



HAL
open science

SRG3: Speech-driven Robot Gesture Generation with GAN

Chuang Yu, Adriana Tapus

► **To cite this version:**

Chuang Yu, Adriana Tapus. SRG3: Speech-driven Robot Gesture Generation with GAN. 16th International Conference on Control, Automation, Robotics and Vision, Dec 2020, Shenzhen (virtual), China. hal-03047565

HAL Id: hal-03047565

<https://hal.science/hal-03047565>

Submitted on 8 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SRG³: Speech-driven Robot Gesture Generation with GAN

Chuang Yu and Adriana Tapus

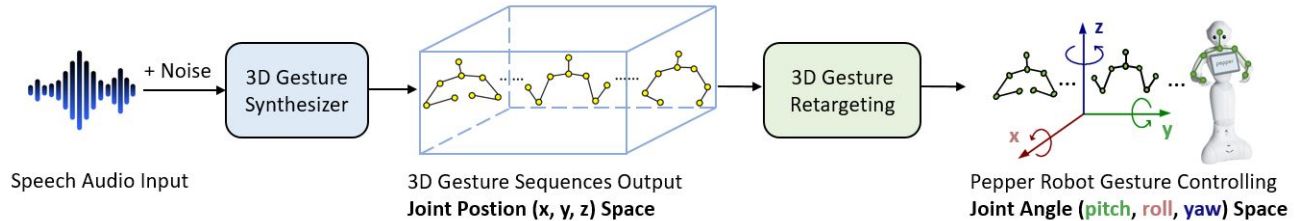


Fig. 1: Our robot gesture generation pipeline. The speech audio is used as an input to the 3D pose synthesizer with a random noise. Then, the natural and human-like 3D gesture sequences (in the joint position(x, y, z) space) are generated. Moreover, one same speech audio with multiple different random noises can align multiple natural gesture expressions in the same manner as humans have similar but different gestures while expressing the same speech in different contexts and situations. The gesture retargeting part maps the obtained gestures in the joint position space of the Pepper robot (i.e., in the joint angle(pitch, roll, yaw) space). Finally, the mapped positions are applied on the real Pepper robot in the human-robot interaction.

Abstract—The human gestures occur spontaneously and usually they are aligned with speech, which leads to a natural and expressive interaction. Speech-driven gesture generation is important in order to enable a social robot to exhibit social cues and conduct a successful human-robot interaction. In this paper, the generation process involves mapping acoustic speech representation to the corresponding gestures for a humanoid robot. The paper proposes a new GAN (Generative Adversarial Network) architecture for speech to gesture generation. Instead of the fixed mapping from one speech to one gesture pattern, our end-to-end GAN structure can generate multiple mapped gestures patterns from one speech (with multiple noises) just like humans do. The generated gestures can be applied to social robots with arms. The evaluation result shows the effectiveness of our generative model for speech-driven robot gesture generation.

I. INTRODUCTION

The gesture as a non-verbal body language is very important for a humanoid robot communication during human-robot interaction [1] [2] [3] [4]. Gestures associated and aligned with speech make a robot more expressive than by using only-speech communication during human-robot interaction [5]. It is very challenging to deal with speech-gesture synchronization problem during the gesture generation with speech [6] [7]. In the past, the natural and human-like robot gestures were mostly handcrafted by researchers again and again, process that is time-consuming and that needs to use prior knowledge in the related domain. Hence, the end-to-end automatic methods of the speech to gesture generation are more and more attracting the attention from researchers. Because the non-verbal behaviors have a random

variation to some degree, humans express alternative and variable instead of repeated speech-driven behaviors [8]. The speech-driven gesture for the robot as a non-verbal behavior should possess this kind of characteristic, namely that the robot can express multiple variable gesture patterns given the same speech in different contexts and situations. However, in the past, researchers focused only on one-to-one instead of one-to-many mapping between the speech and the gesture. Our paper focuses on the end-to-end speech-driven gesture generation for a humanoid robot. The pipeline of the robot gesture generation is as shown in Figure 1. Our 3D gesture synthesizer is based on the Generative Adversarial Network (GAN) [9], capable of using the speech audio as input to generate human-like natural 3D gesture sequences, which are in the 3D joint position (x, y, z) space. Then, these 3D positions are mapped to the joint angles(pitch, roll, yaw) of the Pepper robot gesture controlling by our gesture retargeting algorithm. Besides, our model can complete one-to-many mapping work, where one same speech audio with multiple random noises is used as an input to the gesture synthesizer in order to get multiple interrelated gesture sequences. Our contributions of this paper are as follows:

(1) We build-up an audio-visual database from the public YouTube TED video collection. The link of these videos is the following: <https://www.youtube.com/user/TEDtalksDirector/videos>. The database includes the speech audio data extracted directly from the videos and the associated 3D human pose data extracted from 2D RGB images.

(2) A new temporal GAN framework of the cross-modal generation is proposed for the alignment between the human-like gestures and the speech. Moreover, our model can use one speech audio (with multiple noises) to generate multiple human-like 3D gesture series, which can lead to a natural

This work was supported by China Scholarship Council (CSC).

C. Yu and A. Tapus are with Autonomous System and Robotics Lab/U2IS, Department of Computer Science and System Engineering, ENSTA Paris, Institut Polytechnique de Paris, 828 Blvd des Marechaux, Palaiseau, France chuang.yu@ensta-paris.fr; adriana.tapus@ensta-paris.fr

instead of boring and repeated human-robot interaction. The model solves the problem of how to add the noise for the generative model with time series source as the input because the temporal noise sequence is hard to generate for temporal GAN. In our generator of GAN, the audio speech as a kind of the time sequence input is represented by the encoder part as a vector, which contains all temporal information of the audio speech. Then the decoder of the generator takes the audio representation and the noise directly to generate the synchronous gesture sequence.

