Structure-Aware Human-Action Generation

Ping Yu[1], Yang Zhao[1], Chunyuan Li[2], Junsong Yuan[1] and Changyou Chen[1]

[1] The State University of New York at Buffalo
[2] Microsoft Research, Redmond
{pingyu, yzhao63, jsyuan, changyou}@buffalo.edu chunyl@microsoft.com

Abstract. Generating long-range skeleton-based human actions has been a challenging problem since small deviations of one frame can cause a malformed action sequence. Most existing methods borrow ideas from video generation, which naively treat skeleton nodes/joints as pixels of images without considering the rich inter-frame and intra-frame structure information, leading to potential distorted actions. Graph convolutional networks (GCNs) is a promising way to leverage structure information to learn structure representations. However, directly adopting GCNs to tackle such continuous action sequences both in spatial and temporal spaces is challenging as the action graph could be huge. To overcome this issue, we propose a variant of GCNs (**SA-GCNs**) to leverage the powerful self-attention mechanism to adaptively sparsify a complete action graph in the temporal space. Our method could dynamically attend to important past frames and construct a sparse graph to apply in the GCN framework, well-capturing the structure information in action sequences. Extensive experimental results demonstrate the superiority of our method on two standard human action datasets compared with existing methods.

Keywords: action generation, graph convolutional network, self-attention, generative adversarial networks (GAN)

1 Introduction

Recent years have witnessed the development of skeleton-based action generation, which has been applied in a variety of applications, such as action classification [10, 17, 19, 29, 44], action prediction [2, 24, 39] and human-centric video generation [37, 45]. Action generation is still a challenging problem since small deviations in one frame can cause confusion in the entire sequence.

One of the most successful methods for skeleton-based action generation considers skeleton-based action generation as a standard video generation problem [7, 13, 40]. Specifically, the method naively treats skeleton joints as image pixels and sequential actions as videos, without considering the rich structure information among both joints and action frames. The video-generation based methods may produce distorted actions when applied to skeleton generation, if prior structure knowledge is not well leveraged. A first step to consider structure information into action generation is to represent a skeleton as a graph structure

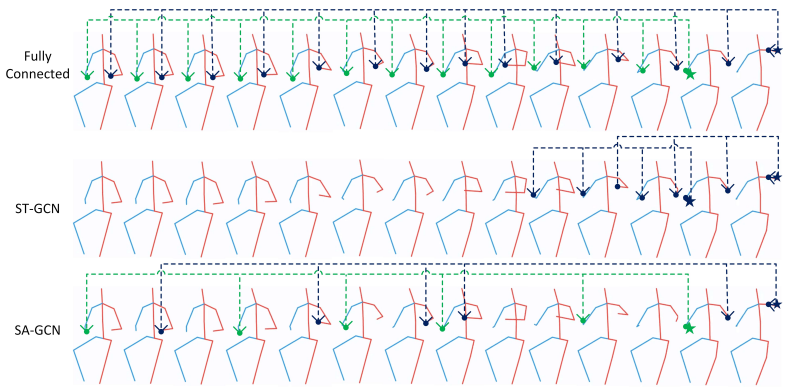

Fig. 1: Comparisons of the construction of action graphs with our proposed method (3rd tow) and two standard methods (1st and 2nd rows) to encode temporal information. First row (*full connection*): the left-hand joint gather information from all left-hand joint of past frames; similar to the right-hand joint. Second row (*ST-GCN*): a 1D convolution of kernel size k is used to encode temporal information. Both the left and right hands could encode information from past k frames with share weights. Third row (*SA-GCN*): both the left- and right-hand joints learn to encode information from a selected left-hand joints based on the attention scores.

to characterize the spatial relations between joints in each frame based on graph convolution networks (GCN) [50, 21, 6]. However, most existing GCN methods do not have the flexibility to process continuous sequential graphs data. This poses a new challenge: *how to construct a representation to effectively incorporate both temporal and spatial structures into action generation?*

Generally speaking, there are two classes of methods with GCN to model action structure information: (*i*) *Full connection*: an entire action sequence is considered as a graph. Each node of the current frame is connected with the corresponding nodes in all the past frames. This construction, however, is computationally very inefficient (if ever possible at all). Moreover, the model could be highly redundant since many frames are similar to each other. (*ii*) *Spatial-temporal graph convolutional networks* [44]: a graph convolution is first applied to intra-frame skeletons, whose extracted features are then applied with a 1D convolution layer to capture temporal information. This method typically requires weight sharing among all nodes, and the ability to model temporal information is somewhat weak.

We advocate that a better solution should be proposed to leverage skeleton structures and gather information from action sequences more efficiently. In this paper, we propose *Self-Attention based Graph Convolutional Networks (SA-GCN)* to build generic representations for skeleton sequences. Our *SA-GCN* aims at building a sparse global graph for each action sequence to achieve both computational efficiency and modelling efficacy. Specifically, for a given frame, the proposed *SA-GCN* first calculates self-attention scores for other frames. Based

on the attention scores, top k past frames with the most significant scores are selected to be connected to the current frame to construct inter-frame connections. Within each frame, the joints are connected as the original skeleton representation. To demonstrate the differences between our construction and the aforementioned two constructions, Fig. 1 illustrates a sequence of samples in terms of every three consecutive frames on the Human 3.6m dataset *Sitting-Down* sequence. As illustrated in the figure, our method can be considered as an adaptive scheme to construct an action graph, with each node assigning a trainable weight instead of a shared weight as in other methods.

The major contributions of this work are summarized in three aspects:

- We propose *SA-GC* layer, a generic graph-based formulation to encode structure information into action modelling efficiently. Our method is the first sparse and adaptive scheme to encode past frame information for action generation.
- By efficiently leveraging action structure information, our model can generate high-quality long-range action sequences with pure Gaussian noise and provided labels as inputs without pretraining.
- Our model is evaluated on two standard large datasets for skeleton-based action generation, achieving superior and stable performance compared with previous models.

2 Preliminaries & Related Work

2.1 Attention Model

Attention models have become increasingly popular in capturing long-term global dependencies [1, 8]. In particular, self-attention [5, 33, 46] mimics human visual attention, allowing a model to focus on crucial regions and to learn the correlation among elements in the same sequence. [38] proposes a non-local operation as a kind of attention on capturing long-range dependencies in videos. [33] develops the transformer model, which is solely based on attention and achieves state-of-the-art on machine translation. Thus, self-attention can typically lead to a better representation learning. One key advance of our proposed model compared with previous ones is that we adopt self attention to efficiently encode frame-wise correlations by inheriting all merits of the self-attention mechanism.

2.2 Skeleton-Based Action Generation

The task of action generation differs from action prediction [3] in that no past intermediate sub-sequence is provided. Directly generating human actions from noise is considered more challenging. The problem has been well studied in early works [4, 27, 28], which applied switching linear models to generate stochastic human motions. These models, however, required a large amount of data to fit a model and are difficult to find an appropriate number of switching states. Later on, the Restricted Boltzmann Machine [30] and Gaussian-process latent variable

models [32, 35, 36] were applied. But they still can not scale to massive amounts of data. The rapid development of deep generative models has brought the idea of recurrent-neural-network (RNN) based Variational Autoencoder (VAE) [20] and Generative Adversarial Net (GAN) models [11, 7, 18, 41, 40, 42, 48]. These models are scalable and usually can generate actions with better quality.

The aforementioned methods still have some limitations, which mainly lie in two aspects. Firstly, spatial relationships among body joints and temporal dynamics along continuous frames have not been well explored. Secondly, these models often require an expensive pre-training phase to capture intra-frame constraints, including the two most recent state-of-the-art works [7, 40]. By contrast, Our work moves beyond these limitations and can be trained from scratch to generate high-quality motions.

