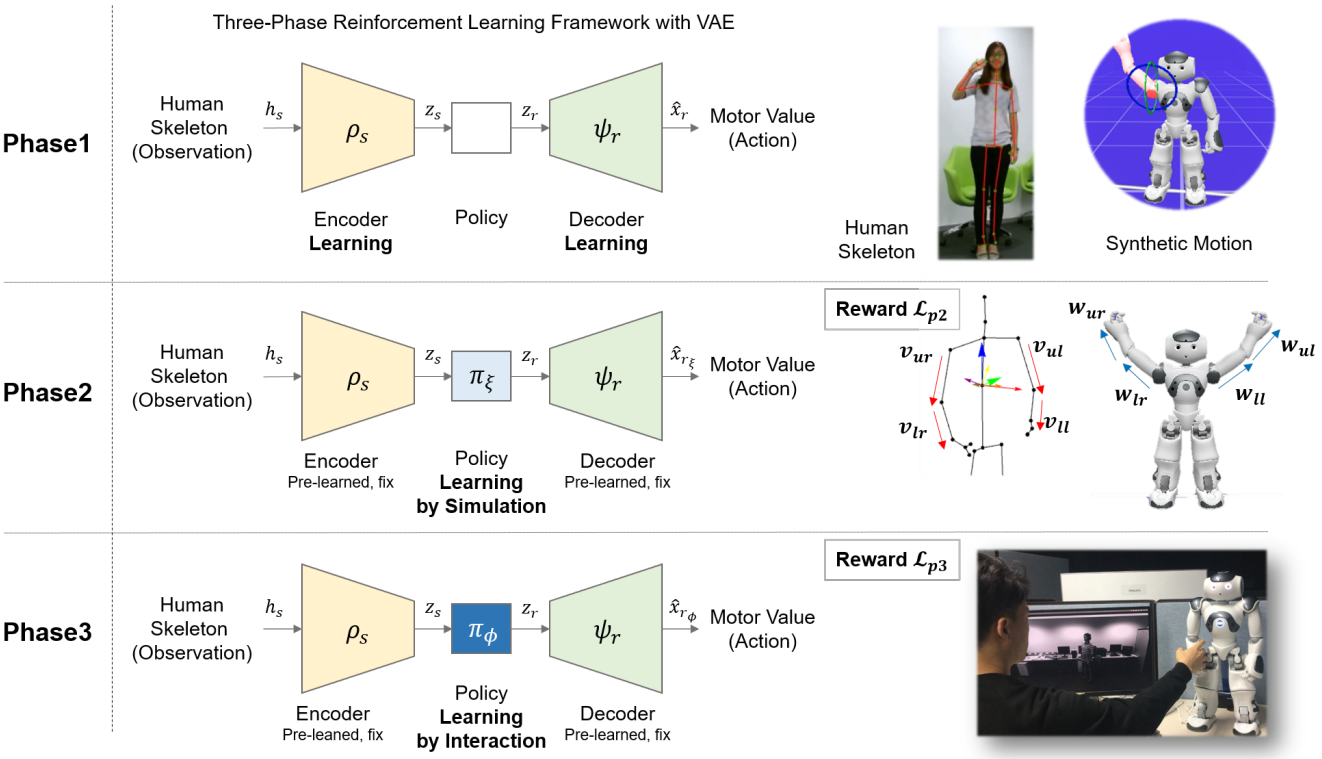

Fig. 3: Three-phase reinforcement learning framework. In phase one, skeleton and NAO motions encoded by variational autoencoders (VAEs) are learned from NTU-DB skeletons and synthetic reference motions generated by a simulator. In phase 2, the mapping policy between the low-dimensional representation of the motions of the skeleton and robot are learned by reinforcement learning. In the last phase, based on the learned policy in the previous phase, the policy is learned again by kinesthetic teaching to correct the errors and learn the motion details.

