Conditional Generative Adversarial Network for Generating Communicative Robot Gestures

Nguyen Tan Viet Tuyen, Armagan Elibol, and Nak Young Chong

***Abstract*— Non-verbal behaviors have an indispensable role for social robots, which help them to interact with humans in a facile and transparent way. Especially, communicative gestures allow robots to have the capability of using bodily expressions for emphasizing the meaning of their speech, describing something, or showing clear intention. This paper presents an approach to learn the synthesis of human actions and natural language. The generative framework is inspired by Conditional Generative Adversarial Network (CGAN), and it makes use of the Convolutional Neural Network (CNN) with the Action Encoder/Decoder for action representation. The experimental and comparative results verified the efficiency of the proposed approach to produce human actions synthesized with text descriptions. Finally, through the Transformation model, the generated data were converted to a set of joint angles of the target robot, being the robot's communicative gestures. By employing the generated human-like actions for robots, it suggests that robots' social cues could be more understandable by humans.**

I. INTRODUCTION

Non-verbal behaviors including facial and bodily expressions have an indispensable role for social robots, allowing them to naturally interact with humans in a facile and transparent manner [1]. Especially for social robots without dedicated facial articulation, communicative gestures endow them with the capability of using bodily expressions for emphasizing the meaning of their speech, describing something, or showing clear intention. These social skills could improve interacting partners' understanding and make interaction outcomes rewarding. In this paper, we propose an approach to learn the relation between human actions and natural language. The generated actions are used for social robots to convey the meaning of their speech. The approach is inspired by Conditional Generative Adversarial Network (CGAN) [2], an extension of Generative Adversarial Networks (GANs) [3]. The designed framework is built upon a Convolutional Neural Network (CNN), which has been used efficiently in many different GAN application domains. This paper investigates on the generative network built upon CNN for generating social robots' gestures synthesized with their verbal content of speech.

The rest of the paper is organized as follows. In Section II, we provide a review of previous studies in generating co-speech robot gestures inspired by the rule-based approach. We then emphasize the importance of utilizing human behaviors to generate robot gestures and recent studies based on the learning from the demonstration approach. In Section III, the proposed model is described in detail. In Section IV, we validate the proposed approach on a publicly available dataset. Finally, our conclusion and future works are described in Section V.

The authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, Japan {ngtvtuyen, aelibol, nakyoung}@jaist.ac.jp

II. RELATED WORK

The approach to communicative robot gesture generation can be divided into mainly two groups: rule-based and data-driven.

A. Rule-based Approach

Most of the existing works on generating communicative robot gestures rely on the ruled-based approach. Behavior Expression Animation Toolkit (BEAT) [4] is a well-known model. It receives speech texts as the inputs and releases non-verbal behaviors. In BEAT, the association between text and gesture is defined by a set of rules derived from state of the art in the non-verbal conversational behavior researches. Similarly, the model [5] accepts both speech text and the audio signal as the inputs, the rule-based system analyzes the input utterance text to generate the facial and bodily expressions for virtual agents. Recently, several social robots such as RoboThespian, Nao, and Pepper have become capable of making the communicative gestures synchronized with their speech. However, their gestures are handcrafted by animation experts to ensure the familiarity and human-likeness of the gesture.

B. Data-driven Approach

Although the handcrafted gestures provide the familiarity and human-likeness of the robots' motions, this approach only allows robots to produce their communicative behaviors in the pre-designed scenarios. Moreover, the generated gestures are constrained by a set of defined rules. It should be remarked that social robots need to be capable of interacting with different types of users in a personal way by adapting and learning new behaviors throughout their lifetime [6], [7]. Thus, robots should be endowed with the capability of learning social skills from human social interactions. In order to attain this objective, the relation between gestures and corresponding natural language context needs to be addressed in a variety of communication topics. In [8], the mapping between bodily gestures and natural language has been investigated. The model receives text descriptions as inputs and produces gestures performed by the Master Motion Map (MMM) model. Since the output actions are represented in the joint space of the MMM model, it is