2.3 Graph Convolutional Network

GCNs have been achieving encouraging results [44]. In general, they can be categorized into two types: spectral approaches [6, 21] and spatial approaches [50, 26]. The spectral GCN operates on the Fourier domain (locality) with convolution to produce a spectral representation of graphs. The spatial GCN, by contrast, directly applies convolution on the spatially distributed nodes. This work is in the spirit of spatial GCNs and incorporates new ideas of GCNs to fit the task. In particular, to model long-term dependent motion dynamics, we are aware of ideas from graph pruning [47] and jump connection [43], which respectively allows one to extract structure representation more efficiently and to build deeper graph convolutional layers. In terms of GCN-based human motion modelling, the most related work is *ST-GCN* [44], which applies a spatial GCN to a different task of action recognition. This method applied a GCN layer for intra-frame skeletons and then used 1D convolution layer for gathering information in temporal space. All nodes in a frame share weights on the temporal space and could only attend limited range of information, depending on the kernel size of the 1D convolution layer. We will compare our method with *ST-GCN* (for action generation) in Section 4.6.

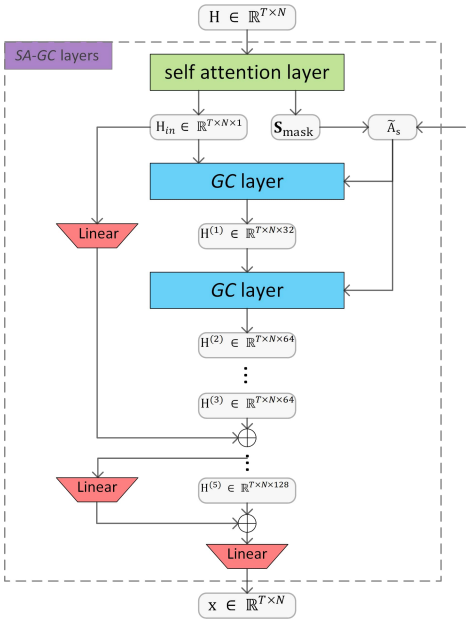

Fig. 2: An illustration of the *SA-GC* layer. $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{A}}_s$ are two adjacency matrices detailed in Section 3.2.

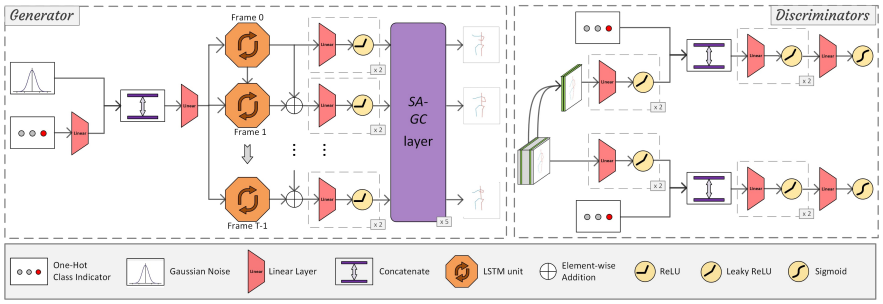

Fig. 3: The overall framework of the proposed method.

3 Structure-Aware Human-Action Generation

Different from the video-generation task, the skeleton-based action generation contains huge amounts of structure information, *e.g.*, intra-frame structural joints information and inter-frame motion dynamics. Directly treating skeleton frames as images will lose most of these structure information, leading to the distortion of some skeletal frames. Moreover, in the context of skeleton-based actions, where only limited positional information is provided, differences between two continuous frames are virtually impossible to be observed. To address these issues, we propose to incorporate GCNs to encode the rich structural information, with additional consideration to reduce computational burden by using self-attention to automatically learn a sparse action graph.

3.1 An Overview of the *SA-GCN*

Fig. 3 illustrates the overall framework of our model for action generation. It follows the GAN framework of video generation [31, 9], which consists of an action generator $\mathcal{G}$ and a dual discriminator: one video-based discriminator $\mathcal{D}_V$ and one frame-based discriminator $\mathcal{D}_F$.

Generator For simplicity, we assume the sequence length to be T. Our action generator starts with a RNN with an input at each time as the concatenation of a Gaussian random noise z and an embedded class representation of a label y. The outputs of the RNN layer are denoted as $[o_0, o_1, o_2, ..., o_{T-1}]$. Following [7, 40], we consider outputting residuals instead of the exact coordinates of different joints, *i.e.*, $c_0 = o_0$, $c_1 = o_1 + c_0$, ..., $c_{T-1} = o_{T-1} + c_{T-2}$. The output of the RNN will go through three linear transformations before being fed as the input of the newly proposed *SA-GC* layer, which will be detailed in Section 3.2.

The *SA-GC* layer The key component of our framework is a newly defined self-attention based graph convolutional layers (*SA-GC* layers), as illustrated in Fig. 2. Specifically, we denote the input of the *SA-GC* layers as a feature

vector $\mathbf{H} \in \mathbb{R}^{T\times N}$. Through a self attention layer [33], the output are a new representation $\mathbf{H}_{in} \in \mathbb{R}^{T\times N\times 1}$ and a learned masked attention score matrix $\mathbf{S}_{mask} \in \mathbb{R}^{T\times T}$. This self attention layer is followed by 5 *GC* layers. Each *GC* layer takes last layer's hidden state vector and masked adjacency matrix $\tilde{\mathbf{A}}_{\mathbf{s}}$ as the input. The hidden states, which are outputs of the 5 *GC* layers, are defined respectively as $\mathbf{H}^{(1)} \in \mathbb{R}^{T\times N\times 32}$, $\mathbf{H}^{(2)} \in \mathbb{R}^{T\times N\times 64}$, $\mathbf{H}^{(3)} \in \mathbb{R}^{T\times N\times 64}$, $\mathbf{H}^{(4)} \in \mathbb{R}^{T\times N\times 128}$ and $\mathbf{H}^{(5)} \in \mathbb{R}^{T\times N\times 128}$. Furthermore, the ResNet mechanism [15] is applied on each two *SA-GC* layers, *i.e.*, we add the output of the first *SA-GC* layer to the third *SA-GC* layer, and the output of the third *SA-GC* layer to the final output. Detailed operations of the *SA-GC* layer are described in Section 3.2.

Dual discriminator The video-based discriminator $\mathcal{D}_V$ takes a sequence of actions and the corresponding labels as the input. The frame-based discriminator $\mathcal{D}_F$ randomly selects k_{frame} frames of an input sequence and the corresponding labels as the input. Both discriminators output either real or fake. In this paper, we apply the conditional GAN objective formulation [11, 25, 22]:

$$\begin{aligned}\mathcal{L} = \min_{\mathcal{G}} \max_{\mathcal{D}_F,\mathcal{D}_V} \mathbb{E}_{x\sim p(x)}[\log \mathcal{D}_F(x|y))] + \mathbb{E}_{z\sim p(z)}[\log(1-\mathcal{D}_F(\mathcal{G}(z|y)))]+ \\ \mathbb{E}_{x\sim p(x)}[\log \mathcal{D}_V(x|y)] + \mathbb{E}_{z\sim p(z)}[\log(1-\mathcal{D}_V(\mathcal{G}(z|y)))]\end{aligned} \tag{1}$$

where $p(x)$ defines the ground truth distribution, $p(z)$ is the standard Gaussian distribution and y is the one-hot class indicator.