III. APPROACH

The primary concept of motion imitation learning is to determine the mapping function π between the low-dimensional state representations of the poses of the skeleton and the NAO. As shown in Figure 2, motion embeddings of the two poses have different distributions; thus, we attempted to determine the mapping for motion retargeting between these poses using reinforcement learning. To achieve this, we initially described the mathematical definition of our problem and illustrated the data preprocessing step.

A. Problem Formulation

From a given human pose image at time t, D_t, and skeleton transformation function, f_s, we defined a skeleton transformation from the image of the human pose as $x_s = f_s(D_t)$. Here, ρ_s denotes the skeleton encoder, $x_s = \{x_1, y_1, z_1, ..., x_{25}, y_{25}, z_{25}\}$ denotes the skeleton itself and $z_s = \rho_s(x_s)$ is the encoding of a skeleton, where z_s is a low-dimensional representation of the skeleton corresponding to the image of the human pose D_t. Similarly, $x_r = \{\theta_1, \theta_2, ..., \theta_{10}\}$ is a pose of the NAO and ρ_r is a robot pose encoder, the low-dimensional state representation of the pose of the NAO is described by $z_r = \rho_r(x_r)$. We aimed to determine the appropriate policy $z_r = \pi(z_s)$, which mapped the pose from the given latent representation of the pose of a human, z_s, to the latent representation, z_r, corresponding to the pose of a NAO. Figure 2 shows the result of plotting the low-dimensional representation of hand waving and saluting motions using a t-distributed stochastic neighbor embedding [38]. The left and right sides of Figure 2 indicate the skeleton and NAO, respectively.

B. NTU-DB

The skeleton provided by NTU-DB consisted of 25 joints, each of which had x, y, z coordinates. Because we defined a reward function with respect to the kinematic configurations of both the skeleton and NAO in phase-two, a preprocessing step was required to calibrate the coordinates of the skeleton to fit it to the coordinates of the NAO. Initially, every joint of the skeleton was moved based on the torso joint, and a vector was created between the torso and the pelvis joint. From the vector, orthogonal vectors were created and then a direction cosine matrix (DCM) was generated with respect to the Kinect reference coordinate as the skeleton was represented on the basis of a Kinect coordinate. After the coordinate rotation using the DCM, it was again rotated to the NAO coordinate. Finally, we finished coordinate transformation with a yaw-direction correction.

C. Phase 1: Encoder and Decoder

As shown in Figure 3, the VAE [30] is used for learning low-dimensional state representations of the skeleton and the