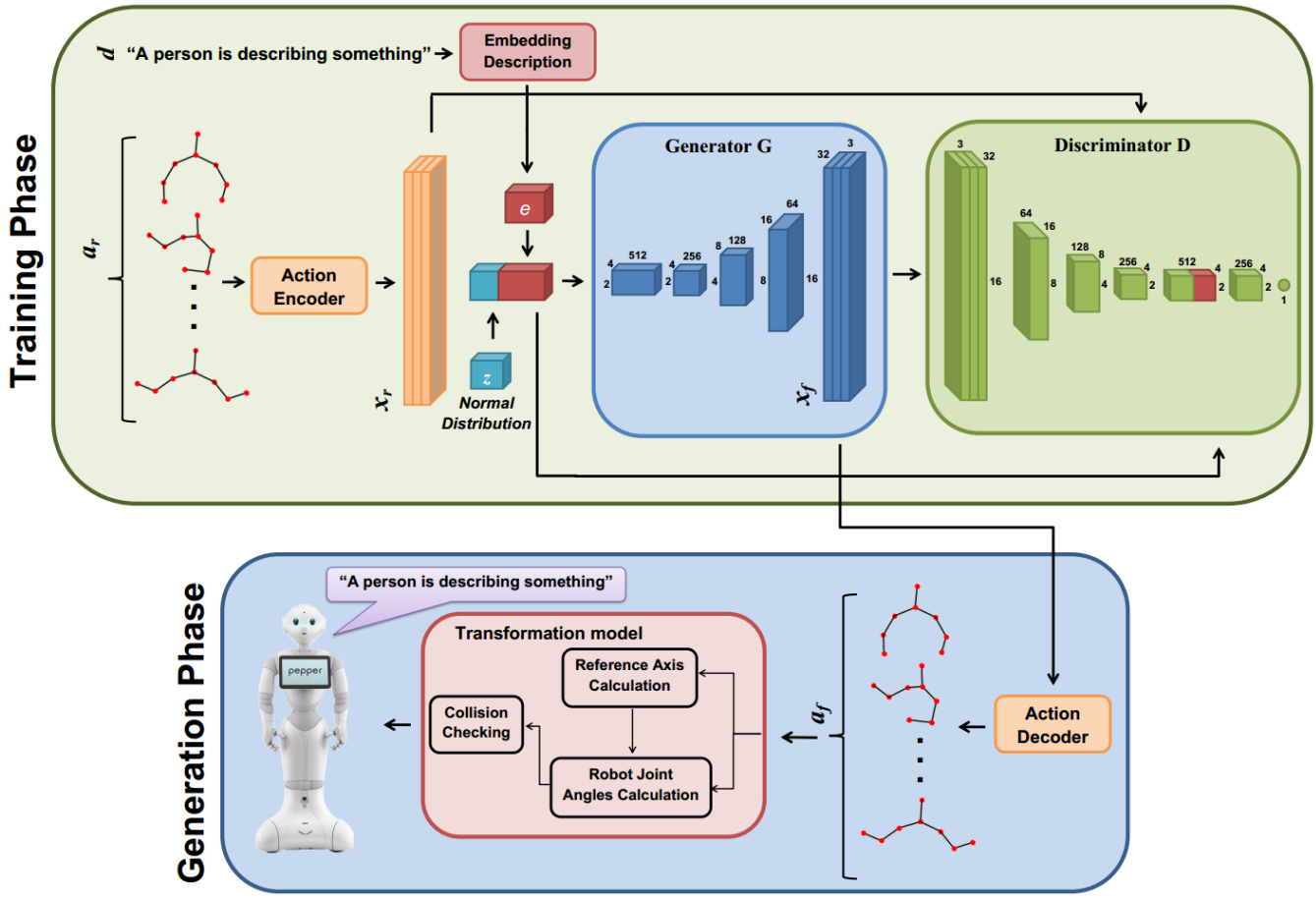


Fig. 1: The designed framework for generating gestures $\boldsymbol{a_f}$ synthesized with the text description d.

difficult to implement the generated gestures of this approach on the other robots whose joint configurations are different from the MMM framework. To address this drawback, our approach generates actions represented in Cartesian space, allowing them to be transformed effectively into different social robot platforms. Recently, GAN has received considerable attention in a variety of research contexts. Especially, it has been applied with great success in image generation task [9]. In terms of action generation, the author [10] proposed the GAN framework named Text2Action to generate co-speech actions. It is constructed based on a sequence to sequence network. Different from Text2Action, our generative framework is built upon CNN, which has been widely used in many application domains such as image [11], [12], video [13], and audio generation [14]. As a result, this paper investigates the convolution operation toward the autonomous generation of communicative gestures.

III. METHODOLOGY

Overview of the Proposed Approach

Fig. 1 presents the proposed framework. In the training phase, $\boldsymbol{a_r} = [S_1, S_2, S_3, ..., S_T]$ ($\boldsymbol{a_r} \in \mathbb{R}^{3\times8\times T}$) denotes a real action containing a sequence of skeleton frames $S \in \mathbb{R}^{3\times8}$ performed over a period of time T. As shown in Fig. 2, S consists of 8 joints defined in 3D space. Through the Action Encoder, $\boldsymbol{a_r}$ is encoded to an action matrix $\boldsymbol{x_r} \in \mathbb{R}^{3\times16\times T}$. On the other hand, $d = [w_1, w_2, w_3, ..., w_k]$ is a natural language sentence composed of k words to describe the action $\boldsymbol{a_r}$. It is started by feeding the description d to the Embedding Description network. The output e is concatenated with the noise vector z sampled from the Normal distribution, and they are fed to the Generator network. The purpose of the Generator G is to generate the fake action matrix $\boldsymbol{x_f} \in \mathbb{R}^{3\times16\times T}$ as much realistic as possible to beat the Discriminator D while D tries to differentiate between $\boldsymbol{x_r}$ and $\boldsymbol{x_f}$ taking into account the embedding vector e.

