S2M-Net: Speech Driven Three-party Conversational Motion Synthesis Networks

Aobo Jin
University of Houston - Victoria
jina@uhv.edu
USA

Qixin Deng
University of Houston
qdeng4@cougarnet.uh.edu
USA

Zhigang Deng*
University of Houston
zdeng4@central.uh.edu
USA

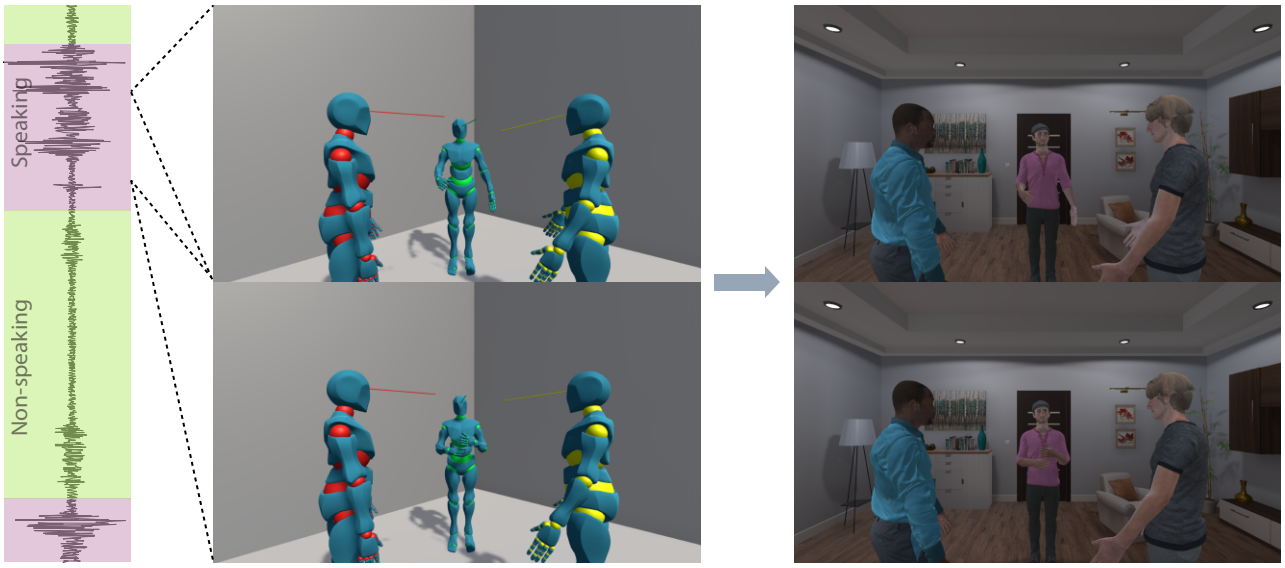


Figure 1: Two frames generated by our approach, given speech input and speaker marking (the purple colored regions in (a)). Our method first generates conversational gesture kinematics (colored with red, green and yellow respectively in (b)), and the gesture motion can be further transferred to photo-realistic virtual human models (c).

ABSTRACT

In this paper we propose a novel conditional generative adversarial network (cGAN) architecture, called **S2M-Net**, to holistically synthesize realistic three-party conversational animations based on acoustic speech input together with speaker marking (i.e., the speaking time of each interlocutor). Specifically, based on a pre-collected three-party conversational motion dataset, we design and train the S2M-Net for three-party conversational animation synthesis. In the architecture, a generator contains a LSTM encoder to encode a sequence of acoustic speech features to a latent vector that is further fed into a transform unit to transform the latent vector into a gesture kinematics space. Then, the output of this transform unit is fed into a LSTM decoder to generate corresponding three-party conversational gesture kinematics. Meanwhile, a discriminator is implemented to check whether an input sequence of three-party conversational gesture kinematics is real or fake. To evaluate our method, besides quantitative and qualitative evaluations, we also conducted paired comparison user studies to compare it with the state of the art.

CCS CONCEPTS

• **Computing methodologies** → **Animation**.

ACM Reference Format:
Aobo Jin, Qixin Deng, and Zhigang Deng. 2022. S2M-Net: Speech Driven Three-party Conversational Motion Synthesis Networks. In *ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '22), November 3–5, 2022, Guanajuato, Mexico.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3561975.3562954

1 INTRODUCTION

Multiparty conversation is one of most common human-human communication forms in our society. Due to its obvious importance and prevalence, researchers have attempted various efforts to understand the behaviors of multiparty conversations [Watzlawick

*corresponding author

MIG '22, November 3–5, 2022, Guanajuato, Mexico

ACM ISBN 978-1-4503-9888-6/22/11…$15.00
https://doi.org/10.1145/3561975.3562954

et al. 2011], in particular, the analysis of multi-party conversational gesture [de Coninck et al. 2019; Ding et al. 2017; Foster et al. 2012; Gu and Badler 2006; Johansson et al. 2013; Kondo et al. 2013; Matsuyama et al. 2010; Mutlu et al. 2009; Otsuka et al. 2005; Vertegaal et al. 2000]. Meanwhile, synthesis of multiparty conversational motion can find its potential use in various applications [Jin et al. 2022], including automated virtual conversations in entertainment, virtual crowds, human interaction with a group of robots, and teleconferencing.

Previously, researchers proposed a number of approaches to generate conversational animations, such as hand gesture, gaze animation, and other gesture modalities, using either rule-based algorithms [Cassell et al. 1994, 2001; Marsella et al. 2013] or data-driven algorithms [Alexanderson et al. 2020; Ferstl et al. 2020; Klein et al. 2019; Levine et al. 2010, 2009; Stone et al. 2004]. However, most of them are often limited in generating the gesture of a *single* talking avatar or *dyadic* conversational animation. Recently, a deep learning based approach [Jin et al. 2019] was proposed to tackle the synthesis of head-and-eye motion in three-party conversations, but it mechanically divides the problem to two independent sub-problems: a LSTM (Long Short-Term Memory) based model for synthesizing the speaker's head-and-eye motion and the second LSTM based model for synthesizing listeners' head-and-eye motion. More importantly, the mechanical task separation may not generate the natural flow of conversational gestures among interlocutors.

Inspired by the above challenge, in this paper we propose a novel holistic framework to automatically generate *conversational gesture kinematics* for three party conversations, with the input speech and its corresponding speaker marking (i.e., the speaking time of each interlocutor). Specifically, based on a pre-collected three-party conversational motion dataset (including eye motion, head motion, hand gesture, torso movement, and acoustic speech of all the interlocutors engaged in natural three-party conversations), we design a conditional Generative Adversarial Networks (cGAN) based architecture, called *S2M-Net*, for the synthesis of speech-driven three-party conversational animation. In its architecture, a generator contains a LSTM encoder to generate a sequence of acoustic speech features to a latent vector. Then, the latent vector is further fed into a transform unit to transform the latent vector into a gesture kinematics space. Finally, the output of this transform unit is fed to a LSTM decoder to generate corresponding conversational gesture kinematics. Meanwhile, a discriminator is designed to check whether an input sequence of three-party conversational gesture kinematics is real or fake. To evaluate our method, we performed both quantitative and qualitative evaluations on the results by our approach, and also compared it with a state of the art approach. In addition, we conducted a paired comparison user study to validate the realism of the synthesized animations by our approach.