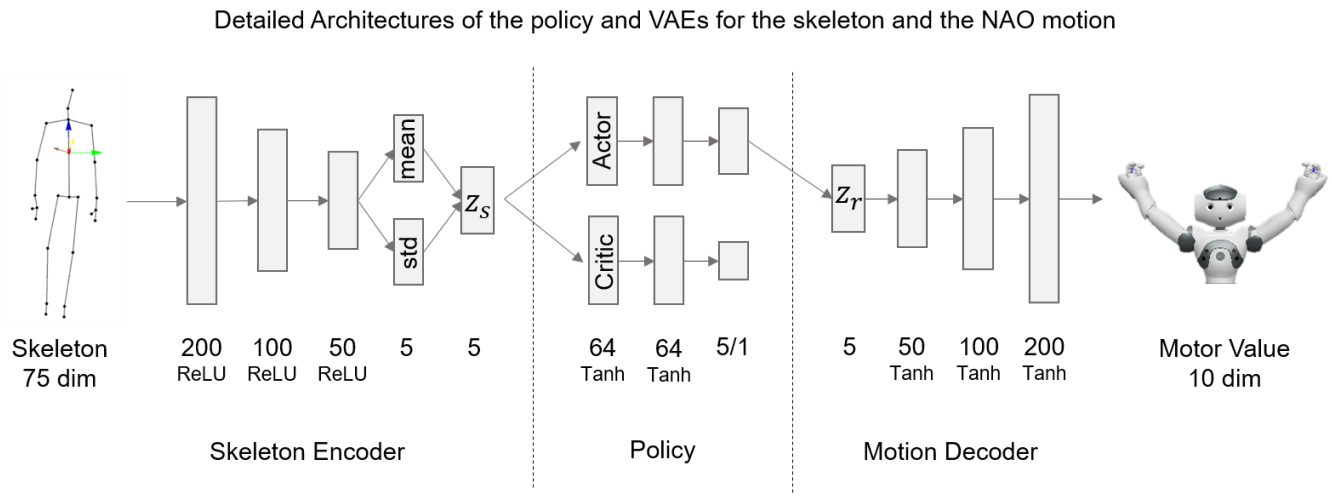

Fig. 4: Detailed architectures of the mapping policy and the VAEs for the skeleton and NAO motions.

reference motion of the NAO. With the exclusion of error data, 40k frames for hand waving action and 32k frames for saluting action were used to train the VAE of the skeleton. A detailed structure for the encoder part of the VAE of the skeleton is shown on the left side of Figure 4. The skeleton encoder ρ_s accepts a skeleton of 75-dimensional (25×3) x_s as an input and after sampling the mean and standard deviation through 200, 100, and 50 layers with a rectified linear unit activation function, outputs a five-dimensional latent representation vector z_s.

The VAE for the NAO motion has a similar structure to that of the skeleton and only a decoder ψ_r was used in our study to generate an estimated motor value $\psi_r(z_r) = \hat{x}_r$ from a latent representation vector z_r of the pose of the NAO. A detailed structure of the NAO motion decoder is shown on the right side of Figure 4. Using the robotic motion latent representation vector z_r, the estimated motor values $\hat{x}_r = \{\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_{10}\}$ through the 50, 100, and 200 layers with Tanh activation function are outputted. The motor values, $\hat{\theta}_1, \cdots, \hat{\theta}_{10}$, correspond to the five joints of the left arm, i.e., shoulder pitch, roll, elbow yaw, roll, wrist yaw, and the similar five motions of the right arm. To train the NAO motion decoder ψ_r for each action class, an official NAO simulator called "Choregraphe" [39] was used to generate several synthetic reference motion frames for each action. In the hand waving motion, from the four reference motion patterns on both arms, we created 20k frames by noise addition data augmentation. In the saluting motion, two reference motion patterns and the noise addition data augmentation were used to generate 10k frames.

D. Phase 2: Simulation-based Reinforcement Learning

In phase 2, we trained the policy π_ξ, which can appropriately map between the skeleton and NAO motion embeddings using the skeleton encoder ρ_s and the NAO motion decoder ψ_r that were learned in phase one. The policy was trained by PPO [37], which is a recent reinforcement learning algorithm. As shown in the central portion of Figure 4, the policy consists of actor and critic networks, each of which has two fully connected layers of 64 layers with a Tanh activation function. An encoded skeleton z_s was fed to the actor and critic networks, which yielded an action by the actor that was used as an input to the NAO motion decoder, and a value by the critic for learning the actor network by evaluation. A V-REP [40] simulator was used for the NAO motion simulation during training.

To train a policy for a proper mapping, an appropriate reward function design was required. Because all the motions chosen for our experiments used only the arms, we designed the reward function using the vectors of both arms to ensure that the NAO follows the pose of the skeleton. The subsequent equations represent the reward function used in reinforcement learning in phase 2.

$$\delta_i = \cos\left(\frac{v_i \cdot w_i}{||v_i|| ||w_i||}\right)^{-1}, \; i \in S = \{ur, lr, ul, ll\} \tag{1}$$

$$\mathcal{L}_{p2} = \sum_{i \in S} \frac{\exp(-2.0 \cdot \delta_i)}{4} \tag{2}$$