Once the training process is completed, the generated action matrix $\boldsymbol{x_f}$, synthesized with text description d, is decoded to $\boldsymbol{a_f} \in \mathbb{R}^{3\times8\times T}$. Through the Transformation model [15], the action $\boldsymbol{a_f}$, defined in 3D Cartesian space, is transformed into the target robot's motion space represented by joint angles. The following parts will detail the designed framework shown in Fig. 1.

A. Embedding Description

In order to capture the meaning of text description $d = [w_1, w_2, w_3, ..., w_k]$ by a fixed-length vector, we use the encoder phase of the skip-thoughts model [16]. The hidden layer h_k, encoded from a sequence of words $\{w_1, ..., w_k\}$, is determined by Eq. 1. Here, c_k is the word embedding of w_k, W and U are the weight matrices, $\odot$ denotes the component-

wise product, z_k and r_k are the update gate and reset gate of Gated Recurrent Unit [17].

$$h_k = (1-z_k)\odot h_{k-1} + z_k \odot tanh(Wc_k + U(r_k \odot h_{k-1})) \quad (1)$$

It has been shown that h_k effectively represents the semantics and syntax of the whole sentence to be encoded [16]. Then, h_k is compressed to a smaller dimensional vector e before being fed into the G and D network.

B. Action Encoder and Decoder

Action Encoder: It has been shown that CNN has a natural ability to learn representation from 2D matrices [18]. Human actions, defined as a sequence of skeleton frames, could be represented as 2D matrices containing three channels representing x, y, z coordinates, respectively. On each channel, the horizontal axis covers the time sequence T, while the vertical axis represents the spatial distribution of joints at a certain timestamp. Then, CNN based approach is utilized to jointly capture spatial and temporal information of actions [18], [19], [20], [21]. It should be emphasized that the chain order of joints in the vertical axis affects the spatial information represented in the action matrix $\boldsymbol{x_r}$. To efficiently capture spatial relations of the adjacent joints of the action $\boldsymbol{a_r}$, the Action Encoder puts its relative joints near each other. With this representation, by feeding the input $\boldsymbol{a_r}$ to the Action Encoder, the encoded matrix $\boldsymbol{x_r}$ is released and it can be seen in Fig. 2. Specifically, on each channel $c \in \{x, y, z\}$ of the matrix $\boldsymbol{x_r}$, the horizontal axis covers the time sequence T of the action $\boldsymbol{a_r}$, while the vertical axis is a sequence of joints in a given order $I = [1, 0, 1, 2, 3, 4, 3, 2, 1, 1, 5, 6, 7, 6, 5, 1]$ ($I \in \mathbb{R}^{16}$) at a certain timestamp. Thus, instead of feeding the raw input $\boldsymbol{a_r}$ to the D network, the Action Encoder allows the spatial-temporal information of the action $\boldsymbol{a_r}$ to be presented as the action matrix $\boldsymbol{x_r}$.

Action Decoder: In order to decode the action matrix $\boldsymbol{x_f}$ to the action $\boldsymbol{a_f}$ as displayed in Fig. 1, our designed Action Decoder calculates the joint value $j_{c,m,t}$ of the action $\boldsymbol{a_f}$ over the time sequence as shown in Eq. 2 and Eq. 3. This calculation allows that $j_{c,m,t}$ is defined based on the average values of its distribution on $\boldsymbol{x_f}$. Here, $j_{c,m,t}$ denotes the value of joint index m ($m = [0, 7]$), on the dimension c ($c \in \{x, y, z\}$), at the time stamp t ($t \in [1, T]$), and $n(m)$ is the number of times the joint index m in the order I.

$$j_{c,m,t} = \frac{1}{n(m)} \sum_{p=1}^{16} x_f(c, p, t)\delta(p, m, I) \quad (2)$$

$$\delta(p, m, I) = \begin{cases} 1 & I(p) = m \\ 0 & I(p) \neq m \end{cases} \quad (3)$$

C. Generator and Discriminator Network

In this paper, the proposed Generator G and Discriminator D are based on CNN similar to our previous work [21]. Initially, the noise vector z is sampled from the Normal distribution $N(0, 1)$. It is concatenated with the vector e, encoded from Embedding Description, before being fed to the G network. As presented in Fig. 1, G is designed with a fully connected layer to reshape the input vector and followed by four fractionally-strided convolutions to up-sample the data to an output target $\boldsymbol{x_f}$. On each layer, batch normalization is applied for stabilizing the learning process, and followed by the Rectified Linear Unit (ReLU) activation [22] except for the last layer. Here, the $tanh$ activation function is used before producing the fake action matrix $\boldsymbol{x_f}$.