3.2 Action Graph Construction

In this section, we describe detailed construction of the action graph, which is used in our *SA-GCN* module. Note a skeleton sequence is usually represented

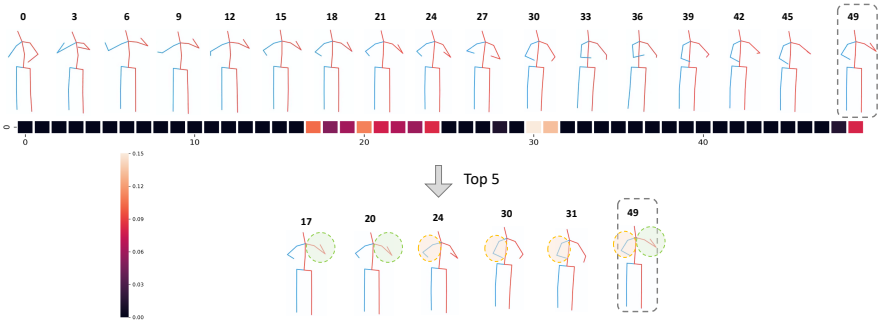

Fig. 4: The pipeline of *SA-GC* layer. The top line shows frames out of every three consecutive frames from Human 3.6 *Direction* class. The heat map under these samples represent the corresponding attention scores for the 49*th* frame. The bottom line shows the top 5 frames with the highest attention scores. Green circles and orange circles show similarity between selected frames and our target frame (the 49*th* frame).

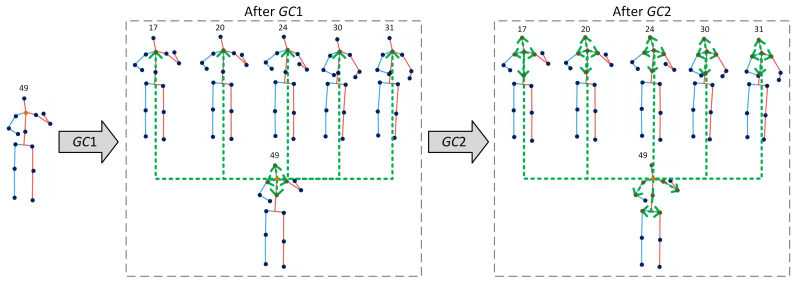

Fig. 5: Information passing through *SA-GC* layers at the node "neck".

by 2D or 3D coordinates of human joints in each frame. The inter-frame action is the static skeleton in spatial space, and the inter-frame action is the movement in temporal space. To capture the temporal correlation, previous work has applied 1D convolution for learning skeleton representing by concatenating coordinate vectors of all joints in one frame. In our framework, as stated before, we propose to construct a connected graph for a whole action sequence, and learn a sparse inter-frame connection by adopting self-attention learning. Particularly, we construct an undirected graph **G=(V, E)** on a whole action sequence of T frames, each consists of N joints. Here, the node set $V = \{v_{ti} | t = 0, \ldots, T-1, i = 1, \ldots N\}$ includes all joints of a skeleton sequence.

Explanation of the *SA-GC* layer Fig. 2 shows the detailed implementation of the *SA-GC* layer. Our *SA-GC* layer consists of one self attention layer and 5 graph convolution (*GC*) layers. To explain the construction, we detail the pipeline of the construction with an example illustrated in Fig. 4.

The Self attention layer Similar to standard self attention [33], our self-attention layer takes a feature vector $\mathbf{H} \in \mathbb{R}^{T \times N}$ as input, and outputs a self-attention matrix $\mathbf{S}_{mask}$, representing how much influence of the past frames on the current frame. Fig. 4 shows one of our generated *Direction* sequences and its corresponding attention score vector's heat map for the last frame (the $49th$ frame). After the self attention layer, we select top 5 past frames with the highest attention scores (only keep 6 elements in each row of the $\mathbf{S}_{mask}$ matrix). As we could see from the example in Fig. 4, the selected 5 past frames have the highest similarity with the $49th$ frame. The skeleton in the $49th$ frame keeps the red arm up and keeps the blue arm bent down. Looking back to past frames, frames before the $21st$ lift up its red arm. Frames between the $24th$ to the $31st$ frame have the similar blue arm pose as the $49th$ frame. Our attention identifies frames $17th$, $20th$, $24th$, $30th$ and $31st$ as the most relevant frames to be attended according to the learned attention matrix $\mathbf{S}_{mask}$.

The GC layers As illustrated, the self-attention layer is followed by 5 *GC* layers. After selection, we will connect each node of the $49th$ frame with the corresponding node in the selected 5 frames and assign edge weights with the corresponding

self-attention scores. Fig. 5 shows information passing path through our *SA-GC* layer at the node neck. The left plot of Fig. 5 shows that after one *GC* layer, the $49th$ neck node can gather information from neck nodes of five selected frames and four neighbor nodes in its own frame. The right plot of Fig. 5 shows that after the two *SA-GC* layers, the $49th$ neck node can gather information for the five nodes of the selected past four frames and seven nodes of its own frame. It is worth noting that nodes in different frame will have distinct attention score for edges in both spatial space and temporal space, thus they will have their particular edge weights through our *SA-GC* layer.

Implementing self-attention based GCN In our case, we consider all joints in an action sequence, ending up with a 2D adjacent matrix with both row size and column size $N*T$. To this end, we first use $\mathbf{A} \in \mathbb{R}^{N\times N}$ to denote the adjacent matrix of intra-frame, which is constructed by strictly following the structure of a skeleton, *e.g.*, the "head" node is connected to the "neck" node. After adding self connections $\mathbf{I}$, the intra-frame adjacency matrix will be $\bar{\mathbf{A}} = \mathbf{A} + \mathbf{I}$. We then define an initial adjacency matrix of a whole sequence as:

$$\tilde{\mathbf{A}} = \begin{pmatrix} \bar{\mathbf{A}} & \mathbf{I} & \cdots & \mathbf{I} \\ \mathbf{I} & \bar{\mathbf{A}} & \cdots & \mathbf{I} \\ \vdots & \vdots & & \vdots \\ \mathbf{I} & \mathbf{I} & \cdots & \bar{\mathbf{A}} \end{pmatrix}_{(N*T)\times(N*T)} , \tag{2}$$

where $\mathbf{I}$ is used to represent connecting each node with all of the corresponding nodes in the temporal space, $(N*T)\times(N*T)$ means $\tilde{\mathbf{A}}$ is a 2D matrix with both row size and column size $N*T$, both N and T are numbers, * means multiply operation. The adjacency matrix $\tilde{\mathbf{A}}$ essentially means each node in one frame is connected to the corresponding node in the temporal space. At the same time, it also connects to the neighboring nodes in spatial space encoded by $\bar{\mathbf{A}}$.

Next, we propose to use self-attention to prune the action graph. The idea is to learn a set of attention scores encoding the relevance of each frame w.r.t. the current frame, and only choose the top-K frames in the temporal space. Specifically, we adopt a similar implementation of the scaled dot-product attention as in [33]. The input of the self-attention layer is represented as $\mathbf{H} \triangleq \{h_0, h_1, \cdots, h_{T-1}\}$, where $h_t \in \mathbb{R}^N$ represents the hidden state vector at time t with N nodes. Following the self-attention in [33], $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are given as:

$$\mathbf{Q} = \mathbf{W}_q\mathbf{H} \, , \; \mathbf{K} = \mathbf{W}_k\mathbf{H} \, , \; \mathbf{V} = \mathbf{W}_v\mathbf{H} \, , \tag{3}$$

where $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v$ are projection weights. The attention score $\mathbf{S} \in \mathbb{R}^{T\times T}$ and the attention layer's output $\mathbf{H}_{in}$ are calculated as:

$$\mathbf{S} = \text{softmax}\left(\mathbf{Q}\mathbf{K}^{\mathsf{T}}\right) ; \; \mathbf{H}_{in} = \mathbf{S}\mathbf{V} \tag{4}$$