(3) The generated gestures given the speech audio are applied to Pepper robot after 3D pose retargeting processing from the generated 3D pose to the Pepper robot pose, which certifies the generated gesture’s effectiveness in reality.

In total, compared with traditional one-to-one mapping models from time series (for example the speech text and the speech audio) to time series (for example the action and the video), we came up with a new one-to-many solution for speech-driven gesture generation which is promising. As the noise of the conditional GAN model with time series as input is hard to be added towards the generator part [10], our GAN model can overcome this problem.

The rest of the paper is structured as follows: Section II shows the related works. Section III describes the methodology; Section IV presents our experiments and results. The conclusions and future work are part of Section V.

II. RELATED WORKS

Generative models including the deep belief networks [11], the variational autoencoder models [12], and the generative adversarial models [9] have attracted a lot of attention of researchers from many research areas including computer vision, robotics, and image processing. These have led to more and more excellent applications of audio-virtual correspondence, where a major focus was on the temporal mapping between the audio and the face action [13] [10] or the human Pose [14] [15]. For speech-driven motion generation, there are two methods based on the type of input speech, namely the speech with text and the speech input with audio.

Speech-driven Pose Generation with Text. Aly et al. [16] built a generation model with the tool BEAT (Behavior Expression Animation Toolkit) [17] to map a human’s verbal behavior to a corresponding combined robot’s verbal-nonverbal behavior, namely gesture, based on the personality dimensions of the interacting human. However, the BEAT toolkit is a rule-based method and generates the gesture without a random variation, which leads to a repetitive and boring experience in the lifelong human-robot interaction. Yoon et al. [18] built a large-scale co-speech gesture database including speech text and spontaneous 2D human gestures extracted from public videos of Youtube TED talks. The authors proposed an autoencoder model based on the *seq2seq* model [19] and the generative model used the speech text as the input to generate 2D human gesture sequence. The authors also applied the generated pose and the speech audio synthesized from text to the NAO robot for the participant

evaluation experiments. However, as mentioned in the paper, the method experienced an unnatural mapping problem where the generated gesture and the speech audio could not be tightly mapped well during the experiments. For the same text sentence, different people can speak with variable emotional states and speeds, which should correspond to different poses. However, this speech text-based approach can align only one pose pattern with only one speech sentence, which leads to an unnatural and not human-like human-robot interaction. Speech audio-based method can deal with this well in order to generate different pose sequences based on the variable speech expression of speakers.

Speech-driven Pose Generation with Audio. Hasegawa et al. [20] used the Bi-Directional LSTM (Long Short-Term Memory) Network to generate the co-speech gestures. The model processed the speech audio into MFCC (Mel-Frequency Cepstral Coefficients) as the audio representation and also included LSTM regression part for the temporal mapping and a temporal filtering to remove noises of the 3D pose data. Using the same database, Kucherenko et al. [21] proposed an autoencoder-based approach to deal with the alignment between the speech audio and the human pose. One autoencoder network, namely *MotionED*, was trained for lower dimensional motion representation and one encoder was trained for the mapping the speech to motion representation obtained in *MotionED*. However, the two studies are one-to-one generation structure, which can generate only one gesture sequence for the same speech input.

III. METHODOLOGY

A. Problem Definition

Speech-driven Gesture Generation: This problem is an one-to-many task by nature to generate the diverse spontaneous gesture sequences $g^m = [g_t^m]_{t=1:T_0}$ with one speech audio $s = [s_t]_{t=1:T_1}$ and multiple noises n^m as inputs. Namely, the research tried to learn a mapping function $F_{generation}$, which will maximize the conditional probability $p(g^m|s)$.

$$\mathbf{g}^m = F_{generation}(\mathbf{s}, \mathbf{n}^m) \quad (1)$$

3D Gesture Retargeting: The objective is to map the generated gestures g^m with the position $p(x, y, z)$ in the joint position space towards the robot gestures r^m with the angle $a(pitch, roll, yaw)$ in the joint angle space, which can be used on the Pepper robot co-speech gesture controlling directly. The mapping consists in finding a retargeting function $F_{retargeting}$:

$$\mathbf{r}^m = F_{retargeting}(\mathbf{g}^m) \quad (2)$$

B. GAN Model for Speech-driven Gesture Generation

The first GAN model is created by Googfellow et al. [9]. GAN model trains two competitive networks, namely the generator G and the discriminator D in the same time, however G and D have opposite objectives. G tries to generate the data as the sample in the distribution of the training set