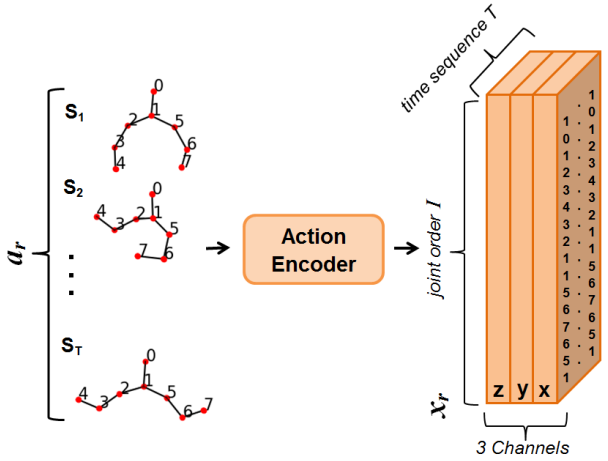


Fig. 2: Action Encoder encodes the raw action $\boldsymbol{a_r}$ to the action matrix $\boldsymbol{x_r}$

Discriminator D is designed with five convolutional layers similar to the architecture of G. D receives either $\boldsymbol{x_r}$ from training data or $\boldsymbol{x_f}$ from G as an input. At the fourth layer, the embedding vector e is concatenated with the output of the convolutional layer. Here, the embedding e provides conditional information for D to evaluate whether the input action satisfies this condition or not. At the last layer, the results are passed into a sigmoid function to produce an output probability.

The training process is summarized in Algorithm 1. The vector e provides conditional information to the G network in order to generate the action matrix $\boldsymbol{x_f}$, synthesized with the action description d. The Generator aims to fool the Discriminator, it is trained to maximize the output probability, y_f. Conversely, D is trained to differentiate between $\boldsymbol{x_r}$ and $\boldsymbol{x_f}$ based on (1) the human-likeness of the action, and (2) the synthesis of an action and its corresponding description. It should be remarked that the second point plays an essential role, allowing the generated action to express the meaning of the input description effectively. To endow D with the capability of evaluating this synthesis, D is trained to maximize the output probability y_r when receiving a pair of real action input $\boldsymbol{x_r}$ and embedding vector e. On the other hand, given a pair of input $\boldsymbol{x_f}$ and e, the Discriminator is trained to minimize the output probability y_f. From the training data, we also collect the miss-matching description $\hat{d}$, which incorrectly describes the action $\boldsymbol{x_r}$. When feeding a pair of the real action $\boldsymbol{x_r}$ and $\hat{e}$ to the D network, the

Discriminator is trained to minimize the output y_m, implying that $\boldsymbol{x_r}$ does not synthesize $\hat{d}$. The binary cross-entropy is applied to compute the miss-classification error L_D, L_G of the network D, and G, respectively. The parameter of D is updated while keeping the parameters of G constant. Then, the parameters of G are adjusted to optimize the error L_G while keeping network D unchanged.

Algorithm 1 The proposed algorithm for the training phase

Input: real action a_r, matching description d, miss-matching description $\hat{d}$, training batch steps S.

for s=0 to S **do**
  $x_r \leftarrow$ ActionEncoder(a_r);
  $e \leftarrow$ EmbeddingDescription(d);
  $\hat{e} \leftarrow$ EmbeddingDescription($\hat{d}$);
  $z \leftarrow$ N(0, 1);
  $x_f \leftarrow$ G(z, e);
  $y_r \leftarrow$ D(x_r, e);
  $y_f \leftarrow$ D(x_f, e);
  $y_m \leftarrow$ D($x_r, \hat{e}$);
  $L_D \leftarrow log(y_r) + log(1 - y_m) + log(1 - y_f)$;
  $D \leftarrow D - \alpha \partial L_D / \partial D$; {Update Discriminator}
  $L_G \leftarrow log(y_f)$;
  $G \leftarrow G - \alpha \partial L_G / \partial G$; {Update Generator}
end for