In sum, our S2M-Net framework is the first cGAN-based architecture specifically designed for the automated holistic generation of three-party conversational motion given the input of acoustic speech signals. Furthermore, we introduce two new loss functions into the S2M-Net architecture for the optimization of the conversational motion generator. We also introduce LSTM units as the encoder and the decoder in the S2M-Net architecture to handle temporal input and output.

2 RELATED WORK

In this section, we briefly review the recent efforts that are most related to our work, including conversational head-and-eye animation, conversational gesture synthesis, and conditional GAN-based learning.

Conversational Head and Eye Animation. Head and eye motions are an indispensable part of animated virtual humans. Many previous works have been done for gaze synthesis [Ruhland et al. 2014]. As one of the early data-based methods, Lee et al. [2002] statistically analyze pre-recorded gaze data of a human subject during speaking/listening and then further generate novel saccadic movements using first-order statistics. Graf et al. [2002] studied the conditional probabilities of pitch accents accompanied by certain primitive head movements (e.g., nodding). Vinayagamoorthy et al. [2004] proposed a computational model to generate gazes for two avatars in a dyadic interaction in virtual environment, based on a pre-collected face-to-face gaze dataset. Busso et al. [2007; 2005] proposed a framework to synthesize expressive head motion sequences based on prosodic features of novel input audio.

In light of the importance of a large amount of data for head motion generation, Chuang and Bregler [2005] first collect a database of head motion sequences, indexed by pitch features, and then generate novel head motion sequences through dynamic programming that aims to maximize the match between the pitch features in the database and those of new inputted speech. Along the data-driven direction, Deng and colleagues [Deng et al. 2005; Le et al. 2012; Ma and Deng 2009] developed several statistical gaze and/or head-gaze models from pre-recorded conversational head/eye motion data, in particular, statistical modeling of the coupling between gaze and head movement. Recently, Jin et al. [2019] proposed a deep learning based method to synthesize gaze and head motions of the interlocutors in a three-party conversation based on speech input. However, their method mechanically separates listeners' motion from the speaker's motion, by training two independent motion synthesis models for listeners and the speaker, respectively. In addition, researchers have also explored the visual attention in virtual environment to guide the generation of plausible avatar gazes [Gu and Badler 2006; Peters and O'Sullivan 2003].

Conversational Gesture Synthesis. As one of the early works in this direction, Cassell et al. [1994] designed a rules-based system to generate hand gestures and other communicative gestures for conversational agents. Later, Cassell et al. [2001] further developed an extensible BEAT toolkit to take texts as the input and generate synchronized nonverbal conversational gesture behaviors, based on a set of rules extracted through linguistic and contextual analysis of the input texts. Recently, Marsella et al. [2013] proposed a rules-based animation method to generate virtual conversations between two parties, driven by speech as well as annotated texts as the input. Realizing the importance of data-driven schemes, Stone et al. [2004] proposed a data-driven method to generate an animated speaking character through the optimal re-combination of existing samples. Later, Levine et al. [2010; 2009] proposed two data-driven animation methods to synthesize conversational gesture kinematics of a talking avatar. Both of them utilize prosody features of speech to animate body language, which can be extended for two-party

conversation applications. Several gesture synthesis methods which use either audio or text as input are proposed [Ahuja et al. 2019; Ginosar et al. 2019; Kucherenko et al. 2020; Liang et al. 2022; Liu et al. 2022; Yang et al. 2020; Yoon et al. 2020, 2019] thanks to the increasing of relative dataset and the deep learning architectures. However, none of those work can effectively synthesis the conversation with three or more persons.

Conditional GANs (cGANs). cGANs have been previously exploited for discrete labels [Denton et al. 2015; Gauthier 2014; Mirza and Osindero 2014], texts [Reed et al. 2016b], and images [Isola et al. 2017]. cGAN models have been employed for image prediction from a normal map [Wang and Gupta 2016], future frame prediction [Mathieu et al. 2015], product quality photo generation [Yoo et al. 2016], image generation from sparse annotations [Karacan et al. 2016; Reed et al. 2016a], image generation from sketches [Isola et al. 2017], etc. However, all these methods only consider the image input using convolutional layers, which cannot be directly utilized for a sequence input such as continuous acoustic speech. Recently, Yu and Canales [2019] proposed a conditional LSTM-GAN for melody generation from lyrics, by taking sequences of noisy vectors conditioned on sequence of syllable-embedding vectors to generate melody. In contrast, our cGAN-based architecture not only handles sequence input/output but also handles the nontrivial mapping problem between speech and three-party conversational gesture kinematics, which cannot be achieved by the existing LSTM-GAN framework [Yu and Canales 2019].

3 OUR APPROACH OVERVIEW

Speech-driven three-party conversational gesture generation can be formulated as a mapping from speech to gesture. Therefore, the main task of our work is to build the mapping from the acoustic speech space to the three-party conversational gesture kinematics space. Note that we primarily focus on three-party conversations without overlapping speech, that is, there is one speaker and two listeners at any moment. Formally, given a set of paired sequence of speech features $X = (x^1, \cdots, x^{|X|})$ and speaker marking $M = (m^1, \cdots, m^{|M|})$ with corresponding three-party conversational gesture kinematics $Y = (y^1, \cdots, y^{|Y|})$, our goal is to establish a mapping $T : (X|M) \rightarrow (Y|M)$. In this work, we treat it as a cross-modal translation problem.

Our method utilizes a pre-recorded three-party conversational motion dataset, which includes the simultaneously recorded conversational gestures (including eye motion, head motion, hand gestures, and body movement) and accompanying acoustic speech of all the interlocutors engaged in natural three-party conversations. From the data, we first extract acoustic speech features paired with gesture kinematics. We also manually annotate the speaker marking information for the three interlocutors. Then, we train the S2M-Net 2 (a cGAN-based architecture network, described in Section 5) that consists of a LSTM encoder, a transform unit, a LSTM decoder in generator, and a LSTM architecture in discriminator. The input for the generator is acoustic speech features conditioned on the speaker marking, the input for the discriminator is conversational gesture kinematics conditioned on the speaker marking. At the second step, since our architecture can only generate fixed-length gesture kinematics, we then concatenate the generated gesture kinematics together to form the final motions of the three interlocutors in a conversation. Figure 2 illustrates the pipeline of our approach. Note that the speaker marking information is user-provided beforehand and is used as one of the inputs to our approach. The root joints of the three interlocutors are fixed during training and synthesis. We fix a relatively short input length to ensure less information loss due to the structural bottleneck of the encoder and decoder.

4 DATA PREPROCESSING

The three-party conversation motion dataset used in this work contains audio and motion of all three interlocutors (e.g., joint angles, 3D head movements, and gaze motion (yaw and pitch angles)). The dataset contains 10 three-party conversation sessions: 5 of the sessions for three males and others for three females. Each of the session lasts from 8 to 10 minutes. The total number of frames in the dataset is about 144k after the dataset was re-sampled to 30 frames per second. All the angles are represented as Euler angles. A ten-cameras VICON optical motion capture system was used to record the joint angles of each interlocutor, and the close-up facial video of the three interlocutors were captured by three Canon HD cameras, respectively. The gaze angles at each frame were extracted using two existing methods [Le et al. 2012; Wang et al. 2016] based on the recorded face video of each interlocutor.