In the task of action generation, we need to modify $\mathbf{S}$ as a masked attention $\mathbf{S}_{mask}$ which prevents current frame from attending to subsequent frames

$$\mathbf{S}_{mask} = \begin{pmatrix} s_{0,0} & 0 & \cdots & 0 \\ s_{1,0} & s_{1,1} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ s_{T-1,0} & s_{T-1,1} & \cdots & s_{T-1,T-1} \end{pmatrix}_{T\times T}, \tag{5}$$

where the element $s_{m,n}$ denotes the n-th frame's influence on the m-th frame and values in the upper triangle are all equal to 0. To enforce the pruning, we further select the top K scores in each row of the $\mathbf{S}_{mask}$ and set the other elements to be 0. Note that, if the number of non-zero elements in some rows is less than K, we will keep all the non-zero elements. Finally, the adjacent matrix is constructed as

$$\tilde{\mathbf{A}}_s = \mathbf{S}_{mask} \odot \tilde{\mathbf{A}} \triangleq \begin{pmatrix} s_{0,0} * \bar{\mathbf{A}} & \mathbf{0} & \cdots & \mathbf{0} \\ s_{1,0} * \mathbf{I} & s_{1,1} * \bar{\mathbf{A}} & \cdots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ s_{T-1,0} * \mathbf{I} & s_{T-1,1} * \mathbf{I} & \cdots & s_{T-1,T-1} * \bar{\mathbf{A}} \end{pmatrix}_{(N*T)\times(N*T)} \tag{6}$$

Consequently, the output (before activation) of the self-attention based graph convolutional layer becomes:

$$\mathbf{H}^{(1)} = \mathbf{D}^{-1}\tilde{\mathbf{A}}_s\mathbf{H}_{in}\mathbf{W}\,, \tag{7}$$

where $\mathbf{D}^{ii} = \sum_j \tilde{\mathbf{A}}_s^{ij}$ represents diagonal node degree matrix for normalizing $\tilde{\mathbf{A}}_s$, $\mathbf{H}^{(1)}$ is the hidden state after first **GC** layer in Fig. 2. We will conduct graph convolution operation in equation 7 using same $\tilde{\mathbf{A}}_s$ for five times. The output of the fifth *GC* layer in *SA-GC* layer is $\mathbf{H}^{(5)}$. After a linear layer, we get the output of the generator, which is a generated sequence $x \in \mathbb{R}^{T\times N}$.

4 Experiments

We perform experiments to evaluate the proposed method on two standard skeleton-based human-action benchmarks, the Human-3.6m dataset [16] and the NTU RGB+D dataset [29]. Several state-of-the-art methods are used for comparison, including [42, 13, 7, 40]. Following [40], the Maximum Mean Discrepancy (MMD) [12] is adopted to measure the quality of generated actions. Further, we pre-train a classifier on training set to test the recognition accuracy of generated actions. We also conduct human evaluation on the Amazon Mechanical Turk (AMT) to access the perceptual quality of generated sequences. To examine the functionality of each component of the proposed model, we also perform detailed ablation studies on the Human-3.6m dataset.

4.1 Datasets

Human-3.6m Following the same pre-processing procedure in [7, 40], 50 Hz video frames are down-sampled to 16 Hz to obtain representative and larger variation 2D human motions. The joint information consists of 2-D locations of 15 major body joints. Ten distinctive classes of actions are used in the following experiments, including *sitting, sitting down, discussion, walking, greeting, direction, phoning, eating, smoking* and *posing*.

NTU RGB+D This dataset contains 56,000 video clips on 60 classes performed by 40 subjects and recorded with 3 different camera views. Compared with Human-3.6m, it can provide more samples in each class and much more intra-class variations. We select ten classes of motions and obtain their 2-D coordinates of 25 body joints following the same setting in [40], including *drinking water, jump up, make phone call, hand waving, standing up, wear jacket, sitting down, throw, cross hand in front* and *kicking something*. We then apply two commonly used benchmarks for a further evaluation in the ten classes: (*i*)*cross-view*: the training set contains actions captured by two cameras and remaining data are left for testing. (*ii*)*cross-subject*: action clips performed by 20 subjects are randomly picked for training and another 20 subjects are reserved for testing.

4.2 Training Details

Following [40], we set the action sequence length for both datasets to be 50. The image discriminator randomly selects 20 frames from every generated sequence and training sequences as the input. The *SA-GC* layer selects top 5 past frames to construct an adjacency matrix $\tilde{\mathbf{A}}_\mathbf{s}$. We set batch size for training to be 100, for testing to be 1000, and the learning rate to be 0.0002.

4.3 Evaluation Metrics

Maximum Mean Discrepancy The MMD metric is based on a two-sample test to measure the similarity between two distributions $\mathcal{P}(x)$ and $\mathcal{Q}(y)$, based on samples $x \sim \mathcal{P}(x)$ and $y \sim \mathcal{Q}(y)$. It is widely used to measure the quality of generated samples compared with real data in deep generative model [49] and Bayesian sampling [14]. The metric has also been applied to evaluate the similarity between generated actions and the ground truth in [34, 40], which has been proved consistent with human evaluation. As motion dynamics are in the form of sequential data points, we denote MMD_{avg} as the average MMD over each frame and MMD_{seq} to denote the MMD over whole sequences.

Recognition Accuracy Apart from using MMD to evaluate the model performance, we also pre-train a recognition network on the training data to compute the classification accuracy of generated samples. The recognition network exactly follows the video discriminator except for the last *softmax* layer. This evaluation metrics can examine whether the conditional generated samples are actually residing in the same manifold as the ground truth and can be correctly recognized. Details are given in the Appendix.

4.4 Baselines

We compare our method with six baselines. We first consider the model in [42], which can be used to generate long-term skeleton-based actions in an end-to-end manner. This includes three training alternatives: end-to-end (*E2E*), E2E prediction with visual analogy network (*EPVA*) and EPVA with adversarial loss (*adv-EPVA*). The second baseline [13] is based on VAE, called the *SkeletonVAE*, which improves previous motion generation methods significantly. Finally, two most recent strong baselines are considered, including the previous state-of-the-art method [7] and an improved version [40] with an auxiliary classifier. The latter utilizes a *Single Pose Training* stage and a *Pose Sequence Generation* stage to produce high-quality motions. These two baselines are respectively referred to as *SkeletonGAN* and *c-SkeletonGAN*.

4.5 Detailed Results

Quantitative results Our *SA-GCN* model shows superior quantitative results in terms of both MMD and recognition accuracies on the two datasets, compared with related baseline models.

Human-3.6m Table 1 shows MMD results of our model and the baselines on Human-3.6m. With structure information considered, our model achieves significant performance gains over all baselines, which even without the need of an inefficient pre-training stage. The recognition accuracies are reported in Table 2. Similarly, our model consistently outperforms three baselines by a large margin. Please note none information of the generated actions are used in the pretrained classifier, thus we avocate that the relatively low recognition accuracies are indeed reasonable. On the other hand, this also indicates that existing action generation models are still far from satisfactory.

NTU RGB+D This dataset is more challenging, which contains more body joints and action variations. In the experiments, we find that three models (E2E, EPVA, adv-EPVA [42]) fail to generate any interpretable action sequences. As a result, we only present MMD results for the other three baselines in Table 3. Again, the proposed method performs the best among all models under *cross-view* and *cross-subject* settings.

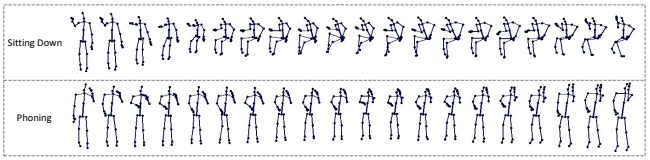

Fig. 6: Randomly selected samples on NTU RGB+D dataset. Top: *sitting down* from *cross-subject*, Bottom: *phoning* from *cross-view*.

Table 1: Model comparisons in terms of MMD on Human-3.6m.

Models	Pretrain	$\text{MMD}_{\text{avg}} \downarrow$	$\text{MMD}_{\text{seq}} \downarrow$
E2E [42]	No	0.991	0.805
EPVA [42]	No	0.996	0.806
adv-EPVA [42]	No	0.977	0.792
SkeletonVAE [13]	No	0.452	0.467
SkeletonGAN [7]	Yes	0.419	0.436
c-SkeletonGAN [40]	Yes	0.195	0.218
Ours	No	**0.146**	**0.134**

Table 2: Action recognition accuracy on the generated actions on the Human-3.6m.

