Pose Sequence Generation with a GCN and an Initial Pose Generator

Kento Terauchi[1] and Keiji Yanai[1]

Department of Informatics, The University of Electro-Communications, Tokyo, Japan
terauchi-k@mm.inf.uec.ac.jp, yanai@cs.uec.ac.jp

Abstract. The existing methods on video synthesis have succeeded in generating higher quality videos by using guide information such as human pose skeletons, segmentation masks and optical flows as auxiliary information. Some existing video generation methods on human motion adopts a two-step video generation consisting of generation of pose sequences and video generation from pose sequences. In this paper, we focus on the first stage, the generation of pose sequences, in the whole processing of video generation of human motion. We incorporate a Graph Convolutional Network (GCN) and an initial pose generator into the model to model poses more explicitly and to generate pose sequences naturally. The experimental results show that the proposed method can generate better quality pose sequences than the conventional methods by improving the initial pose generation and introducing GCN.

Keywords: video generation, human pose sequence generation, GCN, VAE

1 Introduction

In recent years, image generation has achieved great success with the development of Generative Adversarial Networks (GAN) and Variational Auto Encoder (VAE). GANs and VAEs have been applied to various image translation tasks by conditioning on labels, images, sounds and texts. However, video generation is more difficult task than still image synthesis because it requires temporal modeling in addition to spatial modeling. Unconditional video generation is a difficult task in general. Then, many existing video generation methods employ guide information such as motion flows to synthesize a video. Some guide-based video synthesis methods have succeeded in generating higher quality videos by using guide information. There are various types of videos to be generated, such as videos of people, videos of natural scenery such as cloud flow, and videos of driving scenery in a city. In the case of human motion video generation, most of the existing video generation methods such as Cai et al. [2] adopt a two-step video generation consisting of two stages: the generation of pose sequences and the generation of videos from pose sequences. In this paper, we focus on the first stage, the generation of pose sequences, in the whole processing of video generation of human motion.

The main objective of ours is to generate pose sequences with actions corresponding to input labels. To do that, we propose to use a Graph Convolutional Network (GCN) [10] to model human poses more explicitly and enable natural pose sequence generation. GCN is often used as a method for action recognition. However, there are still few methods that use GCN for generating pose sequences. In this paper, we propose a model that incorporates GCN and takes the structure of poses into account. In addition, we add an initial pose generator to the network of Action2Motion [5] which we use as a base method in this work. The experimental results show that the proposed method can generate better quality pose sequences than the conventional methods.

2 Related Work

2.1 Image generation

Image generation methods have achieved great success in recent years with the development of VAE [7] and GAN [4]. GANs can generate high-quality images by alternately training generators and descriptors in an adversarial manner. Generator tries to generate a plausible image. Discriminator tries to distinguish whether the input image is real or generated fake. The VAE reconstructs the image by maximizing the variational lower bound, while the autoencoder forces the latent variables to be normally distributed. In recent years, various improvements of GANs, such as StyleGAN [6] and BigGAN [1], have achieved remarkable success in generating high-resolution images. Some VAE models have also been proposed, such as VQ-VAE2 [11], which can generate images with high accuracy comparable to GAN. In video generation, the quality of individual frames contributes greatly to the overall quality of the video. Therefore, image generation methods are often applied to video generation.

2.2 Video generation

Video generation methods include unconditional video generation from noise and generation using flow and segmentation as guides. In unconditional video generation, VGAN [15] divides the generation into foreground and background. TGAN [13] considers the movement of the latent space with time. MoCoGAN [14] separates the latent space into motion and content, and DVDGAN [3] enables high-quality video generation by training on a large amount of data. However, these methods often have difficulties in generation quality and computational complexity. On the other hand, the guide-based generation methods can simplify the generation and improve the quality of the generation. There are several methods such as a human pose based method [2], a flow based method [12], a segmentation based method [16] and a method that adds 3D information to the guide [9]. In the guide-based method, the generation can be manipulated by editing the guide. In this study, we consider video generation in two stages: pose sequence generation and video generation from pose sequences, In particular, we aim to make the generation of pose sequences more natural.