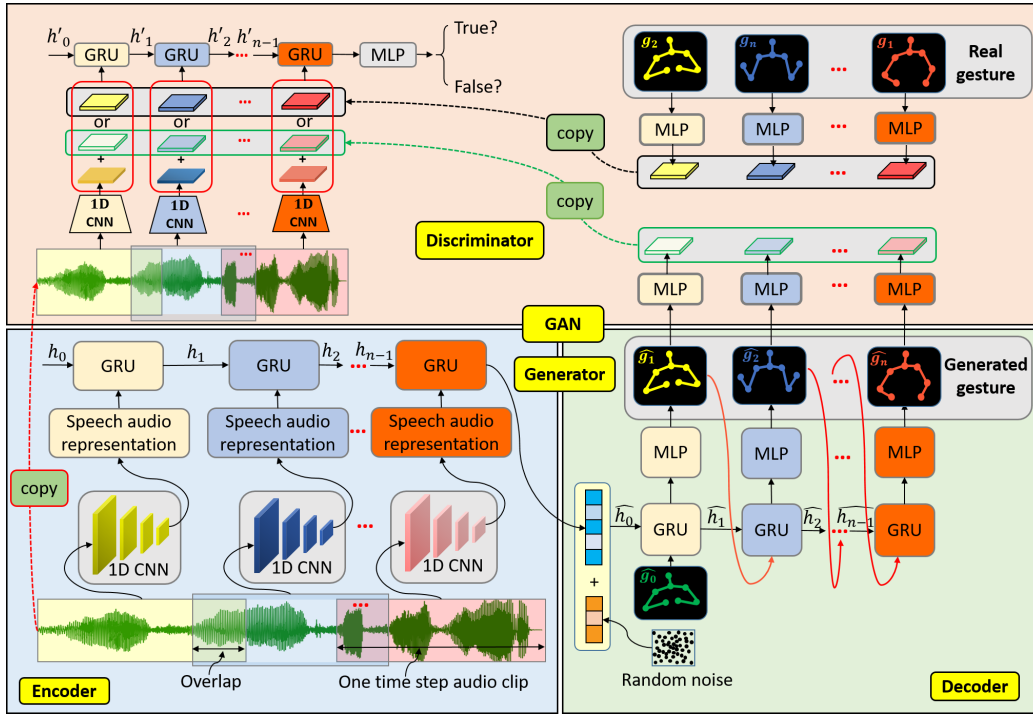


Fig. 2: An overview of our GAN model architecture for speech-driven gesture generation. The whole GAN model consists of a generator and a discriminator. The generator contains a temporal encoder and a temporal decoder. The encoder takes the speech audio as input to get the last hidden state as output for the later decoder input. The next decoder is applied to decode the input with the encoder output and a random noise towards the mapping gesture. The discriminator uses the generated gesture (or the ground-truth gesture) and the spontaneous speech audio as input to predict whether the speech and the gesture match with each other.

while D tries to differ the real data and the generated data. Nowadays, GAN model has been used in a lot of cross-modal generation problems, especially audio-visual mapping, for example audio-to-face and music-to-dance. The authors in [10] present an end-to-end co-speech face generation system based on the temporal GAN including a generator and 2 discriminators. A kind of music-to-dance model based on GAN was proposed in paper [14] that can synthesize realistic dances from music. In order to facilitate the co-speech gestures task, we built a new temporal generative adversarial network (GAN), which contains the generator and the discriminator.

In this paper, the proposed GAN model as shown in Figure 2 can produce diverse speech-matching gestures, which can be applied towards the humanoid robot. Our GAN model is named with *S2GGAN*, namely **S**peech to (**2**) **G**esture **GAN**. *S2GGAN* model contains a generator and a discriminator. The generator includes one encoder and one decoder and takes the speech audio as input and outputs the mapping gesture. The discriminator is used to differ whether the speech and the gesture match with each other.

1) *Generator*: The generator consists of an encoder and a decoder and uses the speech audio as input and outputs the spontaneous gesture sequence extracted from YouTube TED videos. As shown in Figure 2, the audio signal is divided into overlapped audio clips so that each audio clip is one-to-one correspondence with the gesture frames. Each clip has 8000 audio frames. In order to cope with the starting part and the

ending part of the whole audio signal, 4000 audio frames are padded to each end, respectively. The stride is the quotient of the audio frame rate divided by the video frame rate. Then, the audio clips are input to Convolutional Neural Network (CNN) [22] with 1D convolution to extract the speech audio representation, respectively as the 1D CNN can extract the 256 dimensional representation of the time series including the audio signal [23]. 1D CNN structure in the paper [10] is applied in our encoder network and it is composed of 1D convolution, a batch normalization part, and a ReLU part. The 1D CNN structure is as shown in Figure 3. Inspired from [24], the 1D CNN should start with a large kernel, which can make the low-level feature meaningful. In our paper, the first kernel size is 250 and all the followed kernel sizes are 4. We chose leaky ReLU activation [25] instead of a common ReLU because the latter experiences a dying ReLU problem and the former can overcome this problem during the training. Lastly, these representation results are fed into Gated Recurrent Unit (GRU) model [26] to get the 256 dimensional final hidden state as the encoding output of the encoder network.

The decoder network comprises one layer GRU and the Multilayer Perceptron (MLP). The input of the decoder is the output of the encoder adding the 10 dimensional random noise. As described in [27], the random noise can introduce some natural variability during the audio-visual generation task. We used multiple random noises to generate various kinds of mapping gestures from one speech audio. This can