Models	Direct	Discuss	Eat	Greet	Phone	Pose	Sit	SitD	Smoke	Walk	Average
SkeletonVAE	0.37	0.01	0.51	0.47	0.10	0.03	0.17	0.33	0.01	0.01	0.201
SkeletonGAN	0.35	0.29	0.72	0.66	0.46	0.09	0.32	0.71	0.14	0.02	0.376
c-SkeletonGAN	0.34	0.44	0.57	0.56	0.52	0.25	0.67	1.00	0.50	0.03	0.488
SA-GCN	0.42	0.40	0.78	0.55	0.72	0.61	0.95	0.79	0.52	0.18	**0.593**

Table 3: Model comparisons in terms of MMD on NTU RGB+D.

Models	*cross-view*		*cross-subject*	
	$\text{MMD}_{\text{avg}} \downarrow$	$\text{MMD}_{\text{seq}} \downarrow$	$\text{MMD}_{\text{avg}} \downarrow$	$\text{MMD}_{\text{seq}} \downarrow$
SkeletonVAE [13]	1.079	1.205	0.992	1.136
SkeletonGAN [7]	0.999	1.311	0.698	0.788
c-SkeletonGAN [40]	0.371	0.398	0.338	0.402
SA-GCN	**0.316**	**0.335**	**0.285**	**0.299**

Qualitative results We present some generated actions in Human-3.6m dataset and NTU RGB+D dateset in Fig. 7(first and third row) and Fig. 6 respectively. It is easy to see that our model can generate very realistic and easily recognizable actions. We also plot action trajectories on a projected space by t-SNE [23] for each generated action class on the Human-3.6m dataset in Fig. 9. It is observed that a group of actions, *i.e.*, *directions, discussion*, *greeting*, are close to each other, and so is the group *sitting, sitting down*, *eating*; while actions *smoking* and *sitting down* are far away. These are consistent with what we have observed in the ground truth.

Smooth action generation Humans are capable of switching two actions very smoothly and naturally. For instance, a person can show others directions and walking at the same time. In this part, we verify that our model is expressive enough to perform such transitions as humans do. We use (8) to produce a smooth action transition between action classes y_1 and y_2 with a smoothing parameter $\lambda \in [0, 1]$. We generate 100 video clips with every mix and apply t-SNE [23] to project the averaged sequences to a 1D manifold. The histogram of various mixed actions is shown in Fig. 8. As we decrease λ, the mode (action) gradually moves from *directions* towards *walking*, meaning that our model can produce very smooth transitions when interpolating between the two actions.

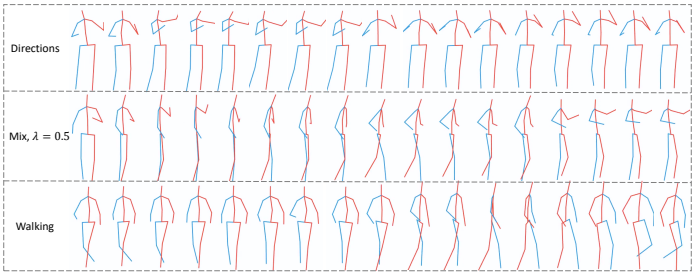

Fig. 7: Generated sequences of *directions*, *walking* and a mixed action with $\lambda = 0.5$.

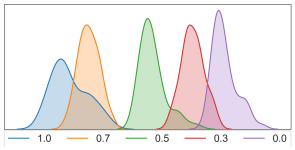

Fig. 8: Histogram of mixed actions where each mode represents an action with a smoothing term λ.

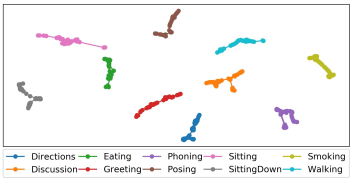

Fig. 9: Action trajectories on Human-3.6m.

Fig. 7 illustrates as randomly selected samples.

$$y_{mix} = \lambda y_1 + (1-\lambda)y_2; \quad x_{mix} = G(z; y_{mix}),\ z \sim \mathcal{N}(0,1) \tag{8}$$

4.6 Ablation Study

Our key innovation in our model is the *SA-GC* layer. As a result, we conduct detailed experiments to verify the effectiveness and usefulness of our self-attention based graph convolutional layer on Human 3.6m dataset. Since the self-attention layer has already been proved to be effective for sequential data, we keep the self-attention layer for all the following baselines. Without special mentioning, we keep all the other parts of the model to be the same.

Baseline 1: replace GCN layers with CNN layers We replace 5 GCN layers with 5 CNN layers using the same hidden dimension and kernel size.

Baseline 2: without the inter-frame A matrix Based on our model, we drop the attention connections to past frames. That setting is the same as setting our top **k** to be 0 in our *SA-GC* layer. Under this Baseline, each frame in the sequence will be an independent graph for graph convolutional layer.

Baseline 3: replace self-attention based GCN layers with the ST-GC layers [44] The *ST-GC* layer leverage graph convolution for skeleton-based action recognition. Each *ST-GC* layer combines one graph convolution layer for learning

intra-frame skeleton and one 1D convolutional layer for feature aggregation in the temporal space.

The *Fully Connected* model described in Fig. 1 is not applicable and can not scale to long sequences because it demands excessive amount of memory and computational resources. The results of above three baselines are shown in Table 4. Comparing with baseline2 and baseline3, we can see that adding the adjacency matrix makes the model harder to train compared with CNN. However, our proposed self-attention can mitigate the difficulties and surpass standard CNN method on the skeleton based action generation task with much lower MMD scores.

Table 4: Ablation study results.

Baselines	$\text{MMD}_{\text{avg}} \downarrow$	$\text{MMD}_{\text{seq}} \downarrow$
Baseline 1	0.240	0.222
Baseline 2	0.915	0.922
Baseline 3	0.580	0.595
Ours	**0.152**	**0.142**

Table 5: AMT Evaluations

Models	Evaluation Score↑
SkeletonVAE	2.401
SkeletonGAN	2.731
c-SkeletonGAN	3.157
SA-GCN	**3.925**

4.7 Human Evaluation

We finally conduct perceptual human evaluations in the AMT platform. Four models are trained on the Human-3.6m dataset, including *SkeletonVAE*, *SkeletonGAN*, *c-SkeletonGAN* and our *SA-GCN*. We then sample 100 action clips for each of the 10 action classes; 140 workers were asked to evaluate the quality of the generated sequences and score them in a range from 1 to 5. A higher score indicates a more realistic action clip. We only inform them of the action class and one real action clip to ensure proper judgements. The design detail is given in the Appendix. Table 5 demonstrates that our model is significantly better than other baselines in human evaluation.