2.3 Pose sequence generation

The generation of pose sequences has several promising applications, such as the use of pose guides as a preliminary step in the generation of videos, and the behavior of 3D models. There are existing studies of pose sequence generation. For example, overall generation using CNNs, sequential generation using RNNs, and reference-based [17] which enables various motion generation by cutting out several references and interpolating between them. Cai et al. [2] deals with two stages of video generation: pose sequence generation and video generation from pose sequences, as shown in Figure 1. For the generation of pose sequences, a model that learns the movement of latent variables and a model that sequentially generates pose sequences from latent variables are trained using adversarial loss with a discriminator. The model can also be used for interpolation and prediction of pose sequences by searching for latent variables corresponding to a certain pose through optimization. For generating pose sequences, a generator only considers time variation by moving latent variables. On the other hand, in our study, we use GRU and Graph Convolutional Network (GCN) to consider both temporal and structural information simultaneously.

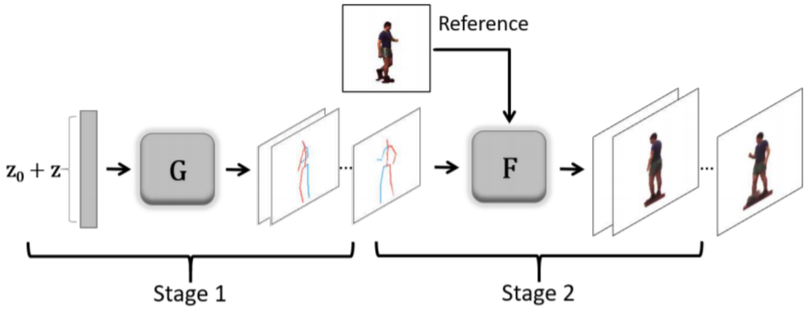


Fig. 1: Cai et al.(cite from [2])

Action2Motion [5] focuses only on the generation of pose sequences, which are generated from conditioning on action labels only. While the prior distribution of VAEs usually follows a normal distribution, Action2Motion assumes that the prior distribution changes as time changes, the architecture infer the prior distribution from the information of the previous time step. In practice, as shown in the Figure 2, the architecture consist of the encoder of the previous frame, the encoder of the next frame and the decoder(in the Figure 2, described as Prior, Posterior and Generator). The architecture introduces Prior Loss which brings the distributions of the outputs of the encoder of the previous frame and the encoder of the next frame closer together. pose sequence is reconstructed by the decoder. In the decoder, the GRU is used to capture the temporal information.

Overall architecture is trained with Prior Loss and MSE Loss. During testing, the encoder of the previous frame produces an output similar to the encoder of the next frame. From the pose of the previous frame, the pose of the next frame can be generated sequentially. The architecture uses the representation of the body's center coordinate and the angle of each joint. This paper also introduces HumanAct12, a dataset that focuses only on the generation of pose sequences. HumanAct12 is temporally smoother and less noisy than the pose sequence annotations obtained from motion capture in the previous datasets. In our sequential generation model, we modify the generation of the initial frame based on Action2Motion and introduce GCN into the decoder to generate a more natural pose sequence.

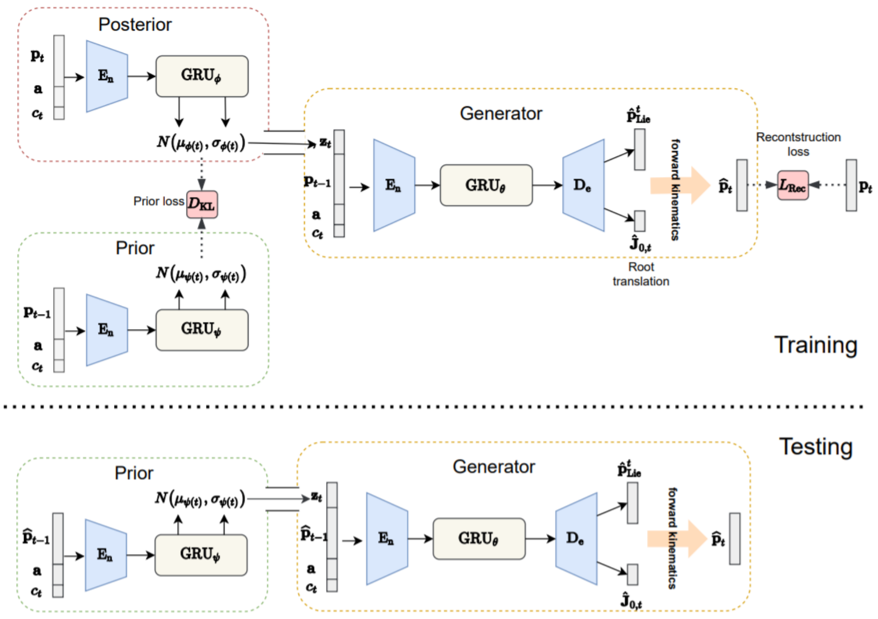


Fig. 2: Action2Motion.(cite from [5])