We also down-sampled the recorded audio to 30Hz. The fundamental frequency (F0) and the first 32 dimensions of Mel-frequency cepstral coefficients (MFCCs) were extracted from audio as the acoustic speech features for every frame. The speaker marking information was represented as a three-dimensional one-hot vector for the three interlocutors in one frame, where 1 indicates speaking at this frame and 0 indicates non-speaking (i.e., listening). To this end, we define the gesture kinematics in our work as a 44-dimensional vector per frame for each interlocutor including gaze (i.e., yaw and pitch angles), head movement (i.e., yaw, roll, and pitch angles), neck joint, spine joints (3 spine joint per interlocutor), hip joint, left/right shoulder joints, left/right upper arm joints, left/right forearm joints, and left/right hands. All the joint angles were normalized to the range from 0 to 180 degree to eliminate the discontinued angle situation. The three 44-dimensional vectors (of the three interlocutors) were concatenated to form a 132-dimensional vector (called a *gesture kinematics vector* in this work) for each frame, in concert with the speaker marking (e.g., the first interlocutor expressed in the speaker marking corresponds to the first 44 elements in the gesture kinematics vector, and so on). Each element in the gesture kinematic vector is normalized to the range from 0 to 1. To preserve the turn taking/keeping flow in natural three-party conversations, we intentionally extracted the above gesture kinematics vectors with time overlapping, as illustrated in Figure 3. In this work, each sample has 120 frames. That is, each sample has 120×33 dimensions for speech features, 120×3 dimensions of speaker marking, and 120×132 dimensions for gesture kinematics. In total, we have 28,689 samples. And, we split them to a training set (90%) and a test set (10%).

5 NETWORK ARCHITECTURE

In this section, we describe our **S2M-Net** architecture to generate three-party conversational gesture kinematics based on novel

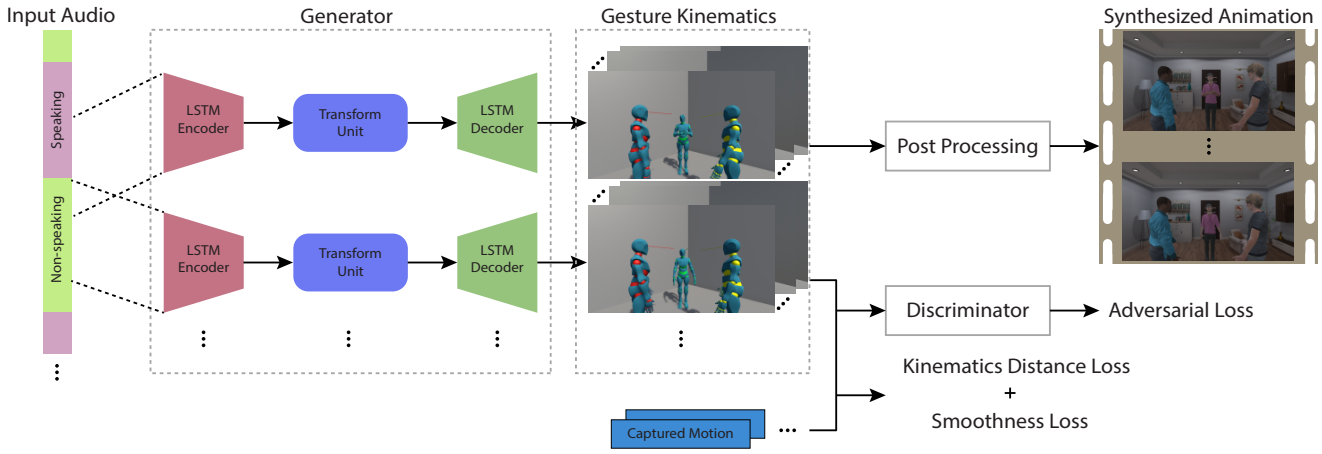


Figure 2: Pipeline illustration of our approach, which consists of a generator to generate three-party gesture kinematics given acoustic speech input, a discriminator with the adversarial loss combined with two new loss functions introduced by us to optimize the generator, and a post-processing step to generate three-party conversational animations.

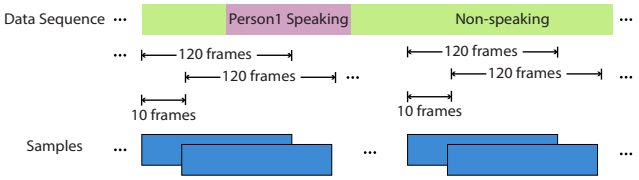


Figure 3: Illustration of data sample generation in this work. Each sample contains 120 frames with 110 frames overlapped with the previous/next samples.

speech input, as illustrated in Figure 2. The conditional generator **G** in our model contains a LSTM encoder, a transform unit, and a LSTM decoder. Our conditional discriminator **D** is designed to ensure **G** can be optimized effectively. We utilize the cGAN architecture [Mirza and Osindero 2014], instead of the traditional GAN architecture [Goodfellow et al. 2014a], to emphasize the role of the speaker marking information during model training.

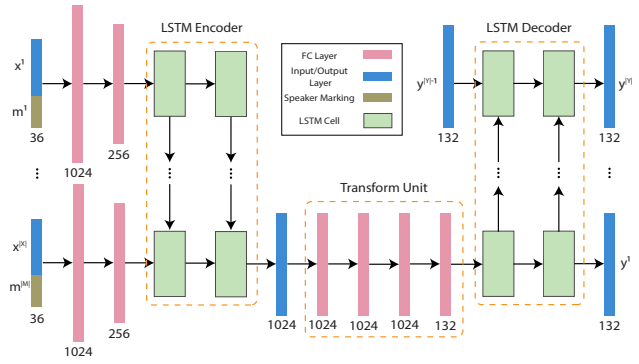


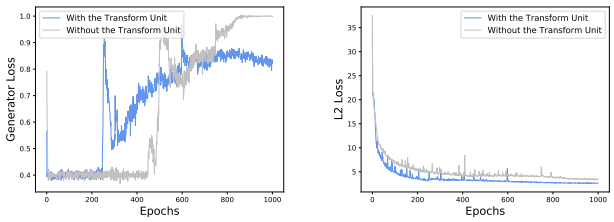
Figure 4: The architecture of the generator G. With the speech features $X = (x^1, \cdots, x^{|X|})$ and the speaker marking $M = (m^1, \cdots, m^{|M|})$, the generator can output gesture kinematics $Y = (y^1, \cdots, y^{|Y|})$ through the LSTM encoder, the transform unit, and the LSTM decoder.

5.1 Conditional Generator

As illustrated in Fig. 4, our conditional generator first takes a sequence of speech features input X with its speaker marking M and encodes it into a latent vector $Z_{(X|M)}$. Then, through the proposed transform unit, we transform the latent vector from the speech features space to the gesture kinematics space $Z_{(Y|M)}$. Finally, the output of the transform unit is fed into the decoder to generate gesture kinematics Y. We describe each module below.