5 Conclusions

In this paper, we have presented the self-attention graph convolutional network (*SA-GCN*) to efficiently encode structure information into skeleton-based human action generation. Self-attention can capture long-range dependencies in continuous action sequences and learn to prune the dense action graph for efficient training. Further, the graph convolution is applied to seamlessly encode both spatial joints information and temporal dynamics information into the model. Based on these ideas, our model directly transforms noises to high-quality action sequences and can be trained end-to-end. On two standard human action datasets, we observe a significant improvement of generation quality in terms of both quantitative and qualitative evaluations.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Barsoum, E., Kender, J., Liu, Z.: Hp-gan: Probabilistic 3d human motion prediction via gan (2017), https://arxiv.org/abs/1711.09561
3. Barsoum, E., Kender, J., Liu, Z.: Hp-gan: Probabilistic 3d human motion prediction via gan. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1418–1427 (2018)
4. Bissacco, A., Soatto, S.: Hybrid dynamical models of human motion for the recognition of human gaits. International journal of computer vision **85**(1), 101–114 (2009)
5. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)
6. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
7. Cai, H., Bai, C., Tai, Y.W., Tang, C.K.: Deep video generation, prediction and completion of human action sequences. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 366–382 (2018)
8. Chen, X., Mishra, N., Rohaninejad, M., Abbeel, P.: Pixelsnail: An improved autoregressive generative model. arXiv preprint arXiv:1712.09763 (2017)
9. Clark, A., Donahue, J., Simonyan, K.: Efficient video generation on complex datasets. arXiv preprint arXiv:1907.06571 (2019)
10. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: CVPR (2015)
11. 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)
12. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. Journal of Machine Learning Research **13**(Mar), 723–773 (2012)
13. Habibie, I., Holden, D., Schwarz, J., Yearsley, J., Komura, T., et al.: A recurrent variational autoencoder for human motion synthesis. (2017)
14. Han, J., Liu, Q.: Stein variational gradient descent without gradient. arXiv preprint arXiv:1806.02775 (2018)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
16. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence **36**(7), 1325–1339 (2013)
17. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: A new representation of skeleton sequences for 3d action recognition. In: CVPR (2017)
18. Kiasari, M.A., Moirangthem, D.S., Lee, M.: Human action generation with generative adversarial networks. arXiv preprint arXiv:1805.10416 (2018)
19. Kim, T.S., Reiter, A.: Interpretable 3d human action analysis with temporal convolutional networks. In: BNMW CVPR (2017)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

21. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
22. Li, C., Liu, H., Chen, C., Pu, Y., Chen, L., Henao, R., Carin, L.: Alice: Towards understanding adversarial learning for joint distribution matching. In: Advances in Neural Information Processing Systems. pp. 5495–5503 (2017)
23. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
24. Martinez, J., Black, M.J., Romero, J.: On human motion prediction using recurrent neural networks. In: CVPR (2017)
25. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
26. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: International conference on machine learning. pp. 2014–2023 (2016)
27. Oh, S.M., Rehg, J.M., Balch, T., Dellaert, F.: Learning and inference in parametric switching linear dynamic systems. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. vol. 2, pp. 1161–1168. IEEE (2005)
28. Pavlovic, V., Rehg, J.M., MacCormick, J.: Learning switching linear models of human motion. In: Advances in neural information processing systems. pp. 981–987 (2001)
29. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1010–1019 (2016)
30. Taylor, G.W., Hinton, G.E., Roweis, S.T.: Two distributed-state models for generating high-dimensional time series. Journal of Machine Learning Research **12**(Mar), 1025–1068 (2011)
31. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: Mocogan: Decomposing motion and content for video generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1526–1535 (2018)
32. Urtasun, R., Fleet, D.J., Geiger, A., Popović, J., Darrell, T.J., Lawrence, N.D.: Topologically-constrained latent variable models. In: Proceedings of the 25th international conference on Machine learning. pp. 1080–1087 (2008)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
34. Walker, J., Marino, K., Gupta, A., Hebert, M.: The pose knows: Video forecasting by generating pose futures. In: Proceedings of the IEEE international conference on computer vision. pp. 3332–3341 (2017)
35. Wang, J.M., Fleet, D.J., Hertzmann, A.: Optimizing walking controllers. In: ACM SIGGRAPH Asia 2009 papers, pp. 1–8 (2009)
36. Wang, J.M., Fleet, D.J., Hertzmann, A.: Optimizing walking controllers for uncertain inputs and environments. ACM Transactions on Graphics (TOG) **29**(4), 1–8 (2010)
37. Wang, T.C., Liu, M.Y., Zhu, J.Y., Liu, G., Tao, A., Kautz, J., Catanzaro, B.: Video-to-video synthesis. In: NeurIPS (2018)
38. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)
39. Wang, Y., Gui, L.Y., Liang, X., Moura, J.M.F.: Adversarial geometry-aware human motion prediction. In: ECCV (2018)

40. Wang, Z., Yu, P., Zhao, Y., Zhang, R., Zhou, Y., Yuan, J., Chen, C.: Learning diverse stochastic human-action generators by learning smooth latent transitions. arXiv preprint arXiv:1912.10150 (2019)
41. Wang, Z., Chai, J., Xia, S.: Combining recurrent neural networks and adversarial training for human motion synthesis and control. IEEE transactions on visualization and computer graphics (2019)
42. Wichers, N., Villegas, R., Erhan, D., Lee, H.: Hierarchical long-term video prediction without supervision. arXiv preprint arXiv:1806.04768 (2018)
43. Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.i., Jegelka, S.: Representation learning on graphs with jumping knowledge networks. arXiv preprint arXiv:1806.03536 (2018)
44. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI conference on artificial intelligence (2018)
45. Yan, Y., Xu, J., Ni, B., Zhang, W., Yang, X.: Skeleton-aided articulated motion generation. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 199–207 (2017)
46. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: International Conference on Machine Learning. pp. 7354–7363 (2019)
47. Zhang, Y., Qi, P., Manning, C.D.: Graph convolution over pruned dependency trees improves relation extraction. arXiv preprint arXiv:1809.10185 (2018)
48. Zhao, Y., Li, C., Yu, P., Gao, J., Chen, C.: Feature quantization improves GAN training. ICML (2020)
49. Zhao, Y., Zhang, J., Chen, C.: Self-adversarially learned bayesian sampling. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5893–5900 (2019)
50. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434 (2018)

A Experiments Results

We further show some action samples on both the Human-3.6m dataset [16] and the NTU RGB+D dataset [29]. We sample one frame in terms of every two consecutive frames to show the whole sequence of actions.

Human-3.6m We show ten classes of action sequences: *direction, discussion, eating, greeting, phoning, posing, sitting, sitting down, smoking* and *walking* in Fig. 10, Fig. 11, Fig. 12, Fig. 13, Fig. 14, Fig. 15, Fig. 16, Fig. 17, Fig. 18 and Fig. 19 on Human-3.6m dataset. For each action class, we present three generated action sequences from random initialization.

NTU RGB+D We show ten classes of action sequences: *drinking water, jumping up, kicking something, making phone call, sitting down, standing up, throwing, hand waving, wearing jacket* and *crossing hand in front* in Fig. 20, Fig. 21, Fig. 22, Fig. 23, Fig. 24, Fig. 25, Fig. 26, Fig. 27, Fig. 28 and Fig. 29 on NTU RGB+D dataset. For each action class, we present two generated action sequences (the top two lines) for *cross-view* and two generated action sequences (the bottom two lines) for *cross-subject* from random initialization.

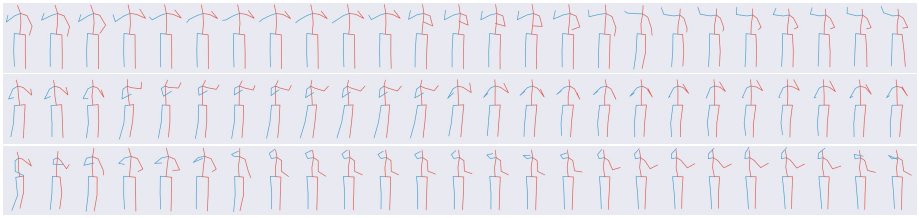

Fig. 10: *direction*: this character is directing traffic.

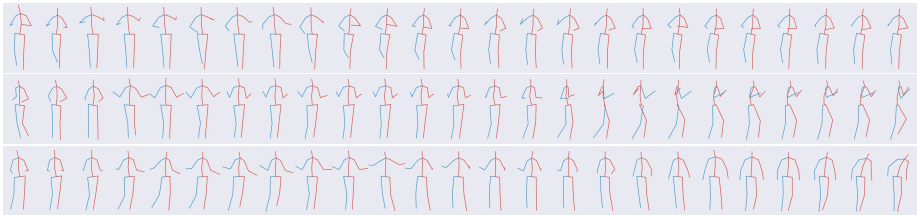

Fig. 11: *discussion*: this character is discussing issues with others.