2.4 Graph convolution network

Graph Convolutional Network (GCN) [10] enables convolution on graphs with complex shapes, unlike CNNs which have grid-like relationships. Attempts to convolve human poses with GCNs have been made in action recognition [8] , and 3D pose estimation from 2D in [18]. However, there are still few studies using GCN for pose generation. In this study, we apply GCN to the generation of pose sequences. By capturing both time and structure, we hope to make the generation of pose sequences more realistic.

3 Method

We propose a method for generating pose sequences corresponding to a given motion category. The proposed model is generating frames one by one sequentially based on Action2Motion [5]. We use GRU and GCN to consider both time and structure at the same time. The training loss functions and pose representation are also based on Action2Motion.

3.1 Proposed model

In the proposed model, two changes are made on the model of Action2Motion: we add (1) an initial pose generator, and (2) GCN to the decoder. We aim to improve the quality of generation by these changes. The basic architecture is the same as that of Action2Motion, which consists of a previous frame encoder E_pr, a next frame encoder E_po, and a decoder D. We add an initial frame encoder and an initial condition decoder that consider the first frame. By incorporating GCN into the decoder, we aim to learn a model with more expressive power.

Initial pose generator In Action2Motion, when the initial frame is generated, the information of the previous frame is treated as zero, which reduces the diversity of the initial frame of the generated pose sequence. We believe that special treatment of the initial frame is necessary. So we propose to add an initial pose generator to the model of Action2Motion. We encode the first pose with the initial frame encoder so that it is normally distributed like VAE. This makes it possible to generate a variety of frames from noise. In addition, when generating the first frame, we give the information decoded by the initial frame condition decoder from the latent space as a condition for the decoder. The whole architecture is shown in Figure 3.

Let the length of the pose sequence be T, the number of joints be J, and the number of categories be C. The pose sequence of the data set is $P = \{p_1, p_2, ..., p_T\}$, the i-th pose representation is $p_i \in \mathbb{R}^{J \times 3}$, and the generated pose sequence is $\hat{P} = \{\hat{p_1}, \hat{p_2}, ..., \hat{p_T}\}$. The conditional vector for the i-th pose, $c_i \in \mathbb{R}^C$, consists of (α, t_i) where α_i is an one-hot vector on action categories, and $t_i \in \mathbb{R}$ is a scalar value on relative duration time.

We obtain the initial latent variable, z_f, by reparameterization from the VAE output of the initial frame encoder, $(\mu_f, \sigma_f^2) = E_f(P, c_1)$ Next, we generate the output of the initial frame condition decoder, $\dot{p_0}$, by $\dot{p_0} = D_f(z_f)$, Using them, the initial frame, $\hat{p_1}$, is generated by the formula, $\hat{p_1} = D(z_f, \dot{p_0}, c_1)$. After the second frame, in the same way as Action2Motion, we obtain the VAE latent values, (μ, σ^2), using the formula, $(\mu, \sigma^2) = E(p_{i-1}, c_i)$. Then we reparameterize it as z, and use the formula, $\hat{p_i} = D(z, p_{i-1}, c_i)$, to generate the pose $\hat{p_i}$.

Decoder The layer structure of the decoder is shown in Figure 3. The decoder takes latent representations and conditions as its inputs and provide them into the two layers of GRU and the two fully connected layers, in the same way as