A cosine similarity, δ_i, between the skeleton and NAO arm vectors is calculated by equation (1). In the second row of Figure 3, v_i and w_i indicate the skeleton arm vectors, and the NAO arm vectors respectively, where the index i means the upper right(ur), lower right(lr), upper left(ul), and lower left(ll) parts of the arm in S. Each arm vector was generated by calculating the position differences between shoulder, elbow and wrist joint of the skeleton and the NAO respectively. Then, the delta, δ_i, was used to calculate the reward function, $\mathcal{L}_{p2}$, which was normalized between 0 and 1. To amplify the similarity error, -2.0 was multiplied to δ_i. The objective function in phase 2 was formulated as:

$$\xi^* = \underset{\xi}{\operatorname{argmax}} \; \mathbb{E}[\mathcal{L}_{p2}(x_s, \hat{x}_{r_\xi})] \tag{3}$$

$$\text{where } \hat{x}_{r_\xi} = \psi_r \circ \pi_\xi \circ \rho_s \circ f_s(D_t)$$

where the goal of phase 2 was to find the optimal parameter ξ^* for the policy π_ξ, which maximizes the expected rewards.

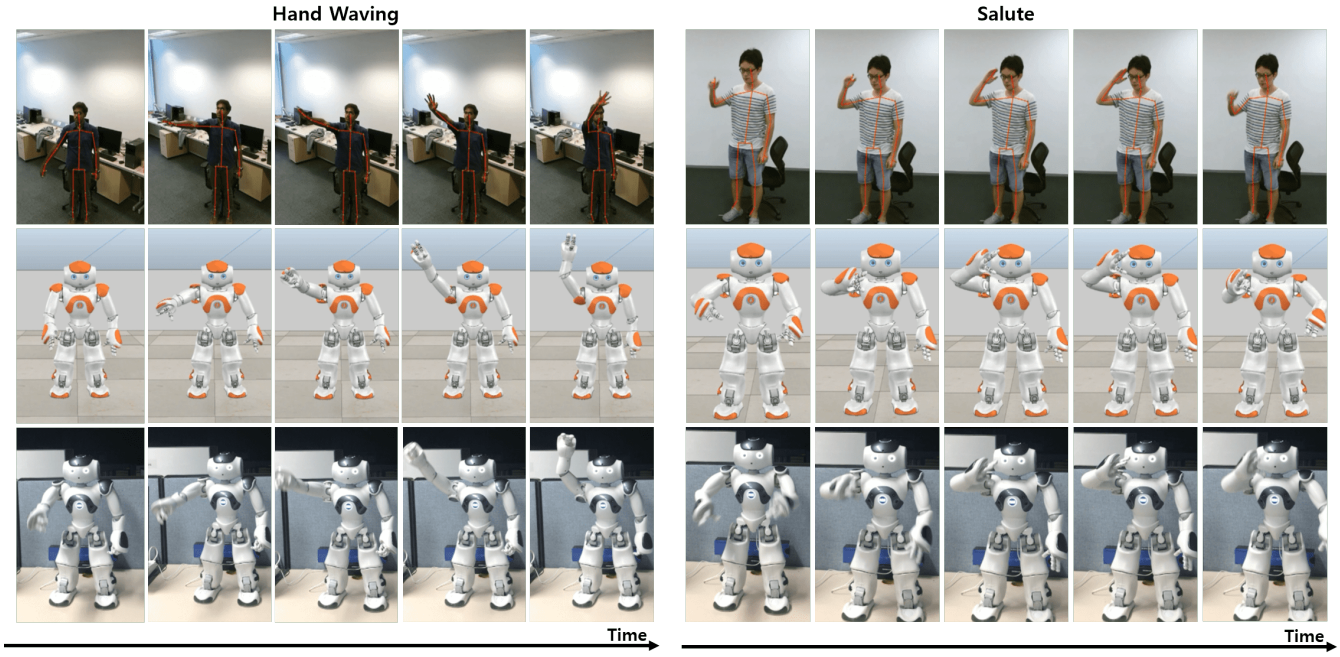

Fig. 5: Experimental results demonstrating robotic motion imitation. The flow of time is from left to right in the series of images. The first, second, and third rows show the input video and skeleton, NAO in simulator, and NAO in real world, respectively.

