electronics

MDPI

Article

Modeling the Conditional Distribution of Co-Speech Upper Body Gesture Jointly Using Conditional-GAN and Unrolled-GAN

Bowen Wu [1,2,*], Chaoran Liu [3], Carlos Toshinori Ishi [1,3,*] and Hiroshi Ishiguro [2,3]

1 Interactive Robot Research Team, Institute of Physical and Chemical Research (RIKEN), Kyoto 619-0237, Japan
2 Graduate School of Engineering Science, Osaka University, Osaka 565-0871, Japan; ishiguro@irl.sys.es.osaka-u.ac.jp
3 Hiroshi Ishiguro Laboratories, Advanced Telecommunications Research Institute International (ATR), Kyoto 619-0237, Japan; chaoran.liu@atr.jp
* Correspondence: wu.bowen@irl.sys.es.osaka-u.ac.jp (B.W.); carlos.ishi@riken.jp (C.T.I.)

Abstract: Co-speech gestures are a crucial, non-verbal modality for humans to communicate. Social agents also need this capability to be more human-like and comprehensive. This study aims to model the distribution of gestures conditioned on human speech features. Unlike previous studies that try to find injective functions that map speech to gestures, we propose a novel, conditional GAN-based generative model to not only convert speech into gestures but also to approximate the distribution of gestures conditioned on speech through parameterization. An objective evaluation and user study show that the proposed model outperformed the existing deterministic model, indicating that generative models can approximate real patterns of co-speech gestures better than the existing deterministic model. Our results suggest that it is critical to consider the nature of randomness when modeling co-speech gestures.

Keywords: gesture generation; social robots; generative model; neural network; deep learning

Citation: Wu, B.; Liu, C.; Ishi, C.T.; Ishiguro, H. Modeling the Conditional Distribution of Co-Speech Upper Body Gesture Jointly Using Conditional-GAN and Unrolled-GAN. *Electronics* **2021**, *10*, 228. https://doi.org/10.3390/electronics10030228

Academic Editor: Michael Wehner
Received: 4 December 2020
Accepted: 22 December 2020
Published: 20 January 2021

1. Introduction

Human-like robots and virtual agents have human appearances, and they are expected to use both verbal and non-verbal behaviors to communicate, like humans do when interacting with others. One crucial non-verbal behavior is the use of hand gestures [1,2]. These spontaneous hand movements accompany speech to complement or even supplement the information relayed by a speaker [3]. The modeling of the relationship between gestures and speech can be incorporated in human-like agents to express themselves comprehensively.

Recently, machine learning and deep learning have achieved great success in generating gestures. The related studies mainly aim at optimizing the parameters of a model to convert speech features into gesture sequences. For instance, the effect of recurrent models such as gated recurrent unit (GRU) and long-short term memory (LSTM) on mapping mel-frequency cepstrum coefficient (MFCC) features of speech to gestures has been analyzed in a study in which a bi-directional LSTM network learned how to map MFCC features to 3D joint coordinates on a skeleton from a dataset collected using motion capture (MOCAP) hardware and software [4]. However, these generation methods are based on a strong assumption: the mapping from speech to gesture is injective, i.e., only one gesture can be generated by these models for one speech segment. In reality, there are alternatives to almost any gesture. Numerous examples help to explain this phenomenon, such as using left, right, or both hands, hands at different heights and radii, and so forth. Additionally, a human may perform new gestures that have never been performed before. We consider this randomness to be an essential part of co-speech gestures and thus aim to design a generative model to incorporate the randomness of co-speech gestures.

Inspired by the success of generative adversarial nets (GANs) for image generation, we propose a GAN-based generative model that can convert speech into gestures while preserving randomness. To optimize the model, we used a discriminator to give dynamic feedback on the generator results. Furthermore, the effect of mode collapse, which is a common type of failure in GAN training, is minimized by using the unrolled generative adversarial net (Unrolled-GAN) algorithm. We experimented with our model on a Japanese speech/gesture dataset. The evaluation shows that the proposed model can approximate real gesture distributions better than baseline could. User studies also confirm the proposed model is effective, showing a significant difference between the results generated by the proposed model and that of the baseline.

The contribution of this work is three-fold: (1) We propose a novel deep-learning-based generative model to generate co-speech gestures. (2) We propose a strategy for changing gesture patterns by manipulating the randomly sampled vector, and we improve the performance. (3) We confirmed that the proposed model outperformed the existing deterministic model through objective and subjective experiments.

The rest of this article is organized as follows: In Section 2, we discuss the research related to the present study. Section 3 briefly mentions the existing methods that are substantial to our work and describes the details of the proposed model and implementation. In Section 4, the objective evaluation metrics and user study are explained, and the obtained results and interpretation are presented. In Section 5, we discuss observations made during our experiment and the limitations and future directions of our approach. Our implementation is available at https://github.com/wubowen416/co-speech-gesture-generation-using-CGAN.

2. Related Work

2.1. Generative Adversarial Nets (GAN)

The essence of GAN is a min–max game between a generator and a discriminator. While the discriminator is optimized to recognize whether its inputs are sampled from real data or are fake data generated by the generator, the generator tries to deceive the discriminator by learning how to generate data that resembles real data. This adversarial system will reach a Nash equilibrium once the generator learns to generate real data. Intuitively, this is equivalent to the generator approximating the real data distribution. Refrerence [5] confirmed this hypothesis by proving that the generator tries to minimize the Jensen–Shannon divergence between the generated distribution and the real data distribution when the discriminator is optimal.

Conditional generative adversarial nets (CGAN) can generate an entity in a specific category [6]. It adds the same conditional labels to both the generator and discriminator. Mathematically, the distribution to which the GAN's generator is trying to approximate is replaced by the conditional distribution conditioned on a specific category. Reference [7] used CGAN to model head motion with speech as the conditional input.

Mode collapse is a common failure in GAN training, i.e., the generator outputs identical results for any noise vector from the prior. By unrolling the discriminator, unrolled-GAN allows the generator to "look into the future" to prevent the discriminator from overfitting on a specific training sample, thereby reducing the effect of mode collapse [8].

2.2. Gesture Generation

Studies on the generation of human-like gestures for robots started years ago. Early on, robot gestures were only designed for a few pre-defined scenarios [9]. The first automatic method was the so-called ruled-based method. A set of human gesture patterns was recorded as sequences of joint data, and their occurrences were statistically studied in relation with the lexicon. These results were then summarized as a number of rules to decide which gesture to select from the recorded database [10]. An advanced rule-based method was proposed to separately model different parts of the human body to generate different combinations as a whole [11].