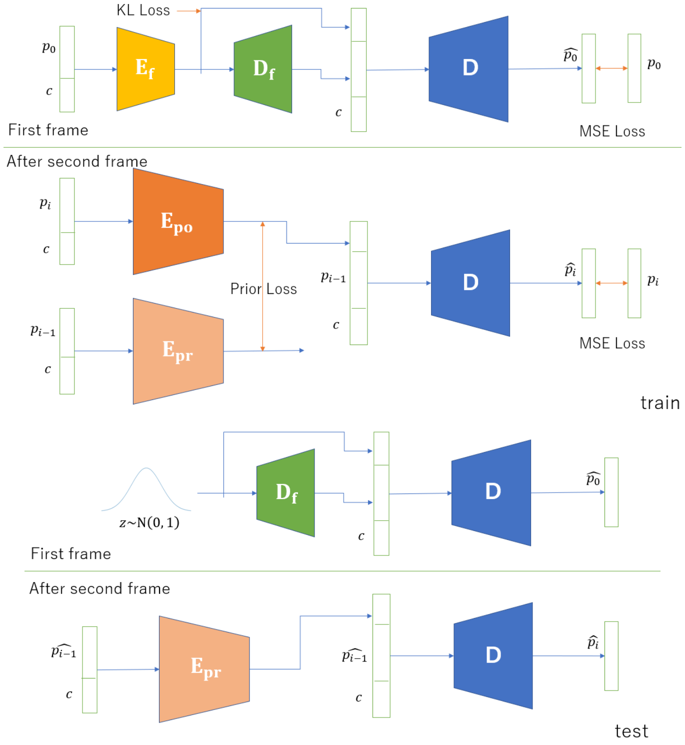


Fig. 3: Overview of the proposed model.

Action2Motion. Then, the decoder embeds it in the pose representation. It learn the structural representation by using three layers of GCN to get a complex representation. We uses a multi-scale GCN that convolves information for each edge in the graph connection. Finally, the output is obtained by passing through a fully connected layer with different weights for each joint. We can maintain the semantic consistency of each joint by passing a fully connected layer to each joint.

3.2 Loss functions

The objective function uses MSE Loss and Prior Loss as in Action2Motion. KL Loss is used to make the output of the initial frame encoder closer to a normal distribution.

$$L_f = -\frac{1}{2}\sum_{j=1}^{dim(z_f)} (1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2) \tag{1}$$

The KL loss allows models to generate the initial frame from noise.

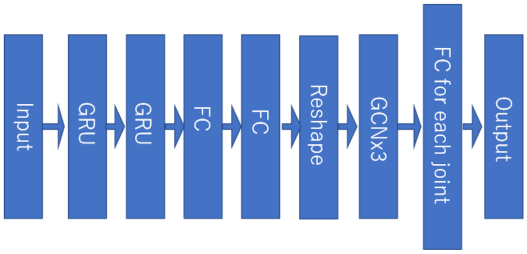


Fig. 4: The layer structure of the decoder.

3.3 Pose representation

The representation of the pose sequence to be learned is similar to the representation in Action2Motion. It consists of the 3D Cartesian coordinates of the center joint of the body and the angular representation of each bone. The angular representation is a Lie algebraic representation, and the rotation is represented by three parameters. When converting to 3D coordinates, the bones connected from the central joint of the body are tilted to the angle of the representation in turn, and the position of the joint moved by the length of the bone is obtained, and the coordinates of the neighboring joint are obtained. In this case, the length of the bones is adjusted manually. By not including the length of the bones in the pose representation, the model can generate stable motions independent of the body size.

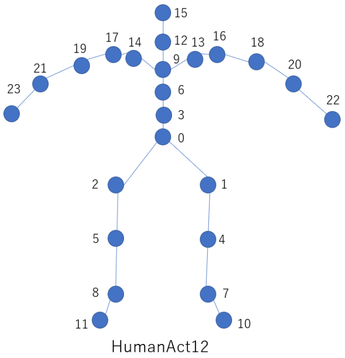
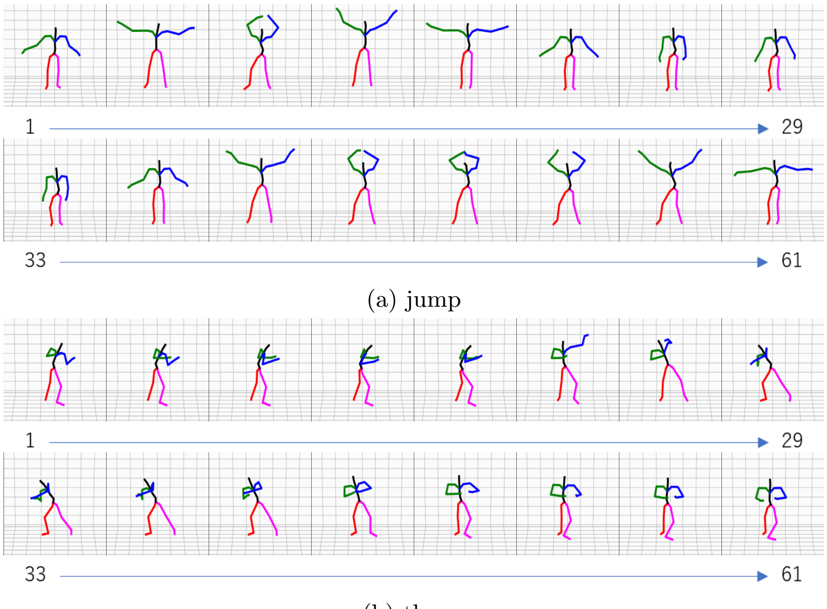