LSTM Encoder. Unlike the previous work [Yu and Canales 2019], we first utilize 2 fully connected (FC) layers to combine input speech features with the speaker marking. Each FC layer followed by batch normalization, leaky ReLU activation, and dropout layer. The first layer has 1024 hidden units, and the output layer has 256 hidden units (refer to Figure 4). Since the dimensions of speech features dominate those of speaker marking (i.e., 33 dimensions vs 3 dimensions in one frame), the FC layers can easily extract the useful information from the two inputs and weaken the domination effect by projecting those inputs to a high-dimensional space. The output sequence of the FC layers (256 dimensions in one frame) is fed into a 2-layers LSTM unit to encode the input sequence to a 1024-dimensional latent vector $Z_{(X|M)}$ (the blue bar connected after the LSTM Encoder). We also record the cell state $c_{|X|}$ (256 dimensions). Each layer of the LSTM contains 1024 hidden units.

Transform Unit. The transform unit can transform the latent vector $Z_{(X|M)}$ (1024 dimensions) to $Z_{(Y|M)}$ (132 dimensions). As illustrated in Fig. 4, we utilize a 4-layers FC layer to handle the transform task since both the input and the output of the transform unit contain vectors with small lengths, which can easily feed into the FC layer. Each FC layer is followed by batch normalization, leaky ReLU activation, and dropout layer. The first three FC layers have 1024 hidden units, and the last FC layer has 132 hidden units. The cell state of the LSTM encoder is also transformed to 132 dimensions through the transform unit.



(a) (b)

Figure 5: (a): The generator loss between our architecture with the transform unit and without the transform unit. (b): The L2 kinematics distance loss of our architecture with the transform unit and without the transform unit.

The transform unit can effectively facilitate the generative capability of the generator since this unit builds the relation between two non-linear feature spaces (i.e., the speech feature space, and the conversational gesture kinematics space). Fig. 5 shows the effect of the transform unit. The generator can generate good gesture kinematics to fool the discriminator even in late epochs with the transform unit (Fig. 5(a)). Since the discriminator can become stronger during training, which will increase the generator loss, but a lower generator loss still reflects the stronger capability of generating samples to fool the discriminator. Also, the L2 distance loss shows that the generated gesture kinematics are better using the architecture with the transform unit than the one without it, as shown in Fig. 5(b).

LSTM Decoder. To decode the latent vector outputted from the transform unit to a sequence of gesture kinematics, we utilize a 2-layers LSTM unit. Each layer of the LSTM contains 132 hidden units. The first frame of the gesture kinematics is generated by the output of the transform unit (i.e., the transformed latent vector and the transformed cell state), and the following frames of the gesture kinematics are generated by the previously generated frame of gesture kinematics and the previous cell state of one LSTM cell in the LSTM unit. Since our training samples contain gesture kinematics with the pre-defined length (i.e., 120 frames per sample), the decoder will not stop until the generated gesture kinematics have the pre-defined length.

5.2 Conditional Discriminator

The conditional discriminator is used to check whether the input sequence is real gesture kinematics Y or fake gesture kinematics $\hat{Y}$ under the speaker marking condition M. Since the input is a sequence, we utilize a LSTM unit combined with a FC layer to handle this classification problem. The architecture is illustrated in Fig. 6. Specifically, we first concatenate the input gesture kinematics with the speaker marking, frame by frame, into a sequence of 135-dimensional vectors. Then, it is fed into the 1-layer LSTM unit to generate a 512-dimensional vector which is further fed into the FC layer followed by the dropout layer. The final output is a scalar value.

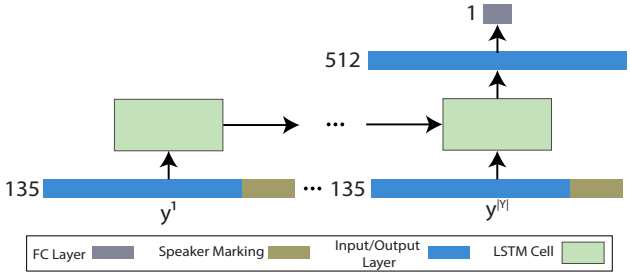


Figure 6: The architecture of the discriminator D. With gesture kinematics $Y = (y^1, \cdots, y^{|Y|})$ as the input, the discriminator checks whether the input sequence is real or fake to optimize the generator through back-propagation.

6 OBJECTIVE FUNCTIONS

To learn the transform functions from the speech features space to the gesture kinematics space (i.e., optimize the generator), we introduce three loss functions: (i) adversarial loss, (ii) L2 distance loss between the generated gesture kinematics and the ground truth, and (iii) smoothness loss. During training, we utilize the back-propagation algorithm on these loss functions to update the weights of our network.

Adversarial Loss. The objective of our network can be expressed as:

$$\begin{aligned}\mathcal{L}_{cGAN}(\mathbf{G},\mathbf{D}) &= \mathbb{E}_{M,Y}[\log \mathbf{D}(M,Y)] \\ &+\mathbb{E}_{M,X}[\log(1-\mathbf{D}(M,\mathbf{G}(M,X))],\end{aligned} \tag{1}$$

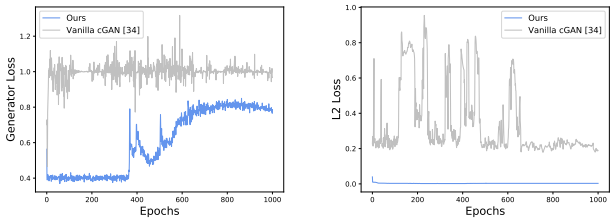
where $\mathbf{G}$ aims to minimize this objective against the adversarial $\mathbf{D}$ that attempts to maximize it, i.e., $\mathbf{G}^* = \arg\min_{\mathbf{G}} \max_{\mathbf{D}} \mathcal{L}_{cGAN}$. The adversarial loss ensures that the output distribution of the conditional generator matches the target distribution of the gesture kinematics Y [Goodfellow et al. 2014b].

Kinematics Distance Loss. Inspired by the previous works [Isola et al. 2017; Pathak et al. 2016] that show the benefit of mixing the GAN objective with a more traditional loss, in order to improve the generation results, we add an L2 distance loss to push the generated results towards the ground truth in the L2 sense using the following function:

$$\begin{aligned}\mathcal{L}_k(\mathbf{G}) &= \mathbb{E}_{M,X,Y}[||w \odot (Y - \mathbf{G}(M,X))||_2], \\ w_i &= \begin{cases} \alpha & \text{if } i \in \mathbb{N}_A \text{ and } M_p = 1, \text{where } p = \lfloor \frac{i}{44} \rfloor \\ 1 & \text{otherwise} \end{cases}\end{aligned} \tag{2}$$

where $\odot$ denotes the element-wise multiplication between vectors and $w_i \in w$, α is a constant, $\mathbb{N}_A$ denotes the positions of the forearms, upper arms and hands elements in one gesture kinematics vector, and M_p is the speaker marking for the interlocutor p. Since the ranges of the joint angles of the forearms, upper arms, and hands are typically significantly larger than the ranges of other joint angles for the speaker during conversations, we introduce w to give larger penalties to the joint angles of the forearms, upper arms, and hands of the speaker.