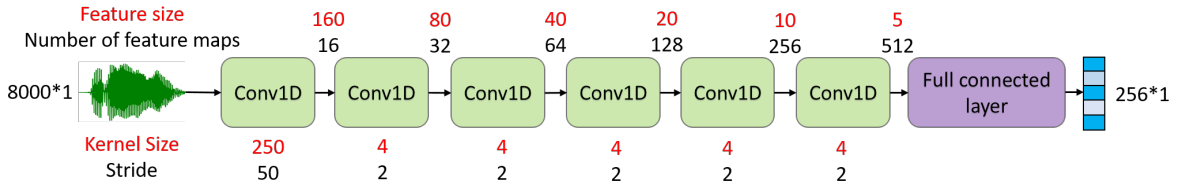


Fig. 3: 1D CNN network framework. 1D CNN model input is 8000 frame audio clip and output is 256 dimensional representation. This 1D CNN start with a big kernel, namely 250 and all the followed convolution operations have a small kernel size with 4. Among convolution operations, there are leaky ReLU operation and batch normalization operation. Finally, a full connected layer is used to get the 256 dimensional speech audio representation.[10]

produce various gestures for the robot and can lead to a natural long-life human-robot interaction. The output of the decoder is the spontaneous gesture sequence mapping with the speech audio in the encoder network. Each gesture frame contains 3D positions of 8 joints including the head, the spine shoulder joint, the left shoulder joint, the right shoulder joint, the left elbow joint, the right elbow joint, the left wrist joint, and the right wrist joint.

2) *Discriminator*: The discriminator works to distinguish whether the gesture sequence is realistic and whether the gesture sequence matches with the speech audio or not. Firstly, the 1D CNN, which is the same with the one in the generator, receives the speech audio divided into the overlapped clips as one of two inputs to get the 256 dimensional representation vectors. Meanwhile, each gesture frame of the real or generated gesture sequence is used as an input to the MLP layer to get 256 dimensional vectors. These vectors concatenate 256 dimensional speech representation vectors each by each respectively to get the 512 dimensional vectors as the input of one layer GRU. The GRU final hidden state is used as input for the MLP layer to distinguish whether the gesture matches with the speech audio.

3) *Objective function*: The objective function of our $S2GGAN$ contains two parts, namely conditional GAN loss part and L_1 loss part. The conditional GAN loss part can be expressed as shown in Eq. 3. In this equation, G tries to minimize the loss while D tries to maximize it.

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{s, g}[\log D(s, g)] + \mathbb{E}_{s, z}[\log(1 - D(s, G(s, z)))] \quad (3)$$

where s denotes the speech audio, g means the gesture sequence, and z is the random noise.

Previous works have found that it is beneficial to add L_1 loss towards the conditional GAN loss in image translation task [28]. In our work, the L_1 reconstruction loss is applied to enhance the realistic gesture generation at the frame level (see Eq. 4).

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{s, z, g}[\|g - G(s, z)\|_1] \quad (4)$$

The final objective of our $S2GGAN$ is as shown in Eq. 5. The hyperparameter λ is empirically set to 100 during $S2GGAN$ training, which is used to decide how much contribution L_1

or \mathcal{L}_{cGAN} make for all the loss.

$$G_{S2GGAN}^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \quad (5)$$

C. Gesture Retargeting

Mapping from 3D joint positions $p(x, y, z)$ to 3D robot joint angles $a(\text{roll}, \text{pitch}, \text{yaw})$ is a problem of robot kinematics. There are a lot of works showing how to transfer human pose with 3D positions to robot pose with joint motor angles [29] [30]. In our work, we use a Pepper robot to present the generated co-speech gesture. The Pepper is a social humanoid robot from SoftBank Robotics. In our research, 8 joint positions are generated, including the head, the spine shoulder joint, the left shoulder joint, the right shoulder joint, the left elbow joint, the right elbow joint, the left wrist joint, and the right wrist joint. Furthermore, 8 joint rotation angles of 4 joints should be obtained from the 3D positions in each gesture frame. These 8 joint angles of the robot upper body gesture can be obtained from the 3D positions of the 8 joints as shown in Figure 4.

The definitions and the angle ranges of the 8 joint rotation angles are shown in Figure 5 according to the Pepper robot official technique documents [31]. In order to make sure that the rotation axes in the left shoulder joint are in the rotation angle space, each unit vector along each positive direction of x axis, y axis, and z axis respectively should be determined. These three unit vectors are defined as shown in Eqs 6, 7, and 8 with $\vec{V}_{0,1}$ and $\vec{V}_{1,2}$.

And, unit vectors of x axis, y axis, and z axis in the right shoulder can be calculated with $\vec{V}_{0,1}$ and $\vec{V}_{1,5}$. From human kinesiology, the left and right shoulders do not have so much relative motion during speaking. In addition, Pepper robot cannot move his shoulders like human beings. In order to finish the mapping task from the joint positions to the joint angles, we simplify the solution where $\vec{V}_{1,5}$ is negative vector of $\vec{V}_{1,2}$. Then, we can use the unit vectors of x axis, y axis, and z axis in the left shoulder to calculate the angles for the right shoulder.

$$\vec{x} = \frac{\vec{V}_{0,1} \times \vec{V}_{1,2}}{|\vec{V}_{0,1} \times \vec{V}_{1,2}|} \quad (6)$$