Fig. 5: Location of the joint in HumanAct12.



(a) jump

(b) throw

Fig. 6: The example of pose sequence in HumanAct12.

4 Experiments

We report the results of actual training using the model described in this paper. We evaluate the results qualitatively and quantitatively in order to confirm that the model is able to generate pose sequences corresponding to the categories. The baseline to be compared is Action2Motion [5]. We use HumanAct12 [5] as the training dataset. We use the Adam optimizer to repeat the 6000 epoch training. We also evaluate "Ours (IPG only)" by adding only the initial pose generator (IPG), and "Ours (GCN only)" by incorporating only GCN in the decoder as ablation studies.

4.1 Dataset

HumanAct12 HumanAct12 is the dataset presented in Action2Motion. HumanAct12 is a dataset for action recognition and motion generation that contains 1191 pose sequences ranging from 9 to 403 frames in length with 12 coarse categories and 34 detailed categories. Each pose sequence consists of 24 joints. Each joint is shown in Figure 5. The data used for generation are randomly selected sequences and fixed-length frames are randomly cut out from the sequences. If the pose sequence is less than a fixed length frame, the last frame is padded. Some examples of pose sequences are shown in Figure 6. For the categorical conditioning, we use 12 coarse categories.

4.2 Qualitative evaluation

Examples of the generated results are shown in Figure 7 and Figure 8 which corresponds to “eating” and “running’, respectively. The positions of the 3D joints are drawn in 3D space, and the poses are represented by straight lines connecting the key points. The figures shows the results of (a) the ground-truth and generated videos by (b) Action2Motion and (c) our proposed method. Each human motion sequence is represented with every 4 frames among the total 64 frames. Each of them is generated from different noise vectors using the trained model. The proposed model is able to generate motions similar to the dataset, and even complex motions such as eating and runing are well generated. There are reasonable movements such as the correspondence of the relationship between hands and feet. Compared to Action2Motion, the proposed model is temporally smoother and more natural in its generation.