Smoothness Loss. Since the velocities of joint angles are important parameters of gesture kinematics, we add an L1 distance loss to evaluate the velocity difference between the output of $\mathbf{G}$ and the



(a) (b)

Figure 7: (a): The comparison of the generator loss between our architecture and the vanilla cGAN [Mirza and Osindero 2014]. (b): Th comparison of the L2 RL between our architecture and the vanilla cGAN [Mirza and Osindero 2014].

ground truth as follows:

$$\mathcal{L}_s(\mathbf{G}) = \mathbb{E}_{M,X,Y}[||ws \odot (\mathrm{d}Y - \mathrm{d}\mathbf{G}(M,X))||_1],$$
$$ws_i = \begin{cases} \alpha_s & \text{if } i \in \mathbb{N}_A \text{ and } M_p = 1, \text{where } p = \lfloor \frac{i}{44} \rfloor \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

where $\odot$ denotes the element-wise multiplication between vectors and $ws_i \in ws$, $\mathbb{N}_A$ denotes the positions of the forearms, upper arms, and hands elements in one gesture kinematics vector, and M_p is the speaker marking for the interlocutor p. This loss function can effectively reduce the stiff motions and loop motions (e.g., arms regularly move up and down in an approximately constant speed).

The **full objective function** can be summarized as follows:

$$\mathbf{G}^* = \arg \min_{\mathbf{G}} \max_{\mathbf{D}} \mathcal{L}_{cGAN}(\mathbf{G},\mathbf{D}) + \lambda_k \mathcal{L}_k(\mathbf{G}) + \lambda_s \mathcal{L}_s(\mathbf{G}). \quad (4)$$

Fig. 7 shows the comparison of the generator loss curves and the RL curves between our proposed loss functions and the vanilla cGAN [Mirza and Osindero 2014] that only has the adversarial loss. As shown in this figure, our loss functions can lead to significantly smaller losses than the vanilla cGAN, in particular, the L2 loss. A lower L2 loss represents a better reconstruction of gesture kinematics. The generator loss increases after several epochs, since the discriminator is trained well and it will be more and more difficult for the generator to generate samples to fool the discriminator. Since our S2M-Net has a more powerful generator that can better fool the discriminator, the generator loss by our S2M-Net is smaller than that of the vanilla cGAN.

7 TRAINING DETAILS

In our experiments, we trained our S2M-Net architecture by following the techniques proposed by Zhu et al. [2017]. We applied the Adam solver and the momentum parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$. For first 40 epochs, we kept the learning rate at 0.0005 for both **G** and **D**. After that, we linearly decayed the learning rate to 0 over the remaining epochs. We trained it with 1000 epochs. The batch size was set to 128, and each batch was randomly selected from the training dataset. We set $\lambda_k = 300$ in Equation 4 and $\lambda_s = 1$ in Equation 4. The keep probability for dropout layers was 0.2, and the β was set to 0.2 in all leaky ReLU activations. We also set $\alpha = 5$ and $\alpha_s = 10$. All the hyper-parameter values were empirically chosen via our experiments. The training of our architecture took about 19 hours on an off-the-shelf computer with an Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, a NVIDIA GeForce RTX 2080TI GPU, and 32GB Memory size (RAM). During tests, we did not apply batch normalization and dropout layers, which would lower the performance during inference.

8 POST-PROCESSING

Since the direct output of our S2M-Net architecture are gesture kinematics sequences with a pre-defined length (i.e., 120 frames), we also need to design a post-processing method to concatenate the outputted gesture kinematics sequences together to form a much longer, continuous gesture kinematics sequence.

The details of our gesture kinematics concatenation algorithm for one interlocutor is shown in Algorithm 1. Specifically, instead of cutting the input speech features to parts with the pre-defined length to generate gesture kinematics, we extract the speech features with an overlap of 30 frames. As shown in Fig. 8, the larger the overlap frame length, the smaller the average MSE value at the concatenation position. However, since a larger overlap frame length would generate artifacts such as possible elimination of some short sequences of hand gesture, we experimentally choose the overlap length of 30 frames. Note that the sampling rate of our motion data is also 30 frame per second.

In this work, we experimentally set the overlap frame length to 30. Thus, the directly generated gesture kinematics also have an overlap of 30 frames. We first split the whole 120 frames × 132-dimensional gesture kinematics sequence to three 120 frames × 44-dimensional gesture kinematics sequences, each of which represents the motion of one interlocutor. For each interlocutor, we iterate the overlapped 30 frames of two consecutive gesture kinematics sequences (called the *preceding/succeeding* sequences) and identify one particular frame that has the minimal mean squared error (MSE) between the two (line 3 ~ line 8 in Algorithm 1). Based on the identified MSE, we select a length $L = 30$ - *offset* to do an interpolation between the $(120-L)$-*th* frame of the preceding gesture kinematics sequence and the first frame of the succeeding gesture kinematics sequence (line 12). In this way, the two sequences can be blended to form a new gesture kinematics sequence with 210 frames. For each interlocutor, we repeat the same process to concatenate more sequences. Finally, we apply the Savitzky-Golay filter on the concatenated gesture kinematics to further smooth the motion.

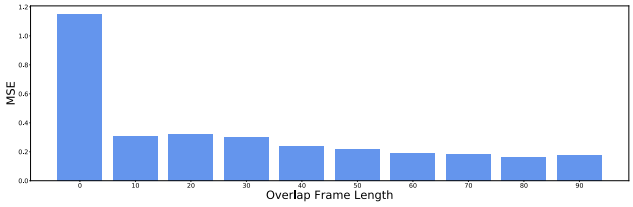


Figure 8: The average MSE values of gesture kinematics at the concatenation positions for three interlocutors (Y axis) with the increase of the overlap frame length (X axis).

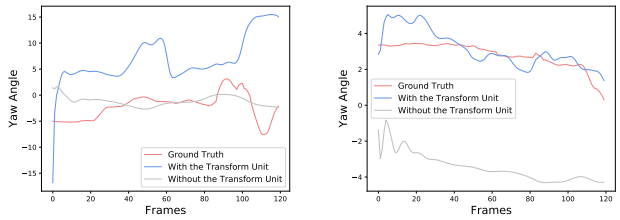
9 RESULTS AND EVALUATIONS

We used our approach to generate many conversational animations based on audio input. In particular, the text content of test speech clips was not included in our training data. For animation results, please refer to the enclosed demo video.

Algorithm 1: Concatenation Algorithm

Input : Gesture kinematics sequences $seq1$ and $seq2$
Output: The concatenated gesture kinematics sequence seq

Function `Frame_Concat(seq1, seq2)`:
  while $i < 30$ **do**
    $dis \longleftarrow MSE(seq1[|seq1| - 30 + i], seq2[i])$;
    if $dis < minDis$ **then**
      $minDis \longleftarrow dis$;
      $minFrame \longleftarrow i$;
  $offset \longleftarrow \max(minFrame - \frac{minDis}{\varepsilon}, 0)$;
  $conPoint \longleftarrow |seq1| - 30 + offset$;
  $firstPart \longleftarrow seq1[0 : conPoint]$;
  $midPart \longleftarrow interpolate(seq1[conPoint], seq2[minFrame])$;
  $lastPart \longleftarrow seq2[minFrame : |seq2|]$;
  $seq \longleftarrow concatenate(firstPart, midPart, lastPart)$;
  return seq



(a) (b)

Figure 9: Comparison of the yaw angle trajectories for both the eyes (a) and the head (b) of a randomly selected interlocutor, among our architecture with the transform unit (blue curves), our architecture without the transform unit (gray curves), and the original captured motion (red curves).

9.1 Ablation Study

We conducted an ablation study to evaluate algorithm components (transform unit, and design of loss functions) of our approach by comparing them with alternative approaches.