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Dataset and Preprocessing

The designed framework was validated on the same dataset as applied in [10]. This dataset consists of $2,822$ actions $\boldsymbol{a_r}$ and $31,863$ corresponding natural language descriptions d (one action could be associated with more than one description). As shown in Fig. 2, $\boldsymbol{a_r} \in \mathbb{R}^{3\times8\times32}$ is a sequence of 32 skeleton frames representing the human upper body motion. Each frame S includes 8 joints defined in 3D. From the dataset, we filtered the actions whose joint positions are out of the range $[-1, 1]$. Totally, $29,663$ pairs of actions $\boldsymbol{a_r}$ and corresponding descriptions d were obtained. For each $\boldsymbol{a_r}$, we also collected the miss-matching description $\hat{d}$. The obtained data $\boldsymbol{a_r}$, d, and $\hat{d}$ were split into 90% for training and 10% for testing. Concerning the Embedding Description, as mentioned in Section III-A, we used the encoder phase of the skip-thoughts model trained with the BookCorpus dataset [23]. As the BookCorpus dataset consists of $11,038$ books in a variety of topics, it allows the encoded vectors to effectively capture the semantics and syntax of the input sentences, without being biased toward any particular domain.

B. Evaluation Metrics

Consider that $\boldsymbol{a_r} = [S_1, S_2, S_3, ..., S_T]$ is the real action associated with the description d, and $\boldsymbol{a_f} = [S'_1, S'_2, S'_3, ..., S'_T]$ is the fake action synthesized with d. In order to verify the synthesis between $\boldsymbol{a_f}$ and d quantitatively, we used covariance description with temporal hierarchical construction [24] to evaluate how similar the generated action

Fig. 3: Skeleton sequence of generated action for "a young woman demonstrates example of lifting exercises."

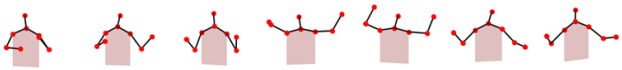


Fig. 4: Generated action for "a girl practices lifting exercise at the gym."

Fig. 5: Generated action for "a woman performs weight lifting exercises."

$\boldsymbol{a_f}$ and the real action $\boldsymbol{a_r}$ are. Given $\boldsymbol{a_r}$ and $\boldsymbol{a_f}$ as the inputs, Eq. 4 encodes them as the corresponding feature vectors C_r and C_f, respectively. Here, $\overline{S}$ is the sample mean of S_i computed over the time T and $^\intercal$ represents the transpose operator. This feature vector efficiently captures spatio-temporal information of action over the time sequence, it has been used for action recognition tasks [24] and unsupervised learning tasks [7]. Finally, the similarity between C_r and C_f is measured by cosine similarity as given in Eq. 5.

$$C = \frac{1}{t-1}\sum_{i=1}^{T}(S_i - \overline{S})(S_i - \overline{S})^\intercal \tag{4}$$

$$Similarity(C_r, C_f) = \frac{C_r \cdot C_f}{||C_r||\,||C_f||} \tag{5}$$

C. Generated Actions Synthesized with Input Descriptions

From the training data, the real action $\boldsymbol{a_r}$, the matching description d, and the miss-matching one $\hat{d}$ were fed to the designed network with the batch size 100. The dimension of the noise vector z is 100. The Adam optimizer [25] with the momentum 0.5 and the learning rate 2×10^{-5} was applied for both G and D network. The Discriminator and Generator were sequentially trained for 700 epochs.

Fig. 3 illustrates the generated action of the proposed model by feeding an input "a young woman demonstrates example of lifting exercises", which is included in the testing data. The action looks like a person is lifting two arms over the shoulder two times. Moreover, we also tested the two modified versions of that sentence such as "a girl practices lifting exercise at the gym" and "a woman performs weight lifting exercise". The resulting actions are presented in Fig. 4 and Fig. 5, respectively. A closer look at Fig. 3, 4, and 5 show that skeleton frames of those actions are not exactly matched to each other at a certain timestamp. However, generated bodily expressions seem to be similar over the time sequence. Those results suggest that G does not merely memorize and reproduce the data. It is able to generate a diverse set of actions to convey a particular meaning. For social robots, this capability would allow them to perform novel behaviors over