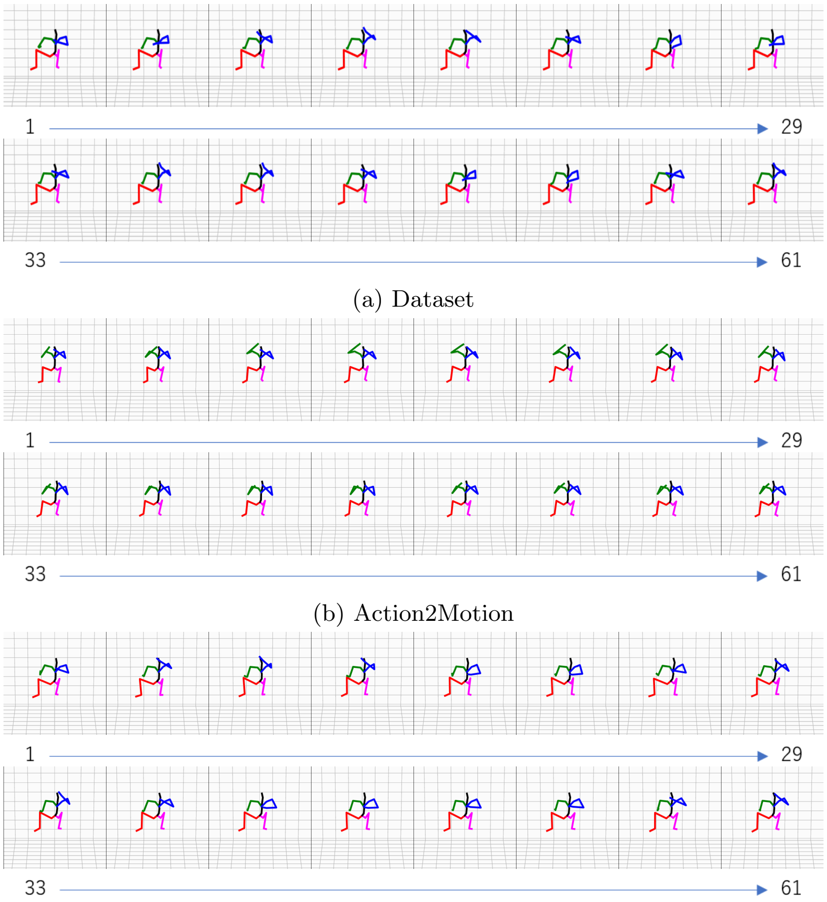
4.3 Interpolation of latent space

To show that the model is not simply storing a data set, we interpolate the latent space. We show that a latent variable in the middle of two latent variables generates a pose sequence in the middle of two pose sequences. If interpolation can be achieved, we can say that the learned model has acquired a continuous representation in the latent space. We generate pose sequences using the learned model from two latent variables generated from random numbers and their intermediate latent variables, respectively. We interpolate in two settings, within the same action and between different actions, and in the interpolation between different actions, we interpolate the action labels at the same time.

The results of interpolation within the same action are shown in Figure 9, and the results between different actions are shown in Figure 10. The figures show the results with every 8 frames with the leftmost image as the first frame. The pose sequence in the middle row is generated by interpolating the latent variables that generate both the pose sequences in the top row and the bottom row. The interpolation in Figure 9 is generated from two kinds of “warming-up” action sequences, and the interpolation in Figure 10 is generated from the “jumping” and “sitting” action sequences. For each interpolation of both the same action and the difference actions, the pose sequence generated from the average of the two latent variables inherits the features of both pose sequences generated from the two latent variables. From this, we can say that interpolation has been achieved.

4.4 Quantitative evaluation

Similar to Action2Motion, we use the following four metrics for evaluation, FID, Accuracy, Diversity, and Multimodality. All the measures are measured by sampling 3000 pose sequences from the dataset and the generated data. The results are shown in Table 1. The proposed method, the results of which are represented as “Ours(FULL)” in the table, outperforms the existing methods in FID while