Transform Unit. To evaluate the effectiveness of the transform unit in our architecture, we compared the generated yaw angle trajectories of both the eyes and the head of a randomly selected interlocutor, among with/without the transform unit and the ground-truth (i.e., the original captured motion), as shown in Fig. 9. For a fair comparison, we used the same training data as well as the same parameters for training. We randomly selected a conversational audio clip from the retained test set and fed its speech features to the two architectures (i.e., with/without the proposed transform unit) to generate corresponding gesture kinematics. We chose the yaw angle for comparison, since the yaw angles of both the head and the eyes are most related to turn management in multiparty conversations. Since our architecture is a *generative* model, the generated result may not be highly similar to the ground truth (red curves). However, our architecture with the transform unit (blue curves) can learn the reasonable trend of gesture kinematics than our architecture without the transform unit (gray curves) for the yaw angle. The results obtained by our architecture without the transform unit (gray curves) show that it barely learns the distribution of the conversational gesture kinematics from the same training data. Note that the blue curves have a similar trend with the red curves, while the gray curves do not have a trend on the other hand.

Loss Functions. We compared our proposed loss functions in Eq. 4 with two other popular designs of the loss functions: (i) the cGAN [Isola et al. 2017] with reconstruction loss (RL) functions (i.e., not having w and ws parameters in Eq. 2 and Eq. 3), and (ii) the vanilla cGAN [Mirza and Osindero 2014]. For a fair comparison, we used the same architecture and the same parameters, except using different loss functions for training. Fig. 10 shows the comparison of the same frames among the three methods based on the same conversational audio input. The results based on our loss functions show more realistic conversational gesture animations than the other two designs. Note that the vanilla cGAN generated meaningless gesture kinematics at some frames, since its discriminator cannot robustly judge an input gesture kinematics sequence is real or fake, and its generator is barely optimized during training with only the adversarial loss. Meanwhile, cGAN with RL functions simultaneously generated gestures for three interlocutors, but the results still fall short of realistic three-party conversational gestures. Animation comparison can be found in the demo video.

9.2 Comparison User Study

We conducted a paired comparison user study to evaluate the realism of the generated conversational gesture animations by our approach. Specifically, we compared our approach with the approach by Jin et al. [2019] in a paired way, respectively. We chose the work of [Jin et al. 2019] for paired comparison, because it is the recent, most related approach for three-party conversational motion synthesis. Besides the generation of head-and-eye motion, it also integrates state-of-art body and hand gesture generation methods for three-party conversational motion. Note that even though some recent works (e.g., [Ginosar et al. 2019]) were proposed to generate conversational gesture, they were primarily focused on the gesture generation on a single avatar, which cannot be straightforwardly extended for three-party conversational motion synthesis.

Similar to the previous study in [Jin et al. 2019], we randomly selected 5 test clips from the retained test set; each lasts about 10 to 20 seconds. Then, we animated three virtual interlocutors to generate three-party conversational animations using two different approaches, respectively: our approach and [Jin et al. 2019]. Note that although the method in [Jin et al. 2019] is focused on the generation of head-and-eye motion for three-party conversations, it also mentions solutions (i.e., borrow or extend previous works) to generate other aspects of three-party conversational animations, including lip-sync, hand/body gesture of the speaker, and hand/body gesture of listeners. For a fair comparison, we used the same lip-sync method in both cases. In this way, we constructed 5 pairs of stimuli for comparing our approach to [Jin et al. 2019]. To avoid the potential bias, we randomized both the displayed order of these stimuli pairs and the left/right positions of the two clips in each

Figure 10: The direct comparison of our proposed loss functions (ours) with the cGAN with RL functions [Isola et al. 2017] and the vanilla cGAN [Mirza and Osindero 2014] at two frames (a) and (b). The green triangles point to the speaker at a specific frame.

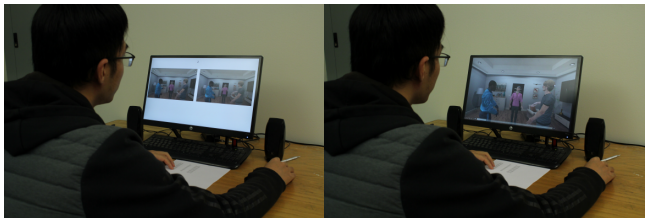


Figure 11: Experiment setup of our paired comparison user study. (left) Two animation videos (with audio) are displayed side by side for the participants to make perceptual judgment. (right) During the study, the participants can choose to zoom any video to full screen to ensure any detailed motion (e.g., eye motion) can be observed clearly.

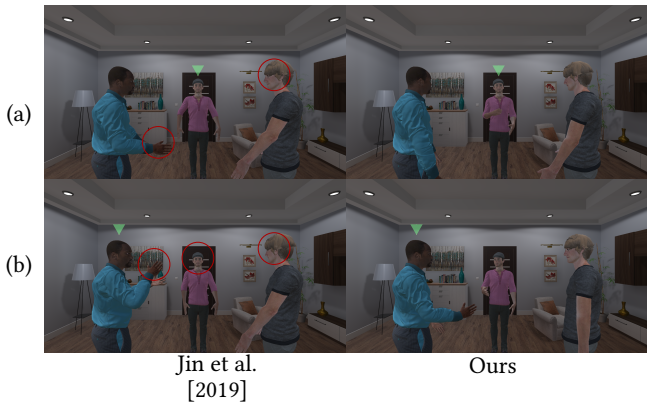


Figure 12: Direct comparisons of two randomly selected frames between the state of the art method [Jin et al. 2019] and our approach based on the same conversational audio input. The green triangles point to the speaker at a specific frame. The red circles mark the wrong gesture that usually should not happen in a real-world three-party conversation.

pair for each participant. Fig. 12 shows the direct comparison of two selected frames between the two approaches. Note that the rendered virtual characters have the same skeleton topology as the three-party conversation motion dataset used in our work.

The same 3D scene, characters, and video resolution (1920 × 1080 pixels) with the same audio for one pair of stimuli were used. Participants were instructed to sit on a chair in a controlled environment, with approximately a 0.6m distance away from a LCD monitor with 1920 × 1080 resolution. Two high quality speakers were used to play the audio. We recruited a total of 16 student volunteers in a university campus (14 males, 2 females; from 22 to 36 years old) to participate in our study. All of them did not have difficulty to understand English-language television or film without subtitles. Fig 11 shows the experiment setup used in our user study. The participants can watch a pair many times as they want before they make a choice, and they can also optionally select any clip from a pair and play it in full screen (refer to the right of Fig. 11). After watching each pair, the participants were asked to select the clip in which three-party conversational gesture motion appears perceptually more natural for them. They also can choose the *undecided* option if they had difficulty to tell which one appears more natural. To make a fair comparison, at the beginning of our user study, we instructed participants to make their perceptual judgements by focusing on the whole three-party conversational gesture, instead of the gesture of any individual interlocutor.