Beyond writing rules, statistical models were also adopted. These models learn co-occurrences between pre-defined high-level speech features and gesture features from the collected data. In [12], for example, abstract concepts were selected from speech text using WordNet. Then, the extracted concepts were mapped to a gesture sample cluster based on gesture functions (i.e., iconic, metaphoric, and so forth) using data-driven probabilistic modeling. The prosody peak of the speech was automatically analyzed to indicate timing and perform a pre-defined beat gesture. The relationship between iconic gestures and lexicon was automatically learned from the corpus using a Bayesian decision network [13]. A dynamic Bayesian network was also utilized to model several meaningful behaviors (e.g., nod) while considering synchronization with speech [14]. The relationship between the prosodic features of speech and rhythmic gestures was modeled using modified hierarchical factored conditional restricted Boltzmann machines (HFCRBMs) [15]. Various characteristics of natural language were analyzed to determine gesture type and posture by using conditional random fields [16]. However, the methods proposed in these studies require elaborate feature engineering of the data collected from humans. The shape of the gesture was constrained to those appearing in the collected data in these studies.

Since data analysis is tedious and time-consuming, machine learning and deep learning approaches have been utilized to automatically map speech to gestures. A hidden Markov model was used to generate pointing gestures from audio features [17]. The effect of recurrent models, such as gated recurrent unit (GRU) and long-short term memory (LSTM), on mapping Mel-frequency cepstrum coefficient (MFCC) features of speech to gestures has been analyzed [4,18]. Text has also been used as input to generate meaningful gestures using sequence-to-sequence neural networks [19]. In [20], text was encoded using bidirectional encoder representations from transformers (BERT) in order to be concatenated with audio features to generate gesture sequences. Due to the high dimensionality characteristic of human motion, a denoising autoencoder (DAE) was used to reduce the number of dimensions of motion to help the neural network to generalize [21]. Reference [22] made use of labeled gesture phase information to constrain the dynamics of the generated gestures. The individual style was concerned with separately training different neural networks with the L1 distance and discriminative loss on a particular person's data [23]. A style transfer model aimed at generating gestures with a personal style from the voices of others was also proposed [24]. Relatively few studies have dealt with probabilistic generation. Reference [25] used MoGlow to generate gestures while controlling the height, radius, or speed by inputting a control variable. However, this work uses mel-frequency power spectrograms as speech features, we use solely prosodic features of speech as the input to the model.

The premise of the above studies is that correlations exist between speech and gesture. In this study, we generate multiple gesture sequences for one utterance. By treating speech features as conditional input, we utilized the concept of CGAN, through which a Gaussian distribution is mapped to the gesture distribution conditioned on the speech features, and realized a one-to-many mapping from speech to gesture.

3. Materials and Methods

3.1. Problem Formulation

The notation used in the rest of this article is as follows: for a speech segment of length T, the features extracted from the audio signal are $\mathbf{s} = [s_t]_{t=1:T}$. The sequence of absolute positions of each joint in three-dimensional (3D) space is $\mathbf{j} = [j_t]_{t=1:T}$, where $j_t = [x_t^i, y_t^i, z_t^i]_{i=1:K}$, and K is the total number of joints. The problem of generating gesture from speech can then be defined as to parameterize a model G by a parameter set θ such that $\mathbf{j}^{(m)} = G_\theta(\mathbf{s}^{(m)})$. Furthermore, we aim to model the conditional distribution $X_\mathbf{j}$ conditioned on the distribution $X_\mathbf{s}$. To achieve this, the model takes a random variable z sampled from a normal distribution $N(0,1)$. Thus, the problem becomes one of finding a parameter set θ such that $p(\mathbf{j}|\mathbf{s}) = G_\theta(z|\mathbf{s}), \mathbf{j} \sim X_G, \mathbf{s} \sim X_\mathbf{s}, z \sim N(0,1)$. The error between the param-

eterized distribution and the real distribution is defined as $dist(p(\mathbf{j}|\mathbf{s})_{\mathbf{j}\sim X_G}, p(\mathbf{j}|\mathbf{s})_{\mathbf{j}\sim X_j})$ to optimize G_θ. A discriminator parameterized by ϕ is optimized to be the measurement of this error. The method of optimizing D_ϕ and G_θ is discussed in Section 3.3.

3.2. Feature Extraction

The motion data in the corpus is composed of joint rotations and offsets of each joint. We used the protocol provided in [21] to convert the joint's rotation values into absolute position values (APV) in 3D space, which is how our problem is posed in Section 3.1. As the active movements are mostly of the upper body, we used only the upper body's APVs as the training labels.

The speech features used in this study are prosodic features. Prosodic features include fundamental frequency (f0), intensity, and their first and second derivatives; they reflect the rhythm of speech. Although MFCC features are frequently used in automatic speech recognition (ASR), they are not preferred here because the extracted features are used as conditions in model D. Low-dimensional features are expected to yield better results than high-dimensional ones, since high-dimensionality conditions will drastically reduce the number of samples included in that condition. An opensource audio signal processing package, Parselmouth, was used to extract the intensity and fundamental frequency from the speech signal. First, 200 frames of every second feature were extracted by using a window size of 40 milliseconds and hop length of 5 ms. Then, the features are averaged every ten frames to be 20 frames per second (fps) to match the frame rate of the motion data.

3.3. Methodology

Our model utilizes the architecture of CGAN, where speech features are used as a condition. An overview is shown in Figure 1.

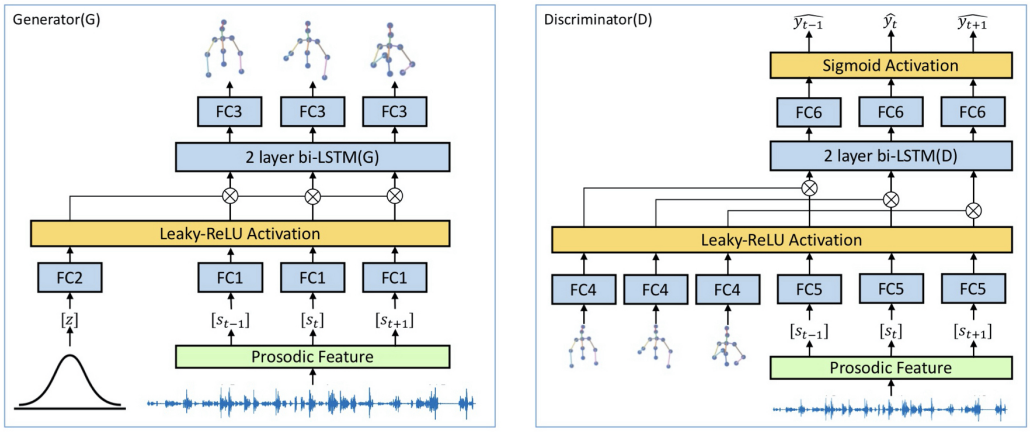


Figure 1. An overview of the proposed model. For the generator, the output of FC2 is replicated to have the same time steps with **s**, and then be concatenated. For the discriminator, the outputs of FC4 are of the same length as **s**. The concatenation follows the order of the sequence.