$$\vec{y} = \frac{\vec{V}_{1,2}}{|\vec{V}_{1,2}|} \quad (7)$$

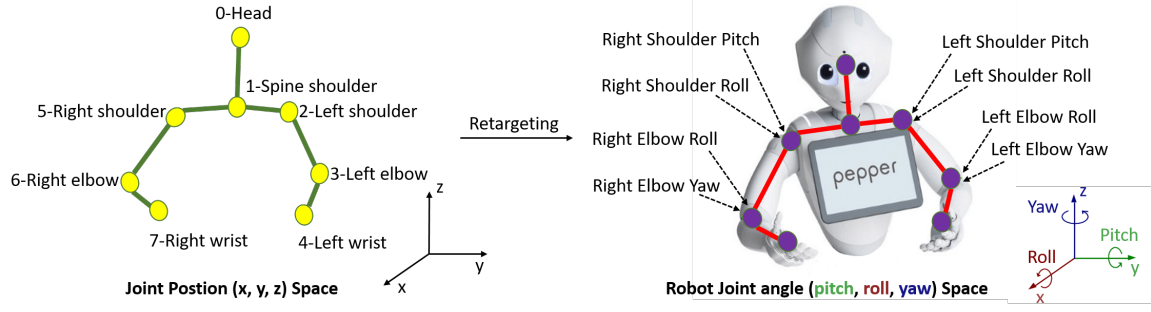


Fig. 4: An overview of the gesture retargeting process. In the joint position (x,y,z) space, there are 8 joints' 3D positions. The joints are the head (joint 0), the spine shoulder joint (joint 1), the left shoulder joint (joint 2), the right shoulder joint (joint 5), the left elbow joint (joint 3), the right elbow joint (joint 6), the left wrist joint (joint 4), and the right wrist joint (joint 7). Based on the robot kinematics, the 8 joint angles of the 4 joints are obtained from the 3D positions. The 8 joint angles are the left shoulder pitch, the right shoulder pitch, the left shoulder roll, the right shoulder roll, the left elbow roll, the right elbow roll, the left elbow yaw, and the right elbow yaw. Joint roll rotations take place around the X axis, joint pitch rotations around the Y axis, and joint yaw rotations around the Z axis.

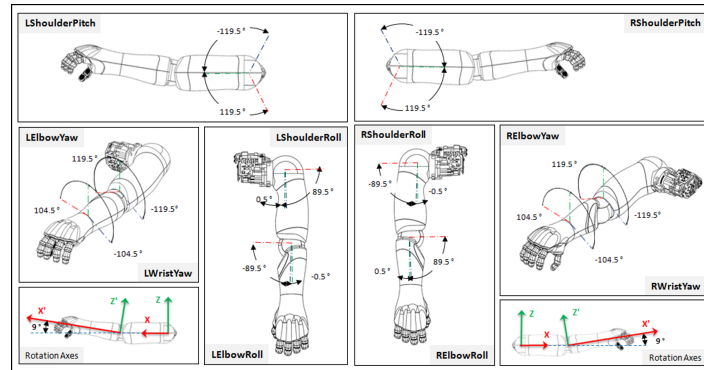


Fig. 5: The definition of the joint rotation angles and their rotation angle ranges in degree. [31]

$$\vec{z} = \frac{\vec{x} \times \vec{y}}{|\vec{x} \times \vec{y}|} \quad (8)$$

where $\vec{V}_{m,n}$ is a vector from the position of the joint m to the position of the joint n ($m, n = 0, 1, 2, 3, 4, 5, 6, 7$). The relation between the joint number and the joint name is as shown in Figure 4. In the mechanical structure of the physical Pepper robot, the spine shoulder joint, the left shoulder joint, and the right shoulder joint keep the fixed spatial relationship. In our research, we fix the position spine shoulder joint towards the position of midpoint of the left shoulder joint and the right shoulder joint.

For the left and right shoulder joints, there are 4 joint rotation angles that should be extracted, including the left shoulder roll, the left shoulder pitch, and the right shoulder pitch.

In order to simplify the calculation process, we used the unit vectors of the left shoulder joint rotation axes, namely \vec{x} , \vec{y} and \vec{z} , for these 4 angles calculation of both the left shoulder and the right shoulder joint. *LSR* (Left Shoulder Roll), *RSR* (Right Shoulder Roll), *LSP* (Left Shoulder Pitch) and *RSP* (Right Shoulder Pitch) are calculated as shown in Eqs. 9, 10, 11, and 12, respectively.

$$LSR = \frac{\pi}{2} - \csc^{-1} \left(\frac{\vec{y} \cdot \vec{V}_{5,6}}{|\vec{y}| \cdot |\vec{V}_{5,6}|} \right) \quad (9)$$

$$RSR = \csc^{-1} \left(\frac{\vec{y} \cdot \vec{V}_{2,3}}{|\vec{y}| \cdot |\vec{V}_{2,3}|} \right) - \frac{\pi}{2} \quad (10)$$

$$LSP = -\tan^{-1} \left(\frac{\vec{V}_{2,3} \cdot \vec{z}}{\vec{V}_{2,3} \cdot \vec{x}} \right) \quad (11)$$

$$RSP = -\tan^{-1} \left(\frac{\vec{V}_{5,6} \cdot \vec{z}}{\vec{V}_{5,6} \cdot \vec{x}} \right) \quad (12)$$

For the left and right elbow joint rotation angles, their related coordinate axes rotate because of the shoulder joints' rotation. Relative to the calculation of four elbow rotation angles, only the unit vectors of x axis should be transferred, including \vec{x}_{left} , and \vec{x}_{right} . The related calculation of the 4 elbow joint rotation angles are shown in the Eqs. 13 - 19. Where $\vec{v}_{plane234}$ is the normal vector of the plane defined by joint 2 (left shoulder), joint 3 (left elbow) and joint 4 (left wrist). $s(\alpha)$ and $c(\alpha)$ means the sine and cosine of α respectively. *LER*, *LER*, *LEY*, and *REY* are the left elbow roll angle, the right elbow roll angle, the left elbow yaw angle and the right elbow yaw angle, respectively.