Fig. 13 shows the obtained user voting result. Based on the result, we can see that our approach can generate more realistic three-party conversational animations than [Jin et al. 2019]. The possible main reason is that, our approach considers the three interlocutors as a

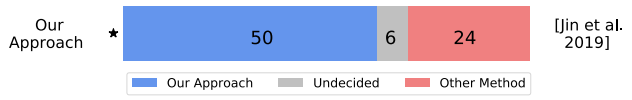


Figure 13: The obtained voting result in our user study. The number shown in each color block is the total count of votes received by one specific approach. The result marked with $*$ denotes p-value < 0.05 for the comparison in the row through a two-tailed independent paired t-test.

group and model their gesture kinematics in a holistic framework, instead of modeling their gesture motions independently as in [Jin et al. 2019]. Therefore, our approach can soundly capture the potential correlations between their conversational gestures.

9.3 Runtime Statistics

In all of our experiments, we ran our system on an off-the-shelf computer with an Inte(R) Core(TM) i7-8700K CPU @3.70GHz, 32GB Memory and NVIDIA Geforce RTX 2080TI GPU. The runtime statistics of several of our experiments were reported in Table 1, including the key steps of our approach: the generation step (about constant time) and the post-processing step (approximately linear to the length of the test audio). In sum, the training of our model takes a few hours on an off-the-shelf computer (reported in §7), but its runtime efficiency is high after the model is offline trained.

Table 1: Runtime statistics of our approach, including the time for the generation step, the time for the post-processing step, and the total computational time.

Test Audio	Audio Len. (second)	Generation (second)	Post-processing (second)	Total (second)
No. 1	97.1	0.001	0.150	0.151
No. 2	145.1	0.001	0.234	0.235
No. 3	214.2	0.001	0.354	0.355

10 DISCUSSION AND CONCLUSION

In this paper, we present a cGAN-based architecture, S2M-Net, to holistically generate realistic gesture kinematics for three-party conversations from acoustic speech input. Our work essentially trains a deep learning network to generate three-party conversational animations based on the audio input alone. To optimize the generator, we introduce two new loss functions as well as a transform unit to guide effective motion generation. Even though our approach does not support the overlapping of multiple speakers at the same time, our algorithm can generate realistic three-party kinematics, which cannot be handled by existing approaches. Our approach could be potentially extended to generate the animations of multi-party conversations involved with more than three parties.

Limitations. Our current work has the following limitations. First, since our approach does not consider semantic and affective aspects of the conversation, the generated gesture kinematics may not be perfectly in concert with the conversation content for some cases. Second, limited to the size and variety of the training data, the generated gesture kinematics by our approach may not have a sufficient variety. A larger dataset with a high variety would help to alleviate this issue. Third, since our training data do not contain overlapping speech (i.e., two or more interlocutors speak at the same time), our approach may not be able to robustly handle the input with overlapping speech (an example result is shown in Fig. 14). Finally, finger motion is often necessary to convey certain subtle communication in multiparty conversations. However, due to the lacking of detailed finger motion in our training data, our approach cannot synthesize such motion.

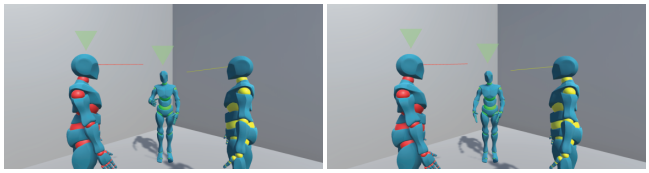


Figure 14: An example result by our approach given an input of overlapping speech. The two green triangles point to the two speakers in the input overlapping speech. As shown in the two frames, our approach can only generate the hand gesture for the green avatar (center), but not for the red avatar (left).

Future work. We would like to explore techniques to utilize various semantic features of conversational content and add detailed finger gesture for the generation of realistic multiparty conversational animation. Also, we plan to make the current pipeline editable, where users can efficiently add/delete conversational gesture segments to refine the animation. We are also interested in extending the current framework for the generation of more general forms of multiparty conversational animations. Finally, real-time generation of continuous conversational gesture given live audio input will be another future challenge to explore.

ACKNOWLEDGMENTS

This work is in part supported by US NSF IIS-2005430.

REFERENCES

Chaitanya Ahuja, Shugao Ma, Louis-Philippe Morency, and Yaser Sheikh. 2019. To React or Not to React: End-to-End Visual Pose Forecasting for Personalized Avatar during Dyadic Conversations. In *2019 International Conference on Multimodal Interaction* (Suzhou, China) *(ICMI '19)*. Association for Computing Machinery, New York, NY, USA, 74–84. https://doi.org/10.1145/3340555.3353725

Simon Alexanderson, Gustav Eje Henter, Taras Kucherenko, and Jonas Beskow. 2020. Style-Controllable Speech-Driven Gesture Synthesis Using Normalising Flows. *Computer Graphics Forum* 39, 2 (2020), 487–496.

Carlos Busso, Zhigang Deng, Michael Grimm, Ulrich Neumann, and Shrikanth Narayanan. 2007. Rigid head motion in expressive speech animation: Analysis and synthesis. *IEEE transactions on audio, speech, and language processing* 15, 3 (2007), 1075–1086.

Carlos Busso, Zhigang Deng, Ulrich Neumann, and Shrikanth Narayanan. 2005. Natural head motion synthesis driven by acoustic prosodic features. *Computer Animation and Virtual Worlds* 16, 3-4 (2005), 283–290.

Justine Cassell, Catherine Pelachaud, Norman Badler, Mark Steedman, Brett Achorn, Tripp Becket, Brett Douville, Scott Prevost, and Matthew Stone. 1994. Animated conversation: rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, 413–420.

Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. 2001. Beat: the behavior expression animation toolkit. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 477–486.

Erika Chuang and Christoph Bregler. 2005. Mood swings: expressive speech animation. *ACM Transactions on Graphics (TOG)* 24, 2 (2005), 331–347.

F. de Coninck, Z. Yumak, G. Sandino, and R. Veltkamp. 2019. Non-Verbal Behavior Generation for Virtual Characters in Group Conversations. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. 41–418.

Zhigang Deng, John P Lewis, and Ulrich Neumann. 2005. Automated eye motion using texture synthesis. *IEEE Computer Graphics and Applications* 25, 2 (2005), 24–30.

Emily L Denton, Soumith Chintala, Rob Fergus, et al. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*. 1486–1494.

Yu Ding, Yuting Zhang, Meihua Xiao, and zhigang Deng. 2017. A Multifaceted Study on Eye Contact based Speaker Identification in Three-party Conversations. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 3011–3021.

Ylva Ferstl, Michael Neff, and Rachel McDonnell. 2020. Adversarial gesture generation with realistic gesture phasing. *Computers & Graphics* 89 (2020), 117 – 130.

Mary Ellen Foster, Andre Gaschler, Manuel Giuliani, Amy Isard, Maria Pateraki, and Ronald Petrick. 2012. Two people walk into a bar: Dynamic multi-party social interaction with a robot agent. In *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM, 3–10.

Jon Gauthier. 2014. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester* 2014, 5 (2014), 2.

S. Ginosar, A. Bar, G. Kohavi, C. Chan, A. Owens, and J. Malik. 2019. Learning Individual Styles of Conversational Gesture. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014a. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014b. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2672–2680.

Hans Peter Graf, Eric Cosatto, Volker Strom, and Fu Jie Huang. 2002. Visual prosody: Facial movements accompanying speech. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE, 396–401.