The estimated NAO motor value $\hat{x}_{r_\xi}$ that corresponds to the input skeleton x_s was generated by the skeleton encoder ρ_s, the policy of π_ξ, and the NAO motion decoder ψ_r.

E. Phase 3: Interactive Reinforcement Learning

In phase 2, we trained our policy for motion imitation with a simulator and an appropriately designed reward function. However, the NAO was not fully capable of representing six degrees of freedom (DOF) with each arm in the Cartesian space owing to the insufficient DOF (five DOF per arm). In addition to this, there were several limitations that resulted in difficulty in the imitation learning. Because the NAO arm has a lower DOF than the skeleton, it has a different manipulability and the detailed motions such as wrist joint movement was not considered in the reward function $\mathcal{L}_{p2}$ of phase 2. Even the wrist joint movement of a skeleton created by f_s demonstrated low reliability owing to its noise. Because of the kinematic configuration differences and difficulties for detailed motion learning, phase 2 approach demonstrated obvious limitations. To overcome the limitations, we imposed the human-in-the-loop teaching method in phase three, which interacted with the robot to teach a motion directly during the training time. In this approach, a human observed each input skeleton frame and generated a NAO pose corresponding to the skeleton by manipulating its arm and recording this to memory. When a certain number of batch sized training data were collected, then the policy was updated by the reward function. The following equations represent the reward function in phase three.

$$e = ||\psi_r(\pi_\phi(z_s)) - x_{gt}||_2 \tag{4}$$

$$\mathcal{L}_{p3} = \exp(-2.0 \cdot e) \tag{5}$$

where the error amplification constant -2.0 was set empirically. An input latent representation of a skeleton, z_s, was mapped to the latent representation of the NAO motion, z_r, by the policy of phase three $z_r = \pi_\phi(z_s)$. The z_r was decoded again by the NAO motion decoder to generate the estimated motor value $\hat{x}_r = \psi_r(z_r)$. Now, from equation (4), ℓ_2-norm error was calculated between the estimated motor value, $\hat{x}_r$, and the ground truth motor value, $x_{gt} = \{\theta_1, \theta_2, \cdots, \theta_{10}\}$, which was created by a human. Using this error, the normalized reward $\mathcal{L}_{p3}$ in phase 3 is then calculated by exponential function and an amplification constant as equation (4). The objective function formulation in phase three is as follows.

$$\phi^* = \underset{\phi}{\text{argmax}} \ \mathbb{E}[\mathcal{L}_{p3}(x_{gt}, \hat{x}_{r_\phi})] \tag{6}$$

$$\text{where } \phi = \xi^* \text{ and } \hat{x}_{r_\phi} = \psi_r \circ \pi_\phi \circ \rho_s \circ f_s(D_t)$$

The optimal policy parameter of phase 2, ξ^*, becames the initial policy parameter of phase three, $\phi = \xi^*$. Now, the objective in phase three was to determine the optimal policy parameter, ϕ^*, which maximized the expected reward function $\mathcal{L}_{p3}$. Consequently, we observed that a coarse tuning was performed in phase 2, and fine tuning was performed in phase three.