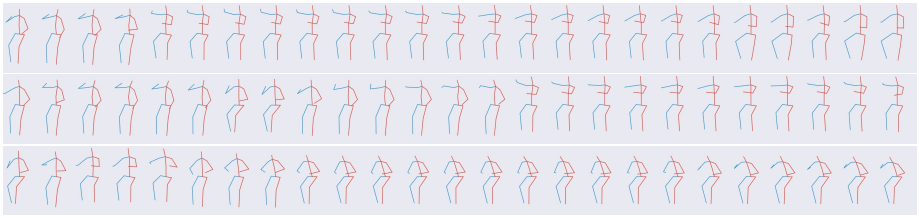

Fig. 12: *eating*: this character is sitting on the chair and having its lunch.

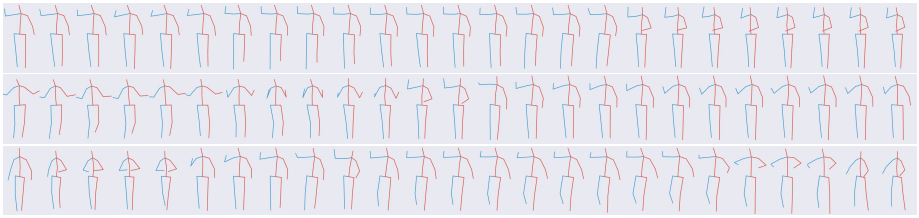

Fig. 13: *greeting*: this character is waving hands and greeting with other people.

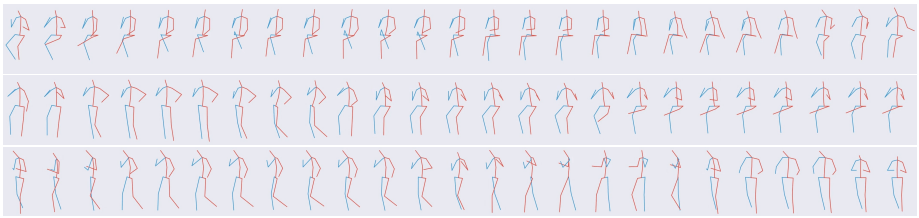

Fig. 14: *phoning*: this character is making a phone call with other people.

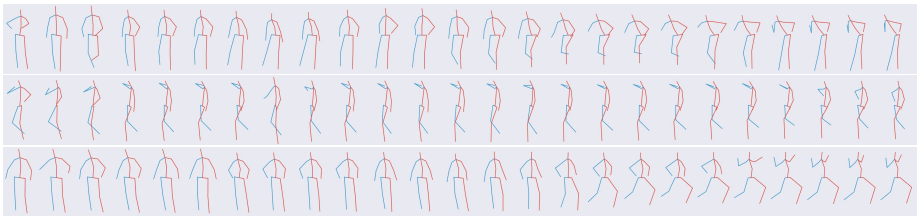

Fig. 15: *posing*: this character is making some exaggerated poses to take photos.

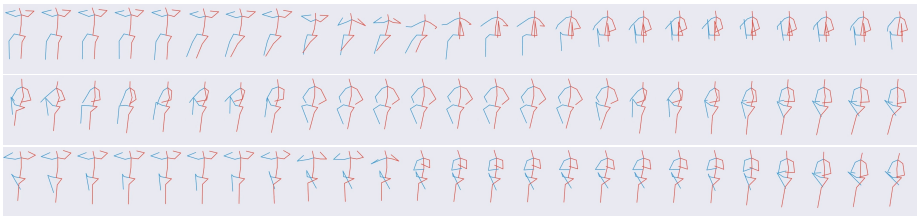

Fig. 16: *sitting*: this character is sitting down on a chair.

Fig. 17: *sitting down*: this character is sitting down on the ground.

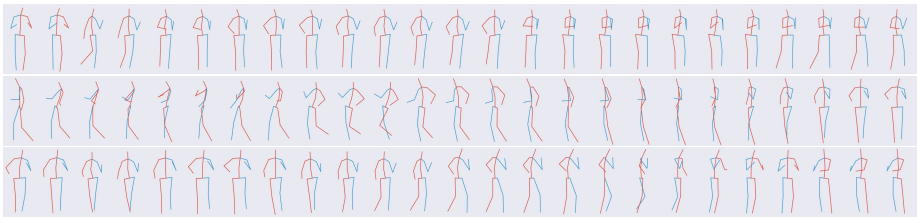

Fig. 18: *smoking*: this character is holding a cigarette in one hand and occasionally smokes.

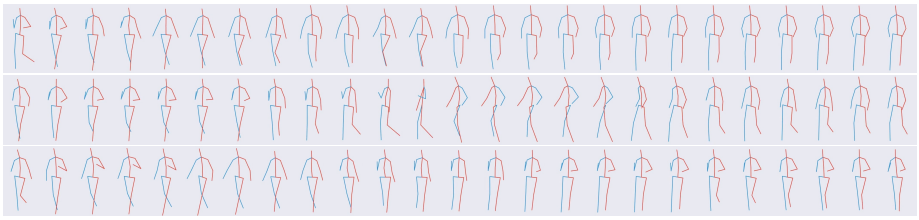

Fig. 19: *walking*: this character is walking.

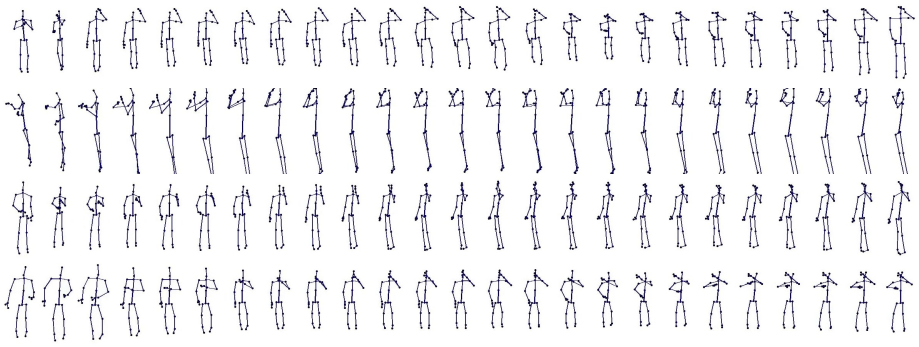

Fig. 20: *drinking water*: this character is holding a water bottle in one hand while drinking water.

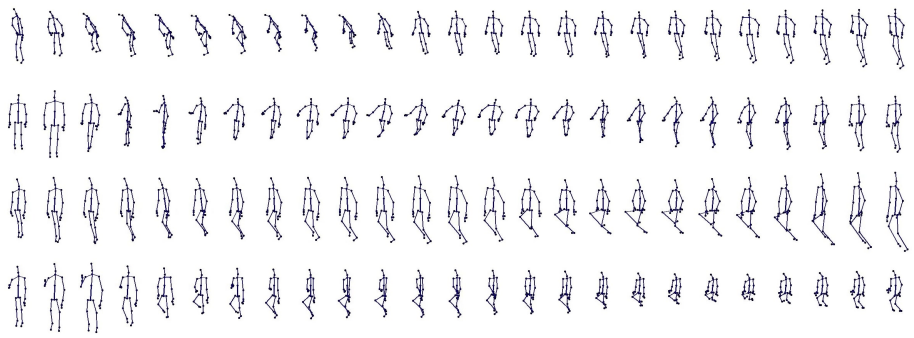

Fig. 21: *jumping up*: this character is jumping.

Fig. 22: *kicking something*: this character is kicking something.