During the generating phase, a randomly sampled vector (noise vector) z from the Gaussian prior is replicated to have the same length as the speech features. Next, z and speech features are processed by fully-connected layers (FC1 and FC2), respectively; then they are concatenated and fed into a two-layer bidirectional long-short term memory (bi-LSTM) [26]. A sequence-wise fully-connected layer then takes the output of the previous layers and outputs a sequence of vectors indicating each joint's absolute positions in 3D space. The reason for replicating a fixed-length random vector instead of sampling a sequence length wise random vector is that we want to maintain the output motion's consistency along the entire sequence. To optimize the generator, we optimize the discriminator simultaneously to compute the error between the generated distribution and

the real distribution conditioned on speech features. The vector of motion sequence and the corresponding speech features are concatenated and fed into a two-layer bi-LSTM layer. The output is squashed between 0 and 1 by using a sigmoid function, and the value indicates whether the input motion is real and corresponding with the speech features. Instead of outputting only one scalar for the whole sequence by the discriminator, we prefer to output one scalar for each time step. The reason for doing so is that although LSTM is claimed to be capable of capturing long-term dependencies, in practice, its effectiveness decreases when the sequence grows relatively long. The equation for optimizing generator and discriminator is

$$\max_{D} \min_{G} \frac{1}{m} \sum_{i=1}^{m} log(D(j^{(i)}, s^{(i)})) - log(D(G(z, s^{(i)}), s^{(i)})) \quad (1)$$

where m is the number of samples, G is the generator, and D is the discriminator, j is the value of the joint positions, s is the speech features, and z is the noise vector.

In our experiment, we found that each noise vector corresponds with a particular pattern of motion, i.e., motions with the same pattern are generated when using the same noise vector throughout the sequence, a result that is not desirable. To increase variations of the generated motions, we proposed a strategy of generating variating noise vectors for a certain length of speech sequence. Specifically, multiple independently sampled noise vectors with the same length are concatenated to be the noise vector input to the model. The length of the concatenated noise vector is the same as the length of the speech feature input. The algorithm is shown in Algorithm 1.

Algorithm 1 Algorithm for variating noise vectors.

Require: T, time steps of speech features. F, time of replicating the same noise vector.
1: $K \leftarrow \text{ceil}(T/F)$ //Compute the number of chunks
2: $zs \leftarrow [z, ..., z]_F \sim N(0, 1)$ //Sample first chunk of noise vector
3: **for** 0 to K **do** //Concatenate sampled noise vector to the first one
4: Sample $P \sim Uniform(0, 1)$
5: **if** $P > 0.5$ **then** //Replicate the previous noise vector
6: append $zs_{:-F}$ to zs
7: **else** //Sample another noise vector
8: $zs_1 \leftarrow [z_1, ..., z_1]_F \sim N(0, 1)$
9: append zs_1 to zs
10: **end if**
11: **end for**

On the other hand, a common failure during GAN training is mode collapse, i.e., the generator outputs identical results for any noise vector from the prior. In practice, we found that the algorithm for unrolled-GAN reduced the effect of mode collapse that appeared in our experiment setting. However, since we used the LSTM layer, the original unrolled-GAN algorithm will tremendously increase the training time. To avoid this problem, we simplified the algorithm and found in our experiment that a similar result was achieved with a shorter training time. Note that we are not claiming that the original algorithm is replaceable by this simplified version. The proposed algorithm is shown in Algorithm 2. As a brief explanation, in every iteration, the discriminator is trained once, and the parameters of the discriminator are stored. Then, the discriminator is trained multiple times; then, it is used as the loss function of the generator to train the generator once. Finally, before the iteration ends, the parameters of the discriminator are restored to the previously stored discriminator parameters.

Algorithm 2 Algorithm for training the proposed model.

Require: α, the learning rate. k_{unroll}, the unrolling steps. m, the batch size. *iteration*, the number of training iterations.
Require: ϕ_0, initial discriminator parameters. θ_0, initial generator parameters.
Require: $(X_{\mathbf{j}}, X_{\mathbf{s}})$, pairs of value of joint positions and speech features.

for 0 to *iteration* **do**
  Sample $\{\mathbf{j}^{(i)}, \mathbf{s}^{(i)}\}_{i=1}^{m} \sim (X_{\mathbf{j}}, X_{\mathbf{s}})$ a batch from the real data
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim N(0,1)$ a batch from the prior
  $g_\phi \leftarrow \nabla_\phi [\frac{1}{m} \sum_{i=1}^{m} log(D_\phi(j^{(i)}|s^{(i)})) - log(D_\phi(G_\theta(z^{(i)}, s^{(i)})|s^{(i)}))]$
  $\phi \leftarrow \phi + \alpha \cdot g_\phi$ //Update discriminator
  $backup_\phi \leftarrow \phi$ //Store this discriminator
  for 0 to k_{unroll} **do** //Update discriminator k_{unroll} times
    Sample $\{\mathbf{j}^{(i)}, \mathbf{s}^{(i)}\}_{i=1}^{m} \sim (X_{\mathbf{j}}, X_{\mathbf{s}})$ a batch from the real data
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim N(0,1)$ a batch from the prior
    $g_\phi \leftarrow \nabla_\phi [\frac{1}{m} \sum_{i=1}^{m} log(D_\phi(j^{(i)}|s^{(i)})) - log(D_\phi(G_\theta(z^{(i)}, s^{(i)})|s^{(i)}))]$
    $\phi \leftarrow \phi + \alpha \cdot g_\phi$
  end for
  Sample $\{\mathbf{s}^{(i)}\}_{i=1}^{m} \sim X_{\mathbf{s}}$ a batch from the real data
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim N(0,1)$ a batch from the prior
  $g_\theta \leftarrow \nabla_\theta [\frac{1}{m} \sum_{i=1}^{m} log(D_\phi(G_\theta(z^{(i)}, s^{(i)})|s^{(i)}))]$
  $\theta \leftarrow \theta - \alpha \cdot g_\theta$ //Update generator
  $\phi \leftarrow backup_\phi$ //Restore the discriminator before unrolling
end for

3.4. Corpus

We evaluated our model on the dataset proposed in [27], in which pairs of recorded audio and motion are provided. The content is an undergraduate student answering questions in Japanese like in an interview while standing and gesturing. The motion data were recorded using a motion capture studio. The motion data files contain information on the offset and rotation of each joint, from which each joint's absolute position can be derived. The audio is saved as WAV files (sampling rate 22,050 Hz, 16 bits). There are 1049 sentences in this dataset: 68.41% are metaphoric gestures, 23.73% are beat gestures, and others are iconic and deictic gestures. The dataset is 298 minutes long.