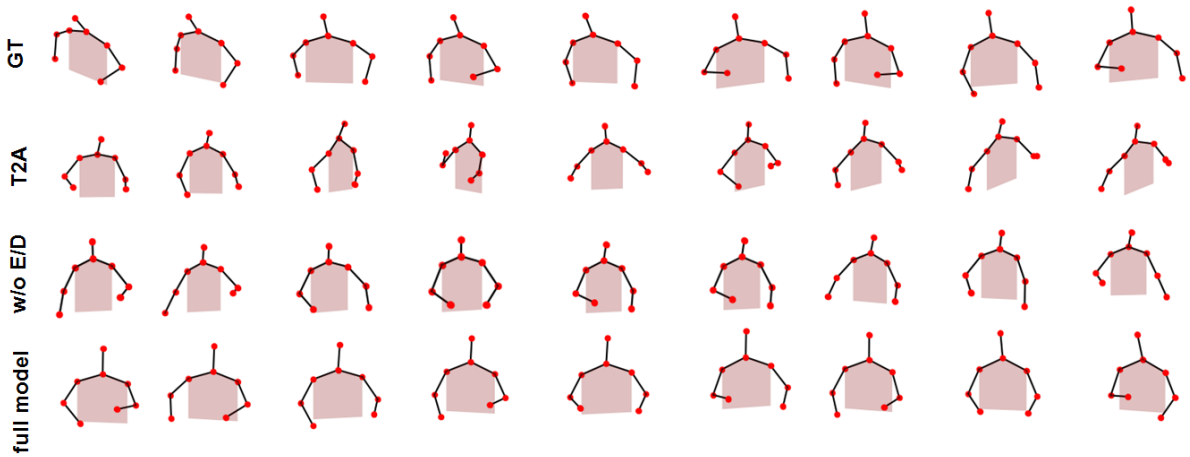


Fig. 6: Comparison with the real action (GT) for "a sprinter is sprinting on the track with his head down": Text2Action (T2A) [10], the model without Action Encoder/Decoder (w/o E/D) [21], and the fully implemented model (full model).

TABLE I: Similarity comparison among Text2Action [10] (T2A), the model without Encoder/Decoder [21] (w/o E/D), and the fully implemented model (full model).

	Text2Action	w/o E/D	full model
Average similarity	0.4196	0.5060	0.5287

time, which positively contributes to the user's engagement during interaction [26].

D. Quantitative Evaluation of Generated Actions

The real action $\boldsymbol{a_r}$ is correctly synthesized with the text description d. Thus, it is reasonable for evaluating actions produced by the G network by measuring the similarity between $\boldsymbol{a_r}$ and $\boldsymbol{a_f}$, since those are synthesized with the same description d. Notice that $\boldsymbol{a_r}$ and $\boldsymbol{a_f}$ could express the same meaning over the time sequence, although their corresponding skeleton frames are not exactly matched each other at a certain timestamp, as mentioned in IV-C. The evaluation metric suggested in IV-B satisfies such requirement. Here, we sequentially fed text descriptions of the testing data to a given G network. Both the generated and real actions were plugged into Eq. 4 and Eq. 5 for measuring their similarity.

In order to quantitatively verify the differences between our proposed network and the related approach - Text2Action [10], we trained their proposed network again on this training data while keeping the same training parameters as suggested by the authors. Additionally, we also verified the efficiency of the action generation framework without the Action Encoder and Decoder, which is described in our previous work [21]. Specifically, the raw action $\boldsymbol{a_r}$ was fed to the designed network without encoding. Then, the action $\boldsymbol{a_f}$ is generated from G without passing through the Action Decoder. Table I presents the similarity between the real actions of testing data and actions generated from Text2Action, our model without Action Encoder/Decoder, and fully implemented model, respectively.

Table I indicates that by feeding the same text descriptions, the generated actions produced from our networks are more similar to the real ones. Thus, our generated data are more connected to the input sentences. It should be emphasized that our D network is trained to differentiate between data generated by G and the real training data taking into account the description d as similar as applied in [10], [21]. Additionally, D is trained to detect the error when the real action is associated with the miss-matching text $\hat{d}$. This strategy enables the Discriminator capable of evaluating the synthesis between a given action and a conditional input in a more efficient way. On the other hand, Table I indicates that the fully implemented framework yields higher accuracy than the one without Action Encoder and Decoder. The experiment showed that by feeding the raw input $\boldsymbol{a_r}$ to the framework as applied in [21], the training process was faster since the Action Encoder encodes $\boldsymbol{a_r}$ as $\boldsymbol{x_r}$, which is the higher dimension matrix. However, by distributing the relative joints near each other as in $\boldsymbol{x_r}$, it allows the spatial and temporal information of $\boldsymbol{a_r}$ to be represented better. Thus, D could detect the motion properties of the input action faster and more efficiently. Consequently, D provides more informative feedback to G, for optimizing the generated action. Fig. 6 displays an example of feeding a sentence "a sprinter is sprinting on the track with his head down" to the three G networks. The real sample indicates a person that is pumping two hands up and down while the head is bent down slightly. Although the posture of bending his/her head down is unsuccessfully expressed neither by the three generated actions, those actions look like persons are pumping two hands up and down several times. Especially with the fully implemented model, the action is more natural and similar to the real one.