IV. Experiments

The target motion classes used in our experiments were hand waving and saluting motions from the NTU-DB [18]. We generated an extra program to filter out the corrupted skeleton data and collected only normal data for each motion class. Using these normal data of the skeletons, the skeleton encoder was trained, and synthetic motion data generated

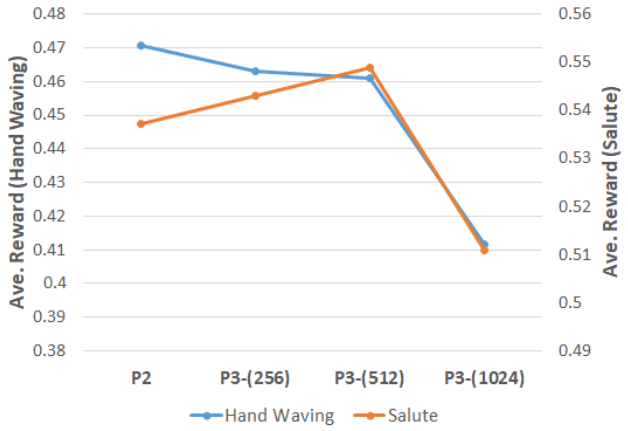

Fig. 6: Evaluation results during 5k frames for each motion class. The last three points of the horizontal axis is the number of human demonstration frames used to train in phase 3 and the vertical axis is the average reward for 5k frames in each motion class and in each trained policy

from the Choregraphe [39] was used to train the NAO motion decoder. For the simulation-based learning in phase 2, we utilized the V-REP [40] simulator. In this section, we describe the detailed parameters used in each learning phase and analyze the experimental results of the simulation and actual NAO robot.

A. Learning Details

In the VAE learning for the skeleton and robot motion encoder/decoder, 128 batch size and ℓ_2-norm loss function were used identically. In phase 2, 500k frames were sampled from a simulator and used to train the policy π_ξ for each motion class with a batch size of 2048, entropy coefficient of 0.005, and PPO epoch of 10. The time required in learning from the 500k frames of data was approximately 5 h using a single simulator. We collected 1024 ground truth data per class in less than 20 min. The policy π_ϕ was updated online with a batch size of 64. The other parameters were similar to phase 2. In phase 2 learning, we set the learning rate to 1e-4 and weight decay to 1e-5. In phase 3, the learning rate was to 1e-3 with PPO epoch of 2. Our framework was implemented by PyTorch [41].

B. Results

We evaluated and compared the performance of the simulation-based method in phase 2 and the interactive method in phase three. The evaluations were performed using the reward function $\mathcal{L}_{p2}$ of phase 2 for the 5k frames in each motion class. As shown in Figure 6, the average reward shows a trend toward a decrease as the number of teaching data increases in phase 3. Although it appears as if the teaching in phase 3 bring performance degradation in robotic motion mimicry, the robot can generate a more human-like motion as shown in Figure 7. The generated motions by the phase 2 policy show a tendency to follow the human skeleton posture as possible. However, they seem somewhat awkward not only in the overall robot posture but also detailed motion

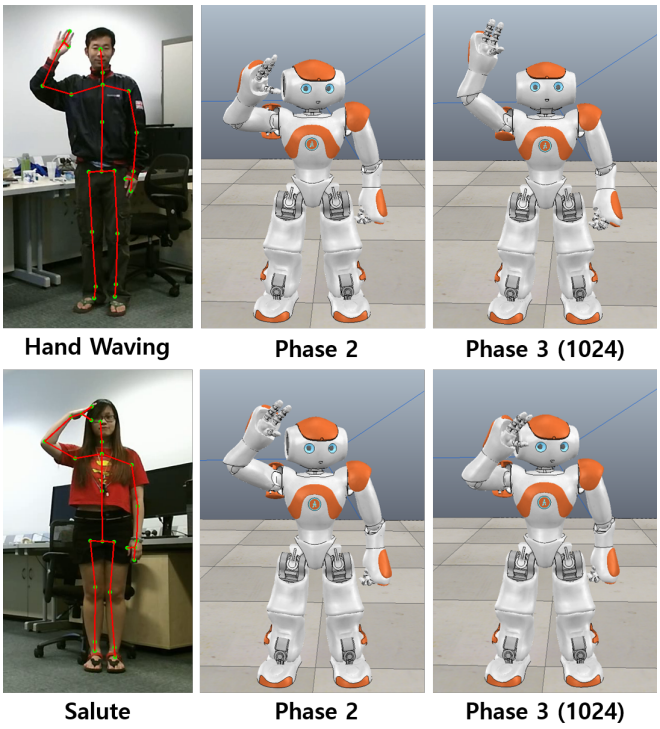

Fig. 7: Evaluation results of the phase 2 and the phase 3 policy. The first and the second row is the results of the hand waving and saluting motion respectively. The second column represent a robot posture generated by the policy of phase 2 and the third is the result of the policy of phase 3 with 1024 human-guided teaching frames.