$$LER = -\cos^{-1} \left(\frac{\vec{V}_{2,3} \cdot \vec{V}_{3,4}}{|\vec{V}_{2,3}| \cdot |\vec{V}_{3,4}|} \right) \quad (13)$$

$$RER = \cos^{-1} \left(\frac{\vec{V}_{5,6} \cdot \vec{V}_{6,7}}{|\vec{V}_{5,6}| \cdot |\vec{V}_{6,7}|} \right) \quad (14)$$

$$\vec{x}_{left} = \begin{bmatrix} c(LSR) & s(LSR) & 0 \\ -s(LSR) & c(LSR) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \vec{x} \quad (15)$$

$$\begin{bmatrix} c(LSP) & 0 & s(LSP) \\ 0 & 1 & 0 \\ -s(LSP) & 0 & c(LSR) \end{bmatrix} \cdot \vec{x}$$

$$\vec{v}_{plane234} = \frac{\vec{V}_{2,3} \times \vec{V}_{3,4}}{|\vec{V}_{2,3} \times \vec{V}_{3,4}|} \quad (16)$$

$$LEY = \cos^{-1} \left(\frac{\vec{x}_{left} \cdot \vec{v}_{plane234}}{|\vec{x}_{left}| \cdot |\vec{v}_{plane234}|} \right) - \frac{\pi}{2} \quad (17)$$

$$\vec{x}_{right} = \begin{bmatrix} c(RSR) & s(RSR) & 0 \\ -s(RSR) & c(RSR) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \vec{x} \quad (18)$$

$$\begin{bmatrix} c(RSP) & 0 & s(RSP) \\ 0 & 1 & 0 \\ -s(RSP) & 0 & c(RSR) \end{bmatrix} \cdot \vec{x}$$

$$\vec{v}_{plane567} = \frac{\vec{V}_{5,6} \times \vec{V}_{6,7}}{|\vec{V}_{5,6} \times \vec{V}_{6,7}|} \quad (19)$$

$$REY = \cos^{-1} \left(\frac{\vec{x}_{right} \cdot \vec{v}_{plane567}}{|\vec{x}_{right}| \cdot |\vec{v}_{plane567}|} \right) - \frac{\pi}{2} \quad (20)$$

IV. EXPERIMENTS AND RESULTS

A. Database

It is very hard to build-up a huge audio-visual database with the speech and the spontaneous gesture. Yoon et al. [18] proposed a new solution to build up a big speech-driven gesture database; they used 52 hours of public YouTube TED videos and related English transcripts. Using *OpenPose* [32], which is a popular Human Pose Estimation (open-source) library, 2D human poses were extracted from videos. Then, they built the 2D speech driven gesture with the speech text and the synchronous 2D pose. Inspired by this work, we built up our own speech-driven database with the speech audio and the spontaneous 3D gesture extracted from YouTube TED videos. Instead of speech text, we used speech audio in our research as the same speech text can be expressed as various speech audios. Instead of 2D gesture, we achieved 3D gesture through *3D-pose-baseline* [33] in our database. *3D-pose-baseline* is a lightweight library doing 2D-to-3D pose estimation and its trained model is open to use. However, the original model was trained with the whole body pose data. Our work focuses only on the upper body gesture because the whole body cannot be visualized in most of YouTube

TED videos. Hence, in order to complete our 2D-to-3D pose estimation task, we trained the *3D-pose-baseline* with 8 joints data from the *Human3.6M* dataset [34], which is an open 3D human pose database with 3.6 million human poses and corresponding images.

Not all the frames of the YouTube TED videos contain a speaker or the whole upper body with 8 joints. Moreover, we made our rules for the continuous clip selection based on the rules discussed in paper [18]. Firstly, all 3D gesture frames in a clip should contain all the eight joints in the upper body and also all 3D gestures of those joints can be detected in our 2D-to-3D pose estimation process. Considering the synchronization of the speech audio and the gesture in our *S2GGAN* model, we only selected the videos with a frame rate of 24 Hz (most of the videos are 24 Hz and few of them are 25 Hz). The clip should be more than five seconds. Furthermore, the upper body height should be half larger than the the height of 2D image in the video. Finally, the speaker in the selected video clip should face the camera and there should be no still frames where the speaker stays still without the gesture movements.

Based on these rule, our final speech-driven gesture database contains 5760 samples¹. The whole process of our speech-driven gesture database building is as follows.

- (1) Download 1760 public TED videos from YouTube through *YouTube Data API*.
- (2) Extract the speech audios from videos downloaded through the *FFmpeg* library [35].
- (3) Extract 2D gesture data of eight joints used in our study from YouTube videos through *OpenPose* library.
- (4) Transform the 2D gesture to the 3D gesture of eight joints by *3D-pose-baseline* model trained by ourselves. In 2D-to-3D pose estimation process, only 7 joints' 3D were predicted from 2D pose of 8 joints and the spin shoulder joint is the origin (0,0,0) in the 3D space.
- (5) Based on the rules of clip selection, we divided the data into clips, made synchronization processing with the audios and the 3D gestures, and built the database for speech-driven gesture generation task.

B. S2GGAN Training

Our audio-gesture database contains 5760 samples which were divided into 180 batches with the batch size of 32. The time steps of the gesture sequence were fixed to 126 and the time duration is of 5.25 seconds. Standardization operation are completed on the 3D gesture data before inputting *S2GGAN* model and batch normalization operation was used in 1D CNN part of *S2GGAN*, which both can effectively reduce the over fitting during model training based on our tuning experiments. We used Adam [36] as the optimization algorithm where the learning rate is 0.0002, the parameter β_1 , β_2 and ϵ are 0.5, 0.999, and 10^{-7} , respectively for both the generator and the discriminator in *S2GGAN* model. Our *S2GGAN* model is completed in TensorFlow 2.0 and the

¹Our new database is available upon request. Please contact the authors for more details.