3.5. Implementation

Since the motions are represented as absolute positions in 3D space, the means and variances of each joint's values are considerably different, which can drastically decrease the model's performance. Therefore, we performed a min–max scaling strategy on the motion features by using Equations (3) and (4) to squash the feature within the range of -1 to 1. The speech features were also scaled using Equations (3) and (4) to be compatible with the motion features in terms of the values' size. Note that the scaling was performed using parameters calculated only from the data in the training set.

$$X_{std} = (X - X_{min})/(X_{max} - X_{min}) \tag{2}$$

where X_{min} and X_{max} are calculated from the split training set.

$$X_{scaled} = X_{std} \times 2 - 1 \tag{3}$$

Numerous studies on gesture generation cut the gesture sequence into several slices to approximate the effect of data augmentation. Instead, we used the entire sequence of speech and motion as samples. The hyper-parameters for training the proposed model used in our experiment are listed in Table 1. The number of nodes of the proposed model is detailed in Table 2. The Adam optimizer was used to update the parameters. The initial

parameters of all layers were drawn from a Gaussian distribution with 0 mean and 1 variance. We saved the trained model every ten iterations and generated samples using speech utterances in the test set. After assessing the quality of these generated results, we chose the generator of the 1000 iteration.

Table 1. Hyper-parameter settings of training.

Hyper-Praram	Value
Iterations	2000
Batch size	32
Learning rate for G	10^{-4}
Learning rate for D	10^{-4}
Unrolling steps	10
Beta for optimizer	(0.9, 0.999)

Table 2. Number of nodes of the proposed model.

Layer	Number of Nodes
FC1	64
FC2	64
2 layer bi-LSTM(G)	128
FC3	33
FC4	64
FC5	64
2 layer bi-LSTM(D)	128
FC6	1

4. Results

4.1. Baseline

To compare the proposed model with the deterministic generation method, the model proposed in [21] was selected as a baseline. We used the protocol provided by the authors and reproduced the reported results. We cut the upper body motion generated using the baseline model in order to make the comparison. Since the dataset for the baseline model is already split into training, development, and test sets, we used the split test set for the evaluation. There are 45 samples in the test set.

4.2. Quantitative Evaluation

It is common for a deterministic model to use the L1 distance or average position error (APE) to evaluate the generated results. Since our motivation is to model the distribution of gestures, it is not appropriate to evaluate the precision of generated key points in comparison with the ground truth. Instead, kernel density estimation (KDE) is a useful tool for approximating the distribution of the data; it was used in [5] for image generation and in [7] for head motion generation. The output of KDE is the log-likelihood of the input samples based on the fitted density function using reference samples. In this study, we used the generated gesture sequences from the speech input in the test set to fit the density function and used the ground truth as the input of KDE. Therefore, as the output value tends to 0, the generator better fits the real data distribution.

We used Algorithm 1 to generate one motion sequence for every speech sample in the test set. The generated motions were used to fit a distribution. The optimal bandwidth in the KDE model was obtained using a grid search with 3-fold cross-validation. Then, the log-likelihood of the real motions in the test set was calculated using the fitted distribution. We also studied how F in Algorithm 1 affects the results. The results are shown in Table 3. The values are the average of five calculations.

Table 3. Quantitative comparison between models. Ground truth is the log-likelihood of real motions in the test set in the kernel density estimation (KDE) distribution fitted using the ground truth itself, indicating the best results that can be expected. * uses replicated noise vectors to generate motions. ** jointly uses the proposed model and the proposed Algorithm 1.

Model	Log-Likelihood	Standard Error
Ground Truth	−29.98	1.03
Baseline [21]	−508.82	87.61
CGAN *	−245.67	44.72
Unrolled-CGAN *	−118.91	17.03
Proposed (F = 20) **	−177.86	29.36
Proposed (F = 30) **	−161.30	26.78
Proposed (F = 40) **	−107.58	15.21
Proposed (F = 50) **	−107.98	15.77
Proposed (F = 60) **	−126.20	19.01

4.3. Motion Dynamics Distribution

Motion dynamics (i.e., velocity) are imperative to human perception. As we aim to model the distribution of human gestures, one reason that the proposed model outperforms the baseline model is assumed to be that the velocity distribution of the motion generated by the proposed model is more similar to the ground truth than the baseline model is. We confirmed this assumption by plotting the histogram of the average velocity of all joints, shoulder, wrist, and hand: the histograms of the proposed model were more similar to the ground truth than those of the baseline, while the hand velocity distributions of both methods were comparable to the ground truth (Figure 2).

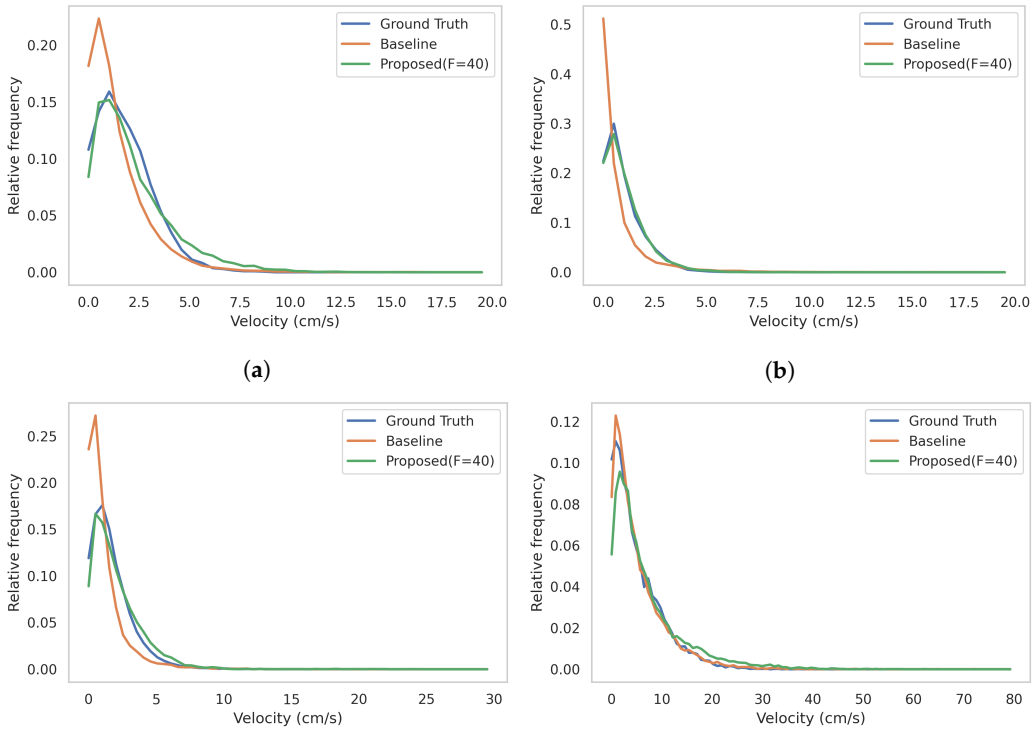


Figure 2. (**a**) Histogram of average velocity over all joints. (**b**) Histogram of velocity of shoulders. (**c**) Histogram of velocity of elbows. (**d**) Histogram of velocity of hands.