(a) Dataset

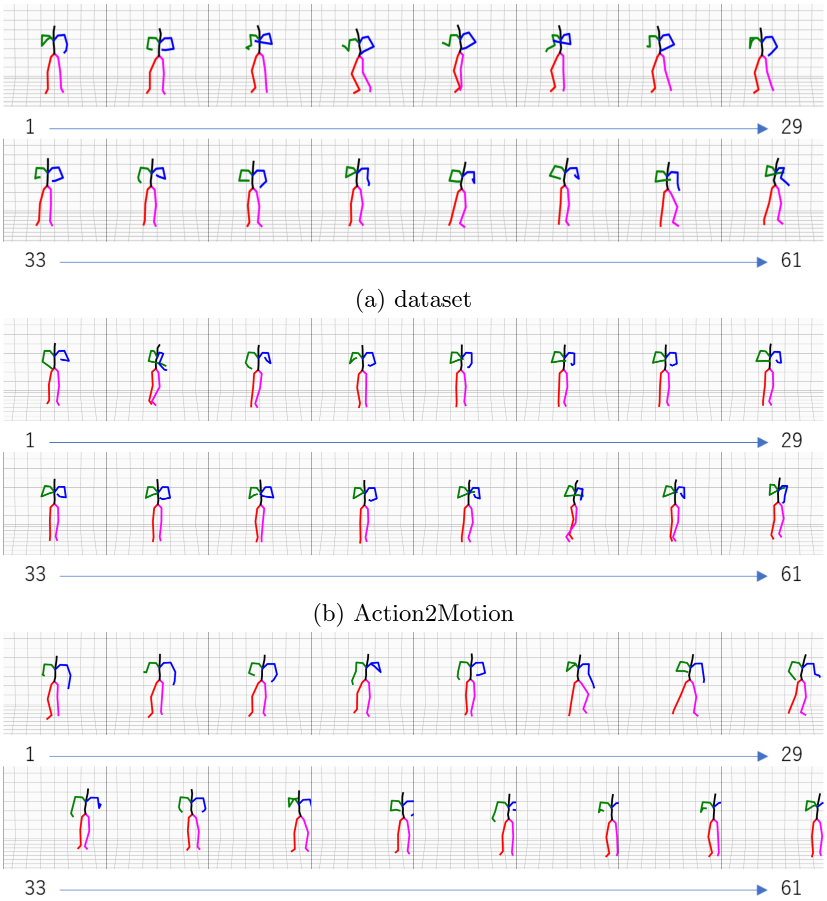
(b) Action2Motion

(c) Ours

Fig. 7: The results of generating pose sequence conditioned on “eat” action.

maintaining the same accuracy. This means that the proposed method is able to generate clear pose sequences more naturally than the existing methods. Note that Diversity and Multimodality should be close to the value of groundtruth (GT), since both too large value of them and too small value of them are not good. Each evaluation metric is explained below.

FID measures the distance between the distributions of the real data and the generated data. We use the feature vectors extracted from the trained model of action recognition. For feature extraction, we use the trained models available on the Action2Motion GitHub. The lower the FID, the closer to the data set and the higher the quality of the generation.



(a) dataset

(b) Action2Motion

(c) Ours

Fig. 8: The results of generating pose sequence conditioned on "run" action.

Accuracy is calculated as classification accuracy by classify the generated pose sequences with the trained action classification model. We use the same model as the model used for FID computation. The higher accuracy, the more clearly the pose sequences with distinct actions are generated.

Diversity is measured as the variance of the pose sequences generated by all the action conditions. With this measure, we judge if the generated sequences are diverse or not. However, even if unnatural sequences are generated, the diversity might be large. Therefore, the diversity value can be an indicator of diverse generation only when the indices of FID and accuracy are good.

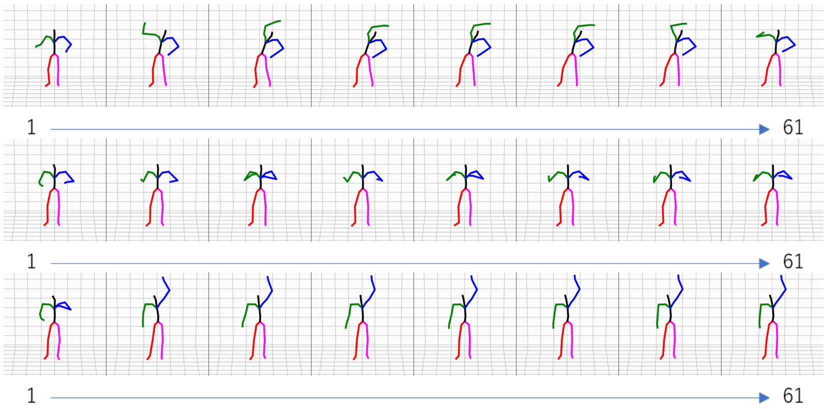


Fig. 9: Interpolation within the “warm-up” action.

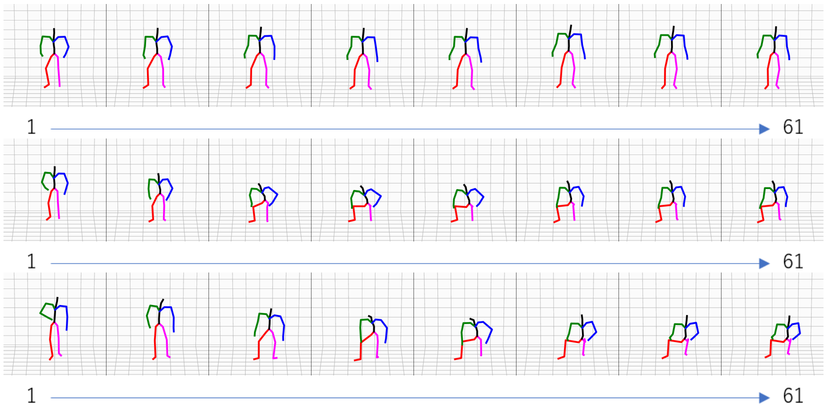


Fig. 10: Interpolation between “jump” action and “sit” action.