E. Transforming Generated Actions into the Pepper Robot

The action $\boldsymbol{a_f}$ defined in 3D Cartesian space, through the Transformation model [15], it is converted to a set of joint angles for controlling the upper body expression of the target robot. Fig. 7 and 10 present the actions produced from our G network, given input sentences included in the testing data. The generated actions are performed by the Pepper humanoid

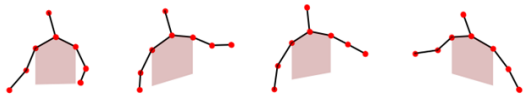


Fig. 7: Generated action for "the man rides on the surf board in the water" from the proposed network

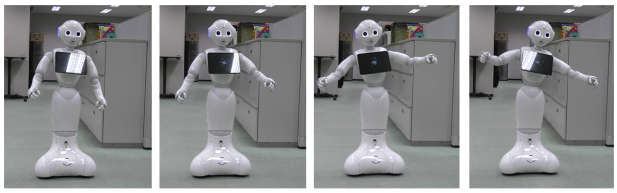


Fig. 8: Action for "the man rides on the surf board in the water" displayed on the Pepper robot

Fig. 9: Action for "the man rides on the surf board in the water" from the NAOqi API *ALAnimatedSpeech*

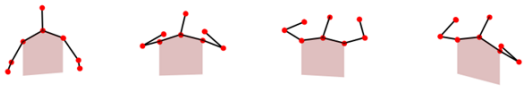


Fig. 10: Generated action for "one girl is dancing to music" from the proposed network

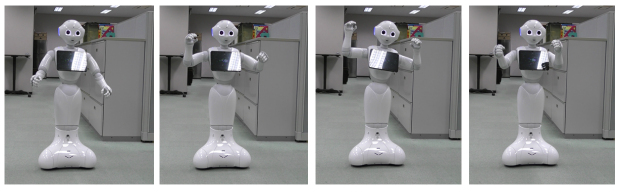


Fig. 11: Action for "one girl is dancing to music" displayed on the Pepper robot

Fig. 12: Action for "one girl is dancing to music" from the NAOqi API *ALAnimatedSpeech*

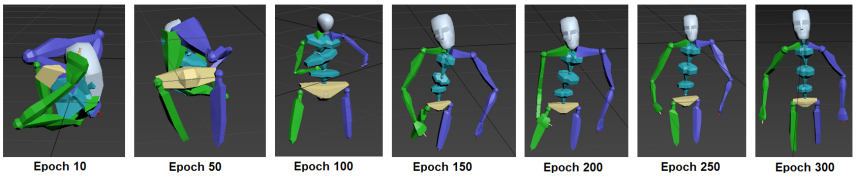
robot as shown in Fig. 8 and 11. Notice that we used the robot's on-board module ALTextToSpeech[1] to enable the robot to utter input sentences while performing bodily expressions. In order to see the differences between our approach and the robot's NAOqi API *ALAnimatedSpeech*[2], the same sentences were fed to that module. Fig. 9 and 12 show the generated actions.

In Fig. 7, given the input sentence "the man rides on the surf board in the water", the generated human action seems like a person with his/her arms fully extended to maintain the balance while the body orientation keeps changing. A similar bodily expression is performed on the robot as shown in Fig. 8. In Fig. 10, the action of "one girl is dancing to music" is expressed by the exaggerated movements of two hands. That action is performed by the Pepper robot as in Fig. 11. Compared to our generated actions, actions produced by the robot's NAOqi API *ALAnimatedSpeech* are mostly not related to the input descriptions. The robot's actions as shown in Fig. 9 and 12 could be observed that a person is describing or conveying something by slight movements of his/her hands. The experiment results revealed that actions produced from the robot's on-board module are not appropriately fit to the robot spoken texts. The reason is that that *ALAnimatedSpeech* consists a set of actions handcrafted by animation experts to ensure the familiarity and human-likeness of the robot's motions. By feeding an input text, random actions could be produced if certain keywords are not detected from the sentence. Thus, this approach only allows robots to produce limited behaviors within specific contexts. It should be remarked that generating appropriate behaviors is considered as an important strategy to maintain the quality of social human-robot interaction [26]. In our proposed approach, embedding vectors capture the meaning of whole input sentences, instead of certain keywords. They are used as essential materials to produce the output data. Consequentially, the robot's actions produced by our approach more appropriately express the meaning of text inputs.

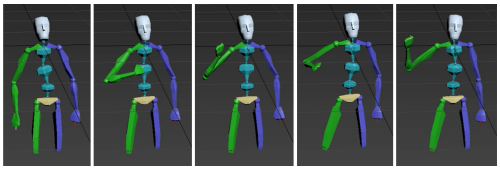
V. CONCLUSION AND FUTURE WORKS

This paper presents an approach to generate communicative gestures for social robots by imitating human behaviors. The generative framework was inspired by CGAN and built upon CNN. It receives a speech text as determining information to control the generated data. The proposed approach was validated on a public dataset. The experimental and comparative results with related works verified the efficiency of the proposed framework. Furthermore, the generated actions were transformed into the target robot motions taking into account the robot's physical constraints, and synchronized with the robot's speech. It has been shown that the proposed framework enables the robot capable of performing bodily expressions to better convey the meaning of their speech. By investigating the connection between human bodily expressions and natural language to generate robots' communicative gestures, this approach allows messages encoded in robots' behaviors are more recognizable to human perception. Our future work aims to utilize the fully designed framework for generating actions defined in a higher "resolution", as shown in Fig. 13. By producing actions employing a higher number of joints, it suggests that sophisticated contexts of input sentences could be expressed transparently.