such as wrist joint. On the other hand, the motions generated by the phase 3 policy seem more natural than policy in phase 2. In particular, we can see that the NAO robot rotates his wrist joint appropriately for each motion class in phase 3. For example, the NAO robot rotated his wrist joint of the right hand to the proximal direction when he is saluting. Neither those motion details can be acquired from the human skeleton nor trained in phase 2. Therefore, the phase 3 which enables training a robot motion details with a small set of teaching data is important process in our framework.

C. Discussion

In our study, we attempted to solve the robotic motion imitation problem by our three-phase framework including human-in-the-loop-based interactive teaching. Although the interactive approach can be applied from the initial learning step, it is impractical considering the data collection time. This is because, 0.04 s was required per sampling in phase 2, while 1.2 s was taken per sampling in phase three. Therefore, we would require approximately 166 h to collect 500k training samples using the interactive teaching method. We have achieved those quantitative learning by simulator and qualitative learning by small set of direct teaching. Therefore, our three-phase approach that divides the learning process into preparation, coarse tuning, and fine tuning is effective and efficient in terms of the effort, cost, and total learning

time. Figure 5 shows the robotic motion imitation results of the NAO in the simulation (second row) and of the actual NAO (third row) with final policy. Through our three-phase framework, we were effectively able to train the NAO robot the human motion imitation skills including motion details with a small set of direct teaching. However, there was a tradeoff between skeleton following dexterity and motion detail. As the teaching in phase 3 progressed, the NAO gained motion details while losing its skeleton following dexterity. In our future work, we will study about optimal reward function design of phase 3 to maintain the skeleton following dexterity while learning motion details.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel learning-based approach to enable a NAO robot to imitate motions of a human skeleton. Typically, the data requirements for a learning-based method are significantly high both in terms of quality and quantity. However, securing such abundant high-quality training data is considerably challenging in terms of time, effort, and cost. To balance between the data quality and quantity simultaneously, we introduced a three-phase reinforcement learning method: phase one and two for learning the quantitative data from a synthetic motion generator, and phase three for learning the qualitative data from interactive human inputs. Through the experiments using reference human motions from NTU-DB, we verified the effectiveness and efficiency of the proposed method.

It is to be noted that our method was based on individual motion frames rather than a consecutive sequence, which intrinsically limits its applicability for multi-class motions as well as further performance gain. Hence, our immediate future research focuses on a trajectory-based approach. Moreover, we also believe that virtual-reality-based inputs could be a good candidate for interactive inputs.

ACKNOWLEDGMENT

This work was supported by UST Young Scientist Research Program through the University of Science and Technology. (No. [18AS1810])

This work was supported by the ICT R&D program of MSIP/IITP. [2017-0-00162, Development of Human-care Robot Technology for Aging Society.

REFERENCES

[1] A. Bandura and R. H. Walters, *Social learning theory*, vol. 1. Prentice-hall Englewood Cliffs, NJ, 1977.

[2] A. Bandura and R. H. Walters, "Social learning and personality development.," 1963.

[3] C. M. Renzetti, D. J. Curran, and S. L. Maier, "Women, men, and society," 2012.

[4] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[5] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 358, no. 1431, pp. 537–547, 2003.

[6] D. Kulić, "Human motion imitation," *Humanoid Robotics: A Reference*, pp. 1–21, 2018.