Erdan Gu and Norman Badler. 2006. Visual attention and eye gaze during multiparty conversations with distractions. In *Intelligent Virtual Agents*. Springer, 193–204.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

Aobo Jin, Qixin Deng, and Zhigang Deng. 2022. A Live Speech Driven Avatar-mediated Three-party Telepresence System: Design and Evaluation. *PRESENCE: Virtual and Augmented Reality* (06 2022), 1–43. https://doi.org/10.1162/pres_a_00358 arXiv:https://direct.mit.edu/pvar/article-pdf/doi/10.1162/pres_a_00358/2031159/pres_a_00358.pdf

Aobo Jin, Qixin Deng, Yuting Zhang, and Zhigang Deng. 2019. A Deep Learning-Based Model for Head and Eye Motion Generation in Three-party Conversations. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 9 (July 2019), 19 pages.

Martin Johansson, Gabriel Skantze, and Joakim Gustafson. 2013. Head pose patterns in multiparty human-robot team-building interactions. In *International Conference on Social Robotics*. Springer, 351–360.

Levent Karacan, Zeynep Akata, Aykut Erdem, and Erkut Erdem. 2016. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215* (2016).

Alex Klein, Zerrin Yumak, Arjen Beij, and A. Frank van der Stappen. 2019. Data-Driven Gaze Animation Using Recurrent Neural Networks. In *Motion, Interaction and Games* (Newcastle upon Tyne, United Kingdom) *(MIG '19)*. Article 4, 11 pages.

Yutaka Kondo, Kentaro Takemura, Jun Takamatsu, and Tsukasa Ogasawara. 2013. A gesture-centric android system for multi-party human-robot interaction. *Journal of Human-Robot Interaction* 2, 1 (2013), 133–151.

Taras Kucherenko, Patrik Jonell, Sanne van Waveren, Gustav Eje Henter, Simon Alexanderson, Iolanda Leite, and Hedvig Kjellström. 2020. Gesticulator: A framework for semantically-aware speech-driven gesture generation. In *Proceedings of the ACM International Conference on Multimodal Interaction*.

Binh H Le, Xiaohan Ma, and Zhigang Deng. 2012. Live speech driven head-and-eye motion generators. *IEEE transactions on visualization and computer graphics* 18, 11 (2012), 1902–1914.

Sooha Park Lee, Jeremy B Badler, and Norman I Badler. 2002. Eyes alive. In *ACM Transactions on Graphics (TOG)*, Vol. 21. ACM, 637–644.

Sergey Levine, Philipp Krähenbühl, Sebastian Thrun, and Vladlen Koltun. 2010. Gesture controllers. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 124.

Sergey Levine, Christian Theobalt, and Vladlen Koltun. 2009. Real-time prosody-driven synthesis of body language. In *ACM Transactions on Graphics (TOG)*, Vol. 28. ACM, 172.

Yuanzhi Liang, Qianyu Feng, Linchao Zhu, Li Hu, Pan Pan, and Yi Yang. 2022. SEEG: Semantic Energized Co-Speech Gesture Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10473–10482.

Xian Liu, Qianyi Wu, Hang Zhou, Yinghao Xu, Rui Qian, Xinyi Lin, Xiaowei Zhou, Wayne Wu, Bo Dai, and Bolei Zhou. 2022. Learning Hierarchical Cross-Modal Association for Co-Speech Gesture Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10462–10472.

Xiaohan Ma and Zhigang Deng. 2009. Natural eye motion synthesis by modeling gaze-head coupling. In *Virtual Reality Conference, 2009. VR 2009. IEEE*. IEEE, 143–150.

Stacy Marsella, Yuyu Xu, Margaux Lhommet, Andrew Feng, Stefan Scherer, and Ari Shapiro. 2013. Virtual character performance from speech. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 25–35.

Michael Mathieu, Camille Couprie, and Yann LeCun. 2015. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440* (2015).

Yoichi Matsuyama, Hikaru Taniyama, Shinya Fujie, and Tetsunori Kobayashi. 2010. Framework of Communication Activation Robot Participating in Multiparty Conversation.. In *AAAI Fall Symposium: Dialog with Robots*.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

Bilge Mutlu, Toshiyuki Shiwa, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. 2009. Footing in human-robot conversations: how robots might shape participant roles using gaze cues. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*. ACM, 61–68.

Kazuhiro Otsuka, Yoshinao Takemae, and Junji Yamato. 2005. A Probabilistic Inference of Multiparty-conversation Structure Based on Markov-switching Models of Gaze Patterns, Head Directions, and Utterances. In *Proceedings of the 7th International Conference on Multimodal Interfaces* (Torento, Italy) *(ICMI '05)*. ACM, New York, NY, USA, 191–198.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2536–2544.

Christopher Peters and Carol O'Sullivan. 2003. Attention-driven eye gaze and blinking for virtual humans. In *ACM SIGGRAPH 2003 Sketches & Applications*. ACM, 1–1.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016b. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396* (2016).

Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. 2016a. Learning what and where to draw. In *Advances in Neural Information Processing Systems*. 217–225.

Kerstin Ruhland, Sean Andrist, Jeremy Badler, Christopher Peters, Norman Badler, Michael Gleicher, Bilge Mutlu, and Rachel Mcdonnell. 2014. Look me in the eyes: A survey of eye and gaze animation for virtual agents and artificial systems. In *Eurographics State-of-the-Art Report*. 69–91.

Matthew Stone, Doug DeCarlo, Insuk Oh, Christian Rodriguez, Adrian Stere, Alyssa Lees, and Chris Bregler. 2004. Speaking with hands: Creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 506–513.

Roel Vertegaal, Gerrit van der Veer, and Harro Vons. 2000. Effects of gaze on multiparty mediated communication. In *Graphics Interface*. 95–102.

Vinoba Vinayagamoorthy, Maia Garau, Anthony Steed, and Mel Slater. 2004. An eye gaze model for dyadic interaction in an immersive virtual environment: Practice and experience. In *Computer Graphics Forum*, Vol. 23. Wiley Online Library, 1–11.

Congyi Wang, Fuhao Shi, Shihong Xia, and Jinxiang Chai. 2016. Realtime 3d eye gaze animation using a single rgb camera. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 118.

Xiaolong Wang and Abhinav Gupta. 2016. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision*. Springer, 318–335.

Paul Watzlawick, Janet Beavin Bavelas, and Don D Jackson. 2011. *Pragmatics of human communication: A study of interactional patterns, pathologies and paradoxes*. WW Norton & Company.

Yanzhe Yang, Jimei Yang, and Jessica Hodgins. 2020. Statistics-based Motion Synthesis for Social Conversations. In *Proceedings of the 2020 ACM SIGGRAPH/Eurographics symposium on Computer animation*.

Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S Paek, and In So Kweon. 2016. Pixel-level domain transfer. In *European Conference on Computer Vision*. Springer, 517–532.

Youngwoo Yoon, Bok Cha, Joo-Haeng Lee, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. 2020. Speech Gesture Generation from the Trimodal Context of Text, Audio, and Speaker Identity. *ACM Transactions on Graphics* 39, 6 (2020).

Youngwoo Yoon, Woo-Ri Ko, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. 2019. Robots Learn Social Skills: End-to-End Learning of Co-Speech Gesture Generation for Humanoid Robots. In *Proc. of The International Conference in Robotics and Automation (ICRA)*.

Yi Yu and Simon Canales. 2019. Conditional LSTM-GAN for Melody Generation from Lyrics. *arXiv preprint arXiv:1908.05551* (2019).

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.