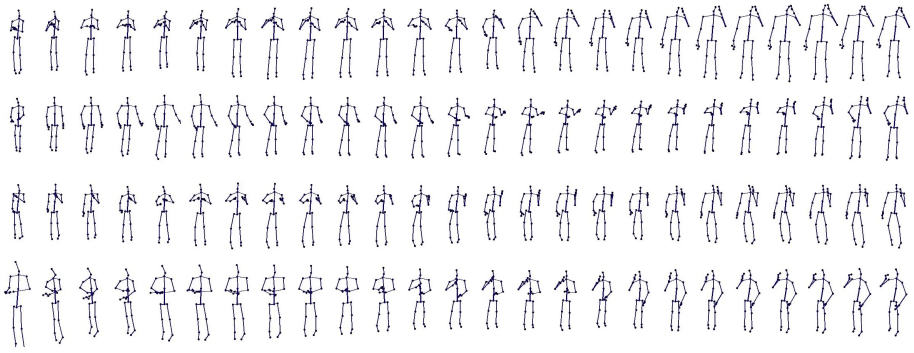

Fig. 23: *making phone call*: this character is raising his mobile phone with one hand and is making a phone call.

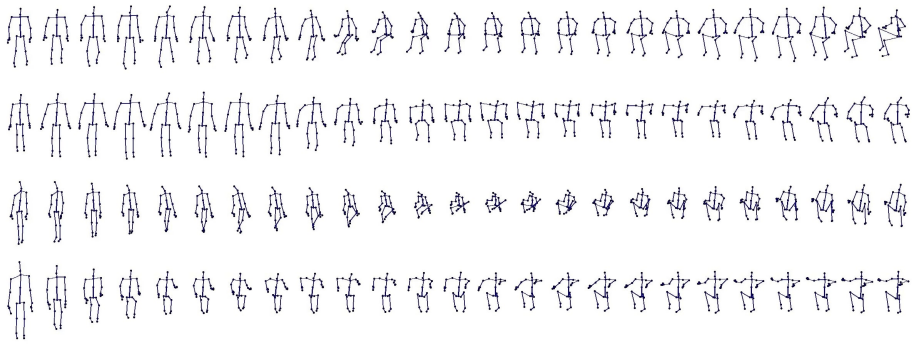

Fig. 24: *sitting down*: this character is sitting down.

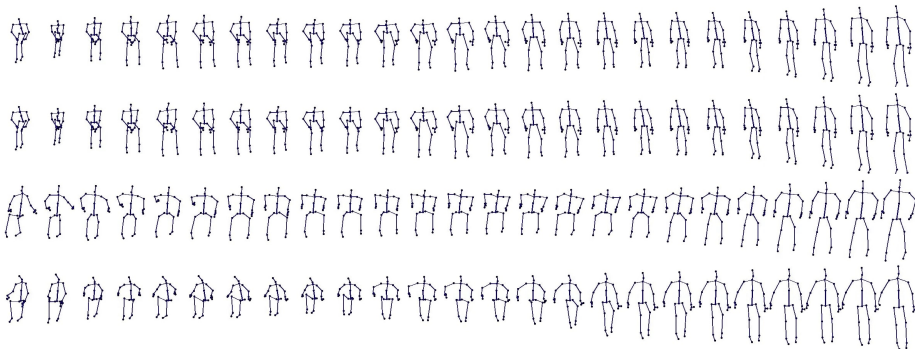

Fig. 25: *standing up*: this character is standing up.

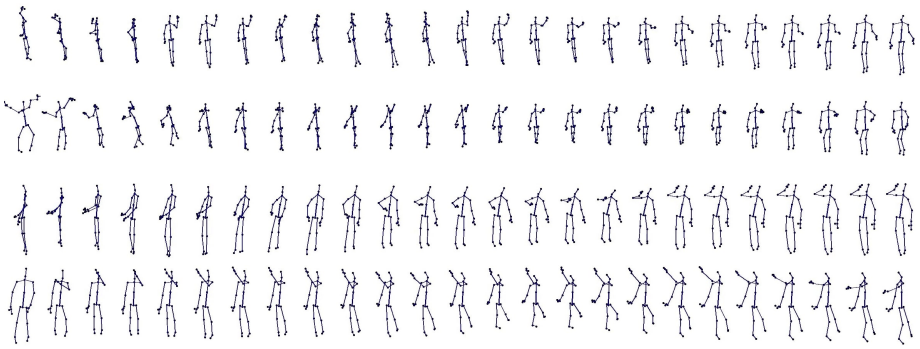

Fig. 26: *throwing*: this character is throwing a ball.

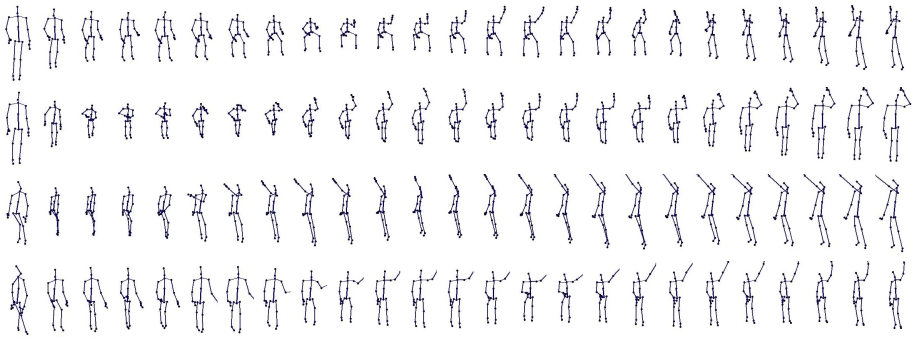

Fig. 27: *hand waving*: this character is waving hands.

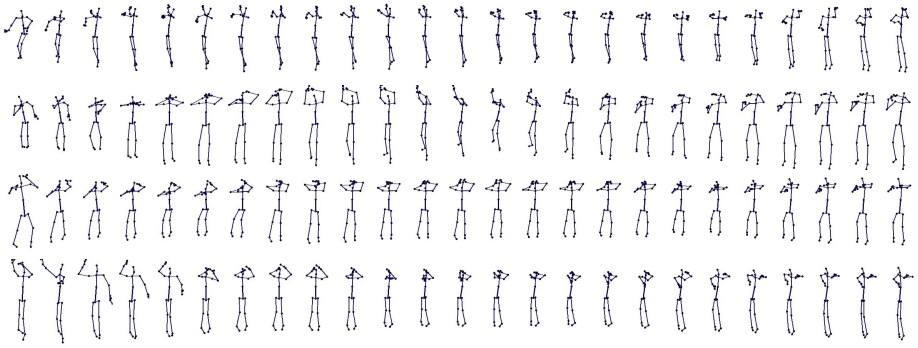

Fig. 28: *wearing jacket*: this character is wearing jacket with its two arms.

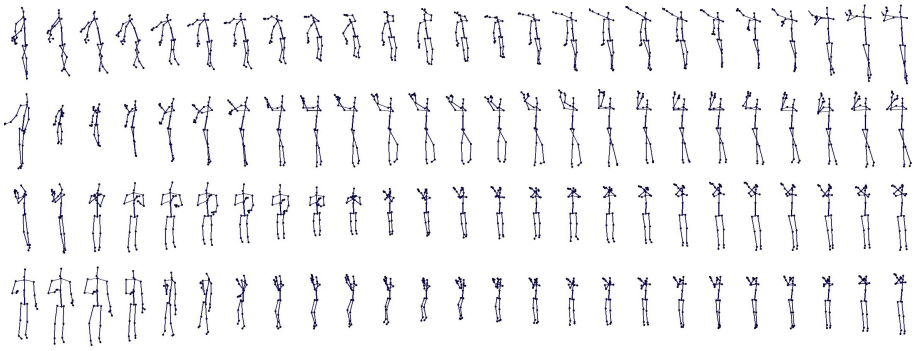

Fig. 29: *crossing hand in front*: the final position for this action is making this character's arm crossing in front.