[7] R. Elbasiony and W. Gomaa, "Humanoids skill learning based on real-time human motion imitation using kinect," *Intelligent Service Robotics*, vol. 11, no. 2, pp. 149–169, 2018.

[8] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[9] G. Grunwald, G. Schreiber, A. Albu-Schaffer, and G. Hirzinger, "Programming by touch: The different way of human-robot interaction," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 659–666, 2003.

[10] D. Kushida, M. Nakamura, S. Goto, and N. Kyura, "Human direct teaching of industrial articulated robot arms based on force-free control," *Artificial Life and Robotics*, vol. 5, no. 1, pp. 26–32, 2001.

[11] T. Tsumugiwa, R. Yokogawa, and K. Hara, "Variable impedance control based on estimation of human arm stiffness for human-robot cooperative calligraphic task," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1, pp. 644–650, IEEE, 2002.

[12] R. D. Schraft, C. Meyer, C. Parlitz, and E. Helms, "Powermate-a safe and intuitive robot assistant for handling and assembly tasks," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 4074–4079, IEEE, 2005.

[13] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin, "On human motion imitation by humanoid robot," in *2008 IEEE International Conference on Robotics and Automation*, pp. 2697–2704, IEEE, 2008.

[14] L. Peternel, T. Petrič, and J. Babič, "Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface," in *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 1497–1502, IEEE, 2015.

[15] T. Zhang, Z. McCarthy, O. Jowl, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, IEEE, 2018.

[16] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz, "Keyframe-based learning from demonstration," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 343–355, 2012.

[17] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning from physical human corrections, one feature at a time," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 141–149, ACM, 2018.

[18] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1010–1019, 2016.

[19] A. Ghadirzadeh, A. Maki, D. Kragic, and M. Björkman, "Deep predictive policy training using reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2351–2358, IEEE, 2017.

[20] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 512–519, IEEE, 2016.

[21] A. Gams, A. Ude, J. Morimoto, *et al.*, "Deep encoder-decoder networks for mapping raw images to dynamic movement primitives," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–6, IEEE, 2018.

[22] L.-Y. Gui, K. Zhang, Y.-X. Wang, X. Liang, J. M. Moura, and M. Veloso, "Teaching robots to predict human motion," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 562–567, IEEE, 2018.

[23] L.-Y. Gui, Y.-X. Wang, D. Ramanan, and J. M. Moura, "Few-shot human motion prediction via meta-learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 432–450, 2018.

[24] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, pp. 3294–3302, 2015.

[25] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[27] B. Delhaisse, D. Esteban, L. Rozo, and D. Caldwell, "Transfer learning of shared latent spaces between robots with similar kinematic structure," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 4142–4149, IEEE, 2017.

[28] N. Makondo, M. Hiratsuka, B. Rosman, and O. Hasegawa, "A non-linear manifold alignment approach to robot learning from demonstrations," *Journal of Robotics and Mechatronics*, vol. 30, no. 2, pp. 265–281, 2018.

[29] N. Makondo, B. Rosman, and O. Hasegawa, "Knowledge transfer for learning robot models via local procrustes analysis," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 1075–1082, IEEE, 2015.

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

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[32] G. Lample and D. S. Chaplot, "Playing fps games with deep reinforcement learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[33] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[34] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, "Imitation from observation: Learning to imitate behaviors from raw video via context translation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1118–1125, IEEE, 2018.

[35] M. Jaritz, R. De Charette, M. Toromanoff, E. Perot, and F. Nashashibi, "End-to-end race driving with deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2070–2075, IEEE, 2018.

[36] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 143, 2018.

[37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[38] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[39] E. Pot, J. Monceaux, R. Gelin, and B. Maisonnier, "Choregraphe: a graphical tool for humanoid robot programming," in *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 46–51, IEEE, 2009.

[40] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1321–1326, IEEE, 2013.

[41] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.