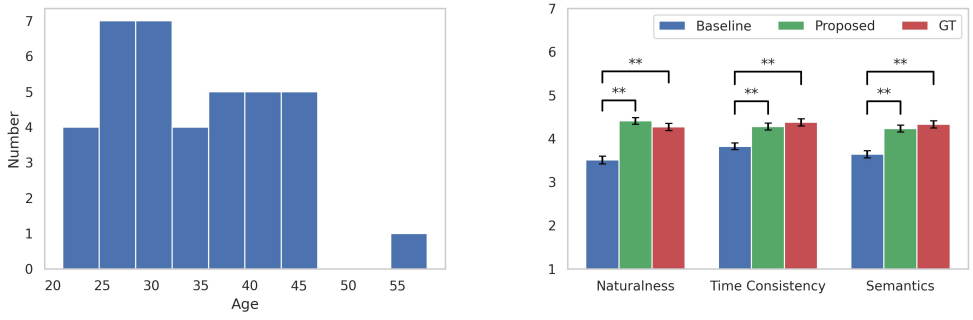
4.4. User Study

The ultimate goal of gesture generation is to generate human-like motions. Here, we conducted a user study to subjectively evaluate the motions generated by the baseline and the proposed model against the ground truth. The Likert scale in the baseline paper was used to evaluate motions on three different scales based on three specific statements for each (Table 4).

Table 4. Likert scale used in the user study.

Scale	Statements (Translated from Japanese)
Naturalness	Gesture was natural Gesture was smooth Gesture was comfortable
Time Consistency	Gesture timing was matched to speech Gesture speed was matched to speech Gesture pace was matched to speech
Semantics	Gesture was matched to speech content Gesture well described speech content Gesture helped me understand the content

Before the evaluation, participants viewed three ground-truth videos to help them understand the real motions that would be played. The first part of the questionnaire was a ranking task. We prepared 12 sets, three videos within each set. There were four sets for ranking (1) the baseline and full proposed model, (2) CGAN with or without unrolling, and (3) the ground truth, baseline, and full proposed model. After watching a set of videos, participants were asked to rank the gesture depicted in the videos in order of naturalness. The second part was to assign a score to each statement within each scale. This part compared the baseline, ground truth, and the proposed model. After watching each video, participants were asked to assign a score to each statement. The value ranged from (0) to (7), where (0) indicates strongly disagree and (7) indicates strongly agree. There were five videos for the baseline, ground truth, and the proposed model, and the score for each scale was the average of three scores of the statements. As a result, five scores for each subject were obtained on each scale in Table 4 from one participant. Proposed (F = 40) was used to generate videos for the full proposed model.



(**a**) (**b**)

Figure 3. (**a**) Age distribution of participants. (**b**) Scores of each scale for different models. **: $p < 0.002$.

We recruited 38 participants (19 male, 19 female, all native Japanese speakers, average 34 years old) through a cloud sourcing service. Analysis of variance (ANOVA) was conducted to test the difference between the three groups' scores. All three scales passed the ANOVA test with $p < 0.001$. Tukey's honestly significant difference test (Tukey HSD) was conducted to test if there was a significant difference pairwisely. For the naturalness

scale, there was a significant difference between the baseline (M = 3.51, SE = 0.09) and the full proposed model (M = 4.41, 0.08), $p < 0.002$, and between the baseline and the ground truth (M = 4.27, SE = 0.08), $p < 0.002$. There was no significant difference between the full proposed model and the ground truth, $p = 0.46$. For the time consistency scale, there was a significant difference between the baseline (M = 3.82, SE = 0.08) and the full proposed model (M = 4.28, 0.08), $p < 0.002$, and between the baseline and the ground truth (M = 4.38, SE = 0.08), $p < 0.002$. There was no significant difference between the full proposed model and ground truth, $p = 0.65$. For the semantics scale, there was a significant difference between the baseline (M = 3.64, SE = 0.08) and the full proposed model (M = 4.23, 0.08), $p < 0.002$, and between the baseline and the ground truth (M = 4.33, SE = 0.08), $p < 0.002$. There was no significant difference between the full proposed model and the ground truth, $p = 0.68$. The age distribution and scores on the scales are shown in Figure 3. These results indicate that the motions generated by the full proposed model were perceived as more natural than those of the baseline and were similar to the ground truth. The ranking tasks revealed similar results (Figure 4).