Multimoldality measures the diversity of pose sequences generated by each action category condition. Unlike the evaluation of diversity over all the action conditions, we measure the diversity within the same action category. In this measure, we can see whether diversity is maintained within a category or not.

4.5 Ablation study

In this research, two changes are made to Action2Motion: (1) adding IPG (Initial Pose Generator) to the encoder and (2) adding GCN (Graph Convolutional Network) to the decoder. With these two changes, we tried to improve the quality of generation. We examine the impact of these two changes on the quality of generation. We also conducted two evaluations: Ours (IPG only) with only (1) changed, and Ours (GCN only) with only (2) changed.

Table 1: The results of quantitative evaluation.

Method	Accuracy↑	FID↓	Diversity→	Multimodality→
Groundtruth (GT)	0.997	0.092	6.857	2.449
Action2Motion	0.923	2.458	7.032	2.870
Ours(FULL)	**0.924**	2.252	**6.962**	**2.861**
Ours(IPG only)	0.864	**1.979**	6.924	3.388
Ours(GCN only)	0.542	13.599	5.933	3.309

The results are shown in Table 1. Ours (IPG only) outperformed the baseline regarding FID. Its accuracy was inferior to Ours(FULL), and it could not generate a clear pose sequence. Ours (GCN only) failed to generate a clear pose sequence because both Accuracy and FID are by far less than others.

5 Conclusion

In this study, we proposed a model that explicitly captures the structural information by considering the initial frame with the intial pose generator and incorporating GCN. The qualitative results showed that the proposed method was capable of generating complex and diverse motions. Quantitatively, the proposed method outperformed the existing methods and showed more natural generation.

As future works, we plan to make additional evaluations on other datasets, since we have evaluated our method on only one kinds of the dataset. In addition, we like to study a model for generating videos from pose sequences, and actually generate videos from generated poses.

References

1. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: Proc. of International Conference on Learning Representations (2019), https://openreview.net/forum?id=B1xsqj09Fm
2. Cai, H., Bai, C., Tai, Y., Tang, C.: Deep video generation, prediction and completion of human action sequences. In: Proc. of European Conference on Computer Vision. pp. 366–382 (2018)
3. Clark, A., Donahue, J., Simonyan, K.: Adversarial video generation on complex datasets. arXiv preprint arXiv:1907.06571 (2019)
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)
5. Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., Cheng, L.: Action2motion: Conditioned generation of 3d human motions. In: Proc. of ACM International Conference Multimedia (2020)
6. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 4401–4410 (2019)

7. Kingma, D., Welling, M.: Auto-encoding variational bayes. In: Proc. of International Conference on Learning Representations (2014)
8. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 143–152 (2020)
9. Mallya, A., Wang, T.C., Sapra, K., Liu, M.Y.: World-consistent video-to-video synthesis. In: Proc. of European Conference on Computer Vision (2020)
10. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: Proc. of International Conference on Machine Learning. pp. 2014–2023 (2016)
11. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. In: Advances in Neural Information Processing Systems. pp. 14866–14876 (2019)
12. Ren, Y., Li, G., Liu, S., Li, T.H.: Deep spatial transformation for pose-guided person image generation and animation. IEEE Transactions on Image Processing (2020)
13. Saito, M., Matsumoto, E., Saito, S.: Temporal generative adversarial nets with singular value clipping. In: Proc. of IEEE International Conference on Computer Vision. pp. 2830–2839 (2017)
14. Tulyakov, S., Liu, M., Yang, X., Kautz, J.: Mocogan: Decomposing motion and content for video generation. In: Proc. of IEEE Computer Vision and Pattern Recognition (2018)
15. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: Advances in Neural Information Processing Systems. vol. 29, pp. 613–621 (2016)
16. Wang, T., Liu, M., Zhu, J., Liu, G., Tao, A., Kautz, K., Catanzaro, B.: Video-to-video synthesis. In: Advances in Neural Information Processing Systems (2018)
17. Xu, J., Xu, H., Ni, B., Yang, X., Wang, X., Darrell, T.: Hierarchical style-based networks for motion synthesis. In: Proc. of European Conference on Computer Vision (2020)
18. Zhao, L., Peng, X., Tian, Y., Kapadia, M., Metaxas, D.N.: Semantic graph convolutional networks for 3d human pose regression. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 3420–3430 (2019)