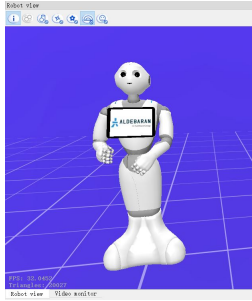


Fig. 6: The virtual Pepper robot runs the generated speech-driven gesture in the simulation environment.

training process with 4086 epochs was done on an NVIDIA GeForce RTX 2080 Ti GPU for about 1 week.

C. Results and Evaluation

After *S2GGAN* training, we tested the model to generate various speech-driven gesture sequences with two kinds of random noises. After pose retargeting from the joint positions to the joint rotation angles, we applied the joint rotation angles to run the Pepper robot in simulation environment-*ChoreGrappe* as shown in Figure 6 ².

1) *Qualitative Evaluation*: As any one-to-many generative model, our *S2GGAN* is able to generate multiple gesture sequences mapping with one same speech audio and multiple random noises. After pose retargeting operation on the gesture with positions, we obtained the joint rotation angles which can be applied to the Pepper robot. The generated samples with the joint angle LSR and RSR as shown in Figure 7, in which (a), (b), and (c) are relative to the ground truth, the generated gesture with noise 1 and the generated gesture with noise 2, respectively. We can find two key points: (1) The generated gestures by *S2GGAN* model can learn the whole gesture pattern or trend well in general. (2) Although the input is the same one speech audio, *S2GGAN* can generate two different gesture sequences with two different noises and the two gesture sequences are similar in general but with the random variation to some degree, which can lead to a natural lifelong human-robot interaction instead of a repetitive and boring one.

In addition, because the random noise is added to the initial hidden state \hat{h}_0 in the decoder of generator, the joint rotation angles of generated gestures in the first steps are not so stable. However, the situation will be improved with the time steps increasing as shown in RSR sequence of Figure 7 (c). The generated gesture data is more smooth because GAN model mostly focus on whether the speech audio and the gesture match with each other. Namely, GAN model pays more attention to the global alignment instead of the frame-level alignment compared with the autoencoder model.

2) *Quantitative Evaluation*: We also estimate the generated pose using a Average Position Error (APE) [20] as shown in Eq. 21, where T is the time steps and is equal to

²Due to Covid19 and the lockdown, we couldn't use the real robot to run the experiments

TABLE I: APE with noise 1 and noise 2

APE(cm)	Noise 1	Noise 2
Head	5.99	5.89
Left shoulder	3.63	3.62
Left elbow	8.97	8.91
Left wrist	21.36	21.79
Right shoulder	3.63	3.63
Right elbow	10.79	10.73
Right Wrist	22.51	22.43

126; M is the number of testing samples and is equal to 960 (30 batches with batch size 32); $y_{\text{real}}(m, t)$ and $y_{\text{generated}}(m, t)$ are the the ground truth and prediction of joint position y of sample m at time step t , respectively.

$$APE = \frac{1}{M \times T} \sum_{m=1}^M \sum_{t=1}^T |y_{\text{real}}(m, t) - y_{\text{generated}}(m, t)| \quad (21)$$

The APE results of 7 joints are shown in Table I. Firstly, you can find that the generation with noise 1 and the generation noise 2 had similar results, which certify that the random noise can make the generated gestures have a random variation to a certain extent. Then, the head APE and the shoulder APE are small while the wrist APE is large. Because the wrist joint at the end of the arm has a large movement space and other joints movements have a limited space in real contexts.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we came up with a new one-to-many *S2GGAN* model for speech-driven gesture generation, which is promising to be used on other cross-modal mapping tasks with the time series as the input and the output. In addition, the model provided a solution to the hard problem, which is how to add noise to the time series input of GAN model or VAE model. We also built-up a speech-driven gesture database with the speech audio and the spontaneous 3D gestures. *S2GGAN* model was trained and tested to generate various speech-driven gestures. Then, the generated gestures with the joint positions were transformed towards robot gestures with the joint rotation angles, which were applied on the Pepper robot. Lastly the generated gestures were qualitatively and quantitatively evaluated.

During *S2GGAN* testing part, we generated 960 speech-driven gesture sequence mapping with the speech audio. Real-world experiments will be run soon. Participants will be asked to evaluate the naturalness of the generated gestures. Furthermore, we will also explore emotional gesture generation.

REFERENCES

- [1] M. Salem, S. Kopp, I. Wachsmuth, K. Rohlfing, and F. Joublin, "Generation and evaluation of communicative robot gesture," *International Journal of Social Robotics*, vol. 4, no. 2, pp. 201–217, 2012.
- [2] C. Yu and A. Tapus, "Interactive robot learning for multimodal emotion recognition," in *International Conference on Social Robotics*. Springer, 2019, pp. 633–642.