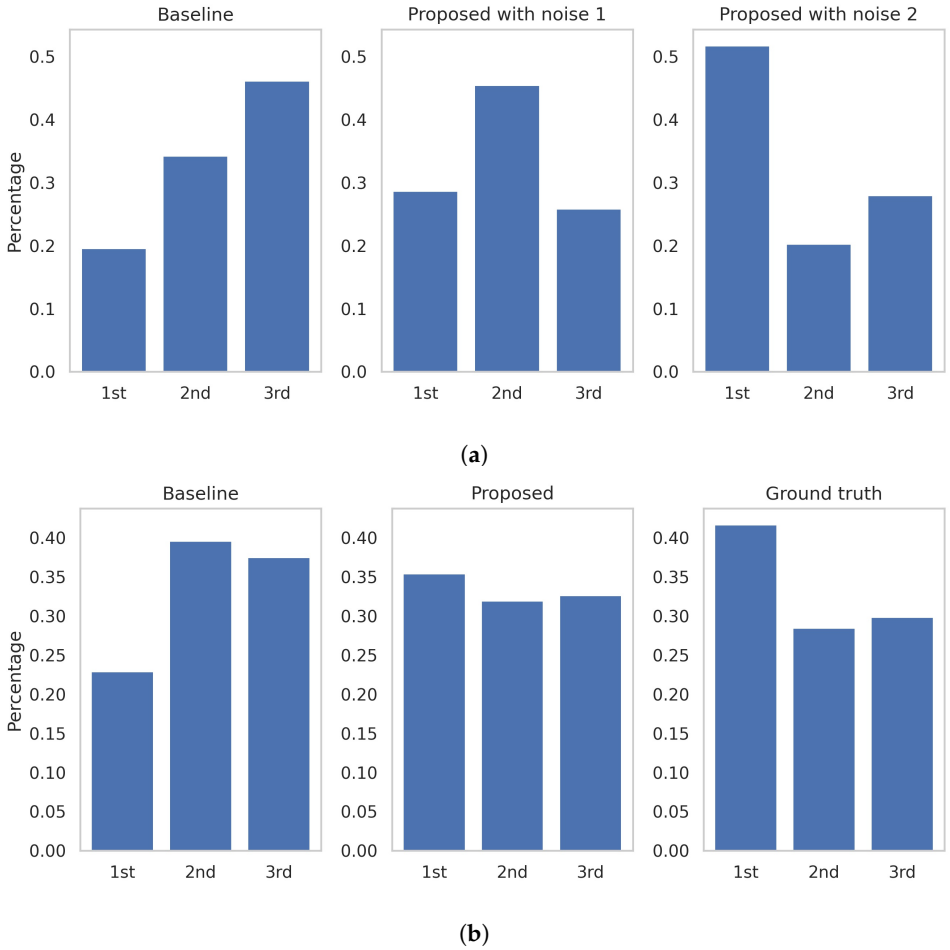


Figure 4. (**a**) Ranking of results of baseline and proposed model with two different noises. (**b**) Ranking of baseline, proposed model, and ground truth.

5. Discussion

5.1. Inappropriateness of Using Euclidean Distance as a Loss Function

There are mainly two reasons that the Euclidean distance, i.e., L1-distance or L2-distance, is not suitable for the gesture generation task. Firstly, motion may be realistic even though the Euclidean distance gives a large error; for example, suppose that the ground truth is a gesture with the left hand and the generated gesture is a mirror symmetry of the ground truth performed by the right hand. It is not reasonable to punish such

realistic motions simply because they are not identical to the ground truth because of the randomness of human gestures. Secondly, the Euclidean distance tends to ignore small unrealistic parts of motions, underestimating the error. For example, even if one frame is modified to be unrealistic for a real motion sequence, the Euclidean distance will still give a relatively low error since most of the sequence is correct. This is inconsistent with human perception because humans immediately notice unrealistic motions.

Instead of using the Euclidean distance as the loss function, the GAN architecture gives the error by looking at a low-dimensional manifold, i.e., the output of the last hidden layer of the discriminator. Specifically, the discriminator judges whether the low-dimensional manifold of the generated samples is similar to that of the real samples, thus preventing the motion from being unrealistic while allowing more variation in the generated motion. Another benefit of this approach is that by interpolating on a low-dimensional manifold, realistic motions that are not in the dataset can be generated.

5.2. Unrolling for More Variation

Since we input a noise vector, by manipulating it, we can interpolate among motions and thereby generate new gestures that are not in the dataset. However, the ranking results shown in Figure 5 indicate that the CGAN without unrolling was as natural as CGAN with unrolling, and better than not using prosody input in the discriminator. This similarity is probably because the generated results of CGAN are already human-like compared with the proposed model, even though the generated motions of CGAN without unrolling are all the same pattern. The ranking task designed in the questionnaire cannot discriminate between performing the same pattern all the time and changing patterns occasionally. Intuitively, always performing the same pattern is not human-like while occasionally changing patterns is human-like.

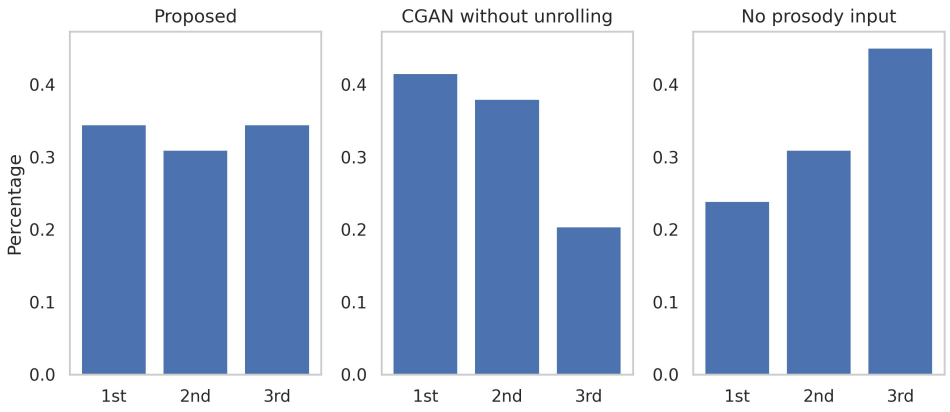


Figure 5. Ranking of the results of the proposed model, the conditional generative adversarial nets (CGAN) model, and model without prosody input for the discriminator.

5.3. The Role of the Noise Vector

To investigate the effect of changing the noise vector, we input a 5-second-long sinusoidal wave to the proposed model. Through the prosodic feature extraction, there were a total of 139 frames of speech features, as well as the generated motions.

The noise vector controls the motion pattern. The results in Figure 6 show that the proposed model can be a controller of the movement pattern. Although we have not investigated much on this topic, disentanglement of the noise vector in the proposed model is worthy of future investigation.

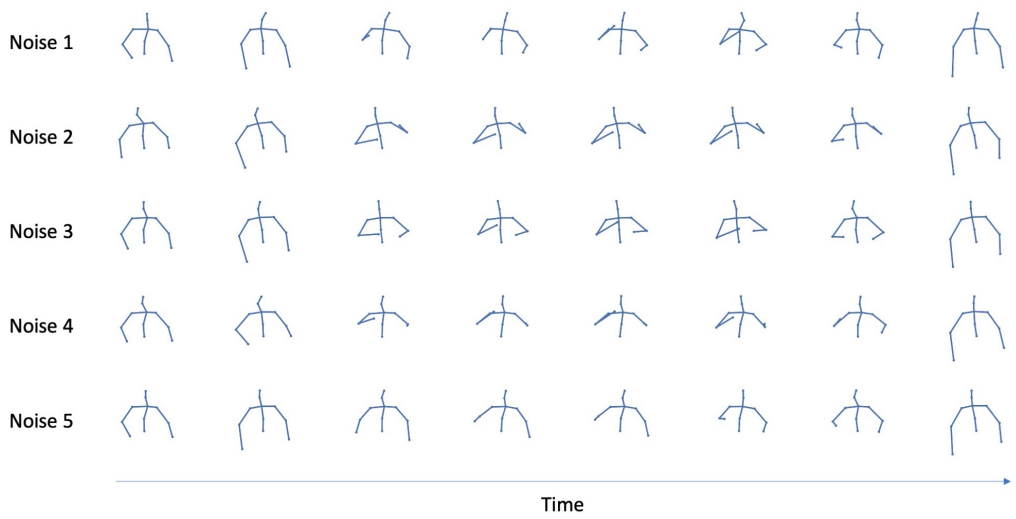


Figure 6. Motions generated using different noise vectors on the same utterance.