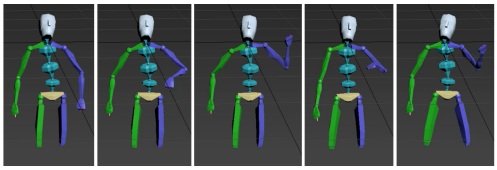
[1]http://doc.aldebaran.com/2-5/naoqi/audio/altexttospeech.html
[2]http://doc.aldebaran.com/2-5/naoqi/audio/alanimatedspeech-api.html



(a) The generated actions throughout the training process



(b) "A person waves with its right hand"



(c) "A person waves with the left hand"

Fig. 13: The proposed framework including Action Encoder/Decoder is validated on the KIT dataset [27]. The generated action $\boldsymbol{a_f} \in \mathbb{R}^{3\times20\times240}$ is defined by 20 joints in 3D Cartesian space. Fig. 13a shows the action $\boldsymbol{a_f}$ produced from G network throughout the training process. Fig. 13b and 13c presents the generated actions to express the meaning of the given sentence inputs.

ACKNOWLEDGMENT

The authors are grateful for financial support from the Air Force Office of Scientific Research under AFOSRAOARD/FA2386-19-1-4015 and the Shibuya Science, Culture, and Sports Foundation 2019 Grant Program.

REFERENCES

[1] M. Salem, F. Eyssel, K. Rohlfing, S. Kopp, and F. Joublin, "To err is human (-like): Effects of robot gesture on perceived anthropomorphism and likability," *Int'l Journal of Social Robotics*, vol. 5, no. 3, pp. 313–323, 2013.

[2] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[3] 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*, 2014, pp. 2672–2680.

[4] J. Cassell, H. H. Vilhjálmsson, and T. Bickmore, "Beat: the behavior expression animation toolkit," in *Life-Like Characters*. Springer, 2004, pp. 163–185.

[5] S. Marsella, Y. Xu, M. Lhommet, A. Feng, S. Scherer, and A. Shapiro, "Virtual character performance from speech," in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2013, pp. 25–35.

[6] C. L. Breazeal, *Designing sociable robots*. MIT press, 2002.

[7] N. T. V. Tuyen, A. Elibol, and N. Y. Chong, "Learning bodily expression of emotion for social robots through human interaction," *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[8] M. Plappert, C. Mandery, and T. Asfour, "Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks," *Robotics and Autonomous Systems*, vol. 109, pp. 13–26, 2018.

[9] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017.

[10] H. Ahn, T. Ha, Y. Choi, H. Yoo, and S. Oh, "Text2action: Generative adversarial synthesis from language to action," in *IEEE Int'l Conf. on Robotics and Automation*, 2018, pp. 5915–5920.

[11] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[12] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[13] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances in Neural Information Processing Systems*, 2016, pp. 613–621.

[14] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," *arXiv preprint arXiv:1802.04208*, 2018.

[15] N. T. V. Tuyen, S. Jeong, and N. Y. Chong, "Emotional bodily expressions for culturally competent robots through long term human-robot interaction," in *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2018, pp. 2008–2013.

[16] 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*, 2015, pp. 3294–3302.

[17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[18] Z. Yang, Y. Li, J. Yang, and J. Luo, "Action recognition with spatio–temporal visual attention on skeleton image sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2405–2415, 2018.

[19] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–11, 2016.

[20] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 3288–3297.

[21] N. T. V. Tuyen, A. Elibol, and N. Y. Chong, "Learning from humans to generate communicative gestures for social robots," in *2020 17th International Conference on Ubiquitous Robots (UR)*. IEEE, 2020, pp. 284–289.

[22] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Int'l Conf. on Machine Learning*, 2010, pp. 807–814.

[23] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *IEEE Int'l Conf. on Computer Vision*, 2015, pp. 19–27.

[24] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations," in *Int'l Joint Conf. on Artificial Intelligence*, vol. 13, 2013, pp. 2466–2472.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] I. Leite, C. Martinho, and A. Paiva, "Social robots for long-term interaction: a survey," *Int'l Journal of Social Robotics*, vol. 5, no. 2, pp. 291–308, 2013.

[27] M. Plappert, C. Mandery, and T. Asfour, "The kit motion-language dataset," *Big data*, vol. 4, no. 4, pp. 236–252, 2016.