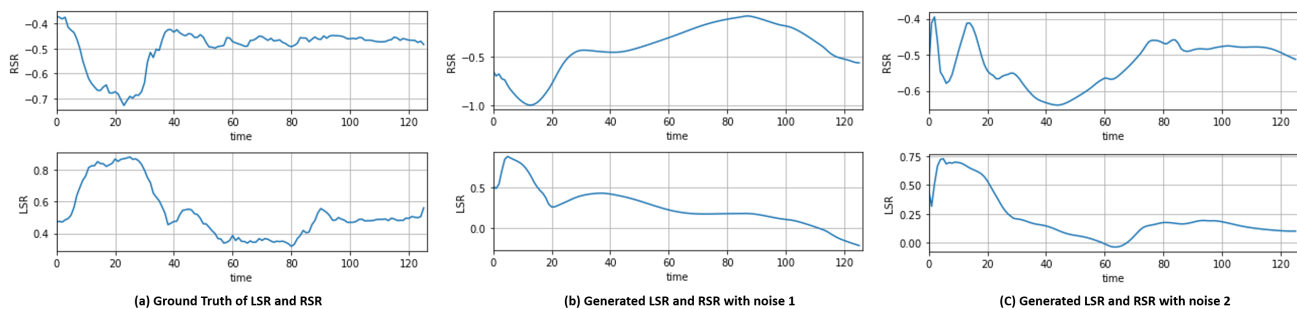


Fig. 7: The samples of the generated speech-driven gesture and the ground truth with LSR and RSR.

- [3] A. Tapus, A. Bandera, R. Vazquez-Martin, and L. V. Calderita, "Perceiving the person and their interactions with the others for social robotics—a review," *Pattern Recognition Letters*, vol. 118, pp. 3–13, 2019.
- [4] C. Yu and A. Tapus, "Multimodal emotion recognition with thermal and rgb-d cameras for human-robot interaction," in *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 532–534.
- [5] S. Rossi, F. Ferland, and A. Tapus, "User profiling and behavioral adaptation for hri: a survey," *Pattern Recognition Letters*, vol. 99, pp. 3–12, 2017.
- [6] A. Aly and A. Tapus, "Multimodal adapted robot behavior synthesis within a narrative human-robot interaction," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2986–2993.
- [7] —, "Prosody-based adaptive metaphoric head and arm gestures synthesis in human robot interaction," in *2013 16th International Conference on Advanced Robotics (ICAR)*. IEEE, 2013, pp. 1–8.
- [8] P. Jonell, T. Kucherenko, E. Ekstedt, and J. Beskow, "Learning non-verbal behavior for a social robot from youtube videos," in *ICDL-EpiRob Workshop on Naturalistic Non-Verbal and Affective Human-Robot Interactions, Oslo, Norway, August 19, 2019*, 2019.
- [9] 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.
- [10] K. Vougioukas, S. Petridis, and M. Pantic, "End-to-end speech-driven facial animation with temporal gans," *arXiv preprint arXiv:1805.09313*, 2018.
- [11] M. Ranzato, J. Susskind, V. Mnih, and G. Hinton, "On deep generative models with applications to recognition," in *CVPR 2011*. IEEE, 2011, pp. 2857–2864.
- [12] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [13] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen, "Audio-driven facial animation by joint end-to-end learning of pose and emotion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–12, 2017.
- [14] H.-Y. Lee, X. Yang, M.-Y. Liu, T.-C. Wang, Y.-D. Lu, M.-H. Yang, and J. Kautz, "Dancing to music," in *Advances in Neural Information Processing Systems*, 2019, pp. 3581–3591.
- [15] T. Tang, J. Jia, and H. Mao, "Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1598–1606.
- [16] A. Aly and A. Tapus, "Towards an intelligent system for generating an adapted verbal and nonverbal combined behavior in human-robot interaction," *Autonomous Robots*, vol. 40, no. 2, pp. 193–209, 2016.
- [17] J. Cassell, H. H. Vilhjálmsón, and T. Bickmore, "Beat: the behavior expression animation toolkit," in *Life-Like Characters*. Springer, 2004, pp. 163–185.
- [18] Y. Yoon, W.-R. Ko, M. Jang, J. Lee, J. Kim, and G. Lee, "Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4303–4309.
- [19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.
- [20] D. Hasegawa, N. Kaneko, S. Shirakawa, H. Sakuta, and K. Sumi, "Evaluation of speech-to-gesture generation using bi-directional lstm network," in *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, 2018, pp. 79–86.
- [21] T. Kucherenko, D. Hasegawa, G. E. Henter, N. Kaneko, and H. Kjellström, "Analyzing input and output representations for speech-driven gesture generation," in *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, 2019, pp. 97–104.
- [22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [23] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.
- [24] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 421–425.
- [25] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.
- [26] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
- [27] T. Kefalas, K. Vougioukas, Y. Panagakis, S. Petridis, J. Kossaiji, and M. Pantic, "Speech-driven facial animation using polynomial fusion of features," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3487–3491.
- [28] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
- [29] S. Franz, R. Nolte-Holube, and F. Wallhoff, "Nafome: Nao follows me-tracking, reproduction and simulation of human motion," *Jade University of Applied Sciences, Germany*, 2013.
- [30] E. Yavşan and A. Uçar, "Gesture imitation and recognition using kinect sensor and extreme learning machines," *Measurement*, vol. 94, pp. 852–861, 2016.
- [31] "Pepper-joints-aldebaran 2.0.6.8 documentation," (Date last accessed 20th-May-2020). [Online]. Available: http://doc.aldebaran.com/2-0/family/juliette.technical/joints_juliette.html
- [32] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7291–7299.
- [33] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *ICCV*, 2017.
- [34] 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, jul 2014.
- [35] B. Fabrice, "Ffmpeg." [Online]. Available: <https://ffmpeg.org>
- [36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.