We expect that noise can maintain gesture patterns across the whole utterance, i.e., the same pattern shifts according to the prosodic peak in the utterance. By shifting the phase of the sinusoidal signal and plotting the generated results, shifting effects appear as the shifts in the apex of a gesture as the prosodic peak shifts, as shown in Figure 7.

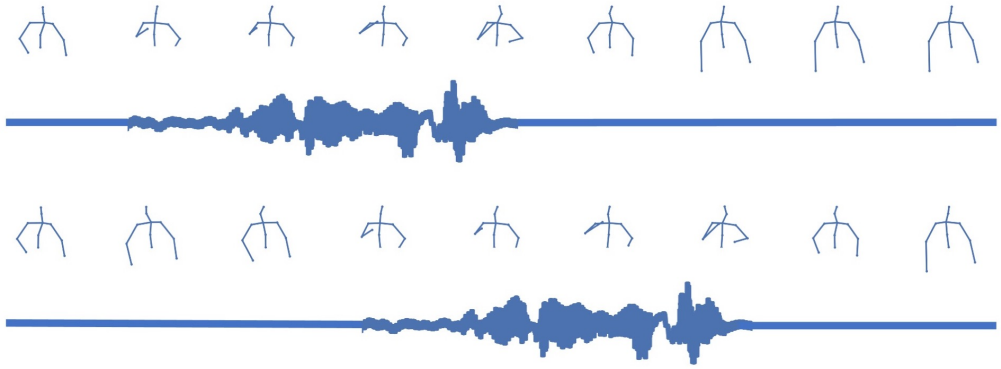


Figure 7. Results generated from shifting speech inputs. The y-axis is two generated results. For the first row, the signal starts at 1 s and ends at 4 s. For the second row, the signal starts at 3 s and ends at 6 s.

5.4. The Role of Prosody as a Condition

Since the prosodic features we used are the fundamental frequency and intensity, we generated motions with different f0 and intensity condition inputs to investigate their effect on the generated gesture. According to [28,29], f0 and intensity are correlated with the heights of the hands and size of the motion. Thus, here, we focused on the heights of the hands and the size of the motion.

The reference values of f0 were set to 100, 150, 200, and 250 Hz. First, a sinusoidal wave signal of a certain f0 was generated. Then, using the trained model, motion sequences were generated. The corresponding results are shown in Figure 8. It is clear that the size becomes larger and the height of the hand becomes higher as f0 increases. Correlations were also observed between intensity and the heights of hands and the size of motion (Figure 9).

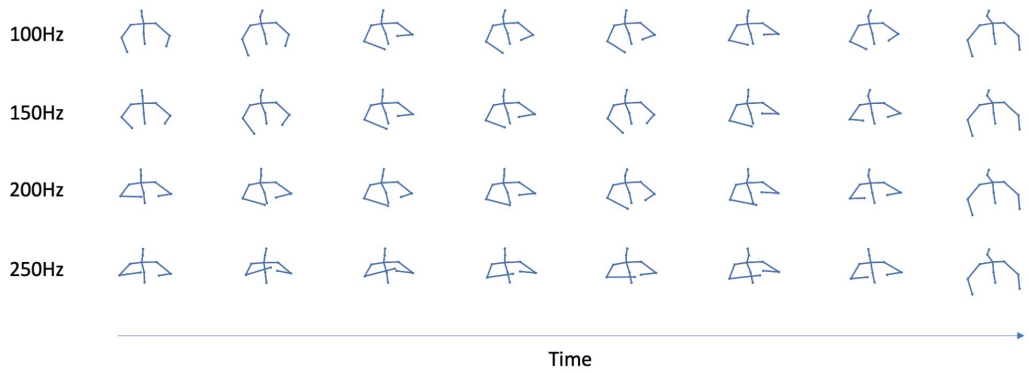


Figure 8. Results generated from different f0s. The x-axis is the time step of the generated frames (1, 15, 30, 45, 60, 75, 90, and 105 from left to right). The y-axis corresponds to different f0s, i.e., 100, 150, 200, and 150 Hz.

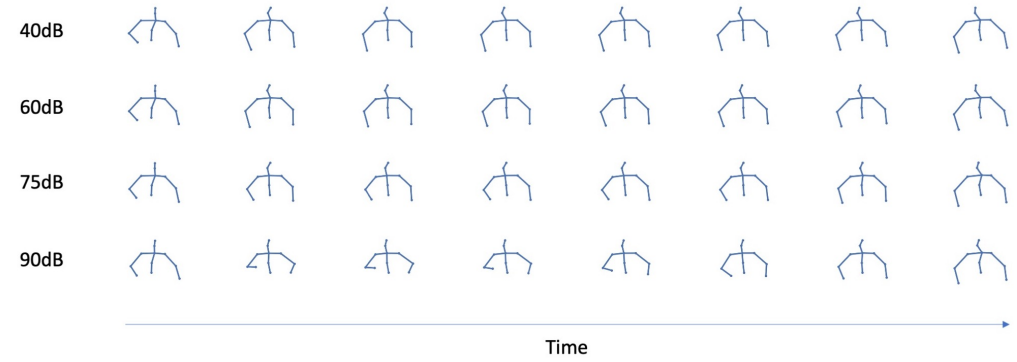


Figure 9. Results generated from different intensities. The x-axis is the time step of the generated frames (1, 15, 30, 45, 60, 75, 90, and 105 from left to right). The y-axis corresponds to the amplitude of the sinusoidal signal, i.e., 40, 60, 75, and 90 dB.

6. Conclusions

Human-like agents play an important role in human–computer interaction, and it is crucial to equip them with the capability of gesturing so that they can be expressive. We presented a model for producing co-speech gestures by modeling the conditional distribution of gestures conditioned on speech features. Incorporating unrolled-GAN and our proposed algorithm, our model outperformed the existing deterministic model in objective and subjective evaluations. Our work provides a powerful tool for human-like agents to express thoughts, thereby enhancing human–computer interactions. Moreover, the success of the distributional modeling revealed that future research in this field should focus more on gesture distribution. Human-like agents should be widely used in HCI. However, without the ability to gesture well, they are too inexpressive to be understood or empathized with by humans. Though our gesture generation model performs better in terms of naturalness and time consistency, the lack of semantics (i.e., meaningful gestures) is still a considerable obstacle to perfect modeling of human gestures; further research should focus on developing a model with semantically meaningful gestures.

Author Contributions: Conceptualization, B.W., C.L., C.T.I. and H.I.; data curation, B.W.; formal analysis, B.W.; funding acquisition, C.T.I. and H.I.; investigation, B.W.; methodology, B.W. and C.L.; project administration, C.T.I. and H.I.; resources, C.L. and C.T.I.; software, B.W.; supervision, C.L., C.T.I. and H.I.; validation, B.W., C.L., C.T.I. and H.I.; visualization, B.W.; writing—original draft, B.W.; writing—review and editing, B.W., C.L. and C.T.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Grant-in-Aid for Scientific Research on Innovative Areas JP20H05576.

Institutional Review Board Statement: The study was conducted according to the guidelines of the ethical review approved by the Ethical Committee of the Advanced Telecommunications Research Institute International (ethical review number 20-605, 17 February 2020).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Publicly available datasets were used in this study. This data can be found here: https://www.dropbox.com/sh/j419kp4m8hkt9nd/AAC_pIcS1b_WFBqUp5ofBG1Ia?dl=0.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kendon, A. *Gesture: Visible Action as Utterance*; Cambridge University Press: Cambridge, UK, 2004.
2. McNeill, D. *Gesture and thought*; University of Chicago Press: Chicago, IL, USA, 2008.
3. Iverson, J.M.; Goldin-Meadow, S. Why people gesture when they speak. *Nature* **1998**, *396*, 228. [CrossRef] [PubMed]
4. Hasegawa, D.; Kaneko, N.; Shirakawa, S.; Sakuta, H.; Sumi, K. Evaluation of speech-to-gesture generation using bi-directional LSTM network. In Proceedings of the 18th International Conference on Intelligent Virtual Agents, Sydney, Australia, 5–8 November 2018; pp. 79–86.
5. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
6. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
7. Sadoughi, N.; Busso, C. Novel realizations of speech-driven head movements with generative adversarial networks. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 6169–6173.
8. Metz, L.; Poole, B.; Pfau, D.; Sohl-Dickstein, J. Unrolled generative adversarial networks. *arXiv* **2016**, arXiv:1611.02163.
9. Cassell, J. A framework for gesture generation and interpretation. In *Computer Vision for Human-Machine Interaction*; Cambridge University Press (CUP): Cambridge, UK, 1998; pp. 191–215.
10. Hartmann, B.; Mancini, M.; Pelachaud, C. Implementing expressive gesture synthesis for embodied conversational agents. In *International Gesture Workshop*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 188–199.
11. Baumert, D.; Kudoh, S.; Takizawa, M. Design of conversational humanoid robot based on hardware independent gesture generation. *arXiv* **2019**, arXiv:1905.08702.
12. Ishi, C.T.; Machiyashiki, D.; Mikata, R.; Ishiguro, H. A speech-driven hand gesture generation method and evaluation in android robots. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3757–3764. [CrossRef]
13. Bergmann, K.; Kopp, S. GNetIc–Using bayesian decision networks for iconic gesture generation. In Proceedings of the International Workshop on Intelligent Virtual Agents, Amsterdam, The Netherlands, 14–16 September 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 76–89.
14. Sadoughi, N.; Busso, C. Speech-driven animation with meaningful behaviors. *Speech Commun.* **2019**, *110*, 90–100. [CrossRef]
15. Chiu, C.C.; Marsella, S. How to train your avatar: A data driven approach to gesture generation. In Proceedings of the International Workshop on Intelligent Virtual Agents, Reykjavik, Iceland, 15–17 September 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 127–140.
16. Ishii, R.; Katayama, T.; Higashinaka, R.; Tomita, J. Generating Body Motions Using Spoken Language in Dialogue. In Proceedings of the 18th International Conference on Intelligent Virtual Agents (IVA '18), Sydney, Australia, 5–8 November 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 87–92. [CrossRef]
17. Sargin, M.E.; Aran, O.; Karpov, A.; Ofli, F.; Yasinnik, Y.; Wilson, S.; Erzin, E.; Yemez, Y.; Tekalp, A.M. Combined gesture-speech analysis and speech driven gesture synthesis. In Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 9–12 July 2006; pp. 893–896.
18. Ferstl, Y.; McDonnell, R. Investigating the use of recurrent motion modelling for speech gesture generation. In Proceedings of the 18th International Conference on Intelligent Virtual Agents, Sydney, Australia, 5–8 November 2018; pp. 93–98.
19. Yoon, Y.; Ko, W.R.; Jang, M.; Lee, J.; Kim, J.; Lee, G. Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4303–4309.
20. Kucherenko, T.; Jonell, P.; van Waveren, S.; Henter, G.E.; Alexanderson, S.; Leite, I.; Kjellström, H. Gesticulator: A framework for semantically-aware speech-driven gesture generation. *arXiv* **2020**, arXiv:2001.09326.
21. Kucherenko, T.; Hasegawa, D.; Henter, G.E.; Kaneko, N.; Kjellström, H. Analyzing input and output representations for speech-driven gesture generation. In Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents, Paris, France, 2–5 July 2019; pp. 97–104.
22. Ferstl, Y.; Neff, M.; McDonnell, R. Multi-objective adversarial gesture generation. *Motion Interact. Games* **2019**, 1–10.

23. Ginosar, S.; Bar, A.; Kohavi, G.; Chan, C.; Owens, A.; Malik, J. Learning individual styles of conversational gesture. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3497–3506.
24. Ahuja, C.; Lee, D.W.; Nakano, Y.I.; Morency, L.P. Style Transfer for Co-Speech Gesture Animation: A Multi-Speaker Conditional-Mixture Approach. *arXiv* **2020**, arXiv:2007.12553.
25. Alexanderson, S.; Henter, G.E.; Kucherenko, T.; Beskow, J. Style-Controllable Speech-Driven Gesture Synthesis Using Normalising Flows. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2020; Volume 39, pp. 487–496.
26. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
27. Takeuchi, K.; Kubota, S.; Suzuki, K.; Hasegawa, D.; Sakuta, H. Creating a gesture-speech dataset for speech-based automatic gesture generation. In Proceedings of the International Conference on Human-Computer Interaction, Vancouver, BC, Canada, 9–14 July 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 198–202.
28. Laukka, P.; Juslin, P.; Bresin, R. A dimensional approach to vocal expression of emotion. *Cogn. Emot.* **2005**, *19*, 633–653. [CrossRef]
29. Dael, N.; Goudbeek, M.; Scherer, K.R. Perceived gesture dynamics in nonverbal expression of emotion. *Perception* **2013**, *42*, 642–657. [CrossRef]