Generative Adversarial Graph Convolutional Networks for Human Action Synthesis

Bruno Degardin[1,4,5], João Neves[2,4], Vasco Lopes[2,4,5], João Brito[5], Ehsan Yaghoubi[3], Hugo Proença[1,4]

[1]IT - Instituto de Telecomunicações, [2]NOVA LINCS, [3]C4-Cloud Computing Competence Center
[4]Universidade da Beira Interior, Portugal   [5]DeepNeuronic

`bruno.degardin@ubi.pt`

Abstract

Synthesising the spatial and temporal dynamics of the human body skeleton remains a challenging task, not only in terms of the quality of the generated shapes, but also of their diversity, particularly to synthesise realistic body movements of a specific action (action conditioning). In this paper, we propose Kinetic-GAN, a novel architecture that leverages the benefits of Generative Adversarial Networks and Graph Convolutional Networks to synthesise the kinetics of the human body. The proposed adversarial architecture can condition up to 120 different actions over local and global body movements while improving sample quality and diversity through latent space disentanglement and stochastic variations. Our experiments were carried out in three well-known datasets, where Kinetic-GAN notably surpasses the state-of-the-art methods in terms of distribution quality metrics while having the ability to synthesise more than one order of magnitude regarding the number of different actions. Our code and models are publicly available at `https://github.com/DegardinBruno/Kinetic-GAN`.

1. Introduction

Human behaviour analysis through skeleton-based data has been widely investigated for decades. The advent of deep learning-based architectures increased its popularity even more, mainly due to the robustness of skeleton data in handling dynamic circumstances, appearance variations, and cluttered backgrounds. Over the last decade, the rise of data-driven approaches highly correlates performance with the scale of the learning set. Hence, generating high-quality synthetic human actions can address the problem of limited data. However, existing methods are still severely limited, particularly in conditioning desirable actions and considering the generation at the global movement level.

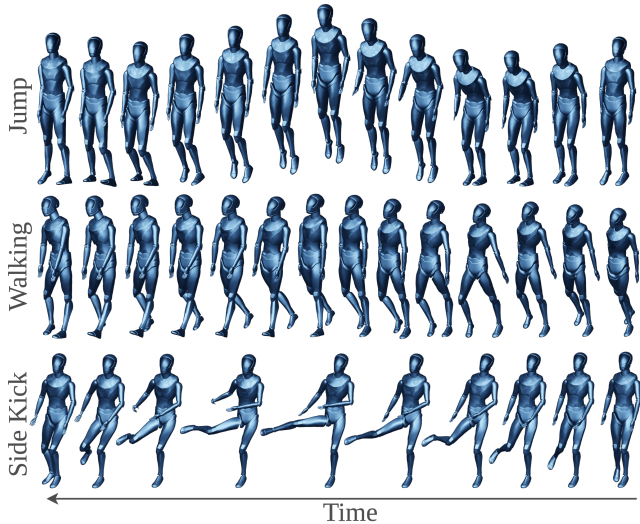The existing skeleton-based human action synthesis algorithms are classified into two categories: autoregres-

Figure 1: **Synthetic set of actions** generated by our graph convolutional generator trained on NTU RGB+D [31] (first two rows) and NTU-120 RGB+D [25] (last row). Kinetic-GAN is able to generate up to 120 different actions even under global movement settings. See accompanying video.

sive and generative approaches. Autoregressive approaches [10, 23, 48] are usually based on Recurrent Neural Networks (RNNs) and consider the skeleton data as a vector sequence to model an action from several seen frames. Despite the decent quality of the individual samples, autoregressive approaches have two disadvantages: (1) Extracting inherent structural body information using vectorized skeleton sequences is suboptimal. (2) The use of LSTMs [16], GRU [1], and Seq2Seq [36] potentially limits the scalability regarding the *bidirectional temporal dependency*, i.e., a future frame modifying the past ones, and the increased difficulty of a past frame conditioning a distant future frame. To solve the latter drawback, generative approaches such as [2, 4, 42, 44] use the concept of Generative Adversarial Networks [12] (GANs) to produce an entire body skeleton sequence from a latent space. However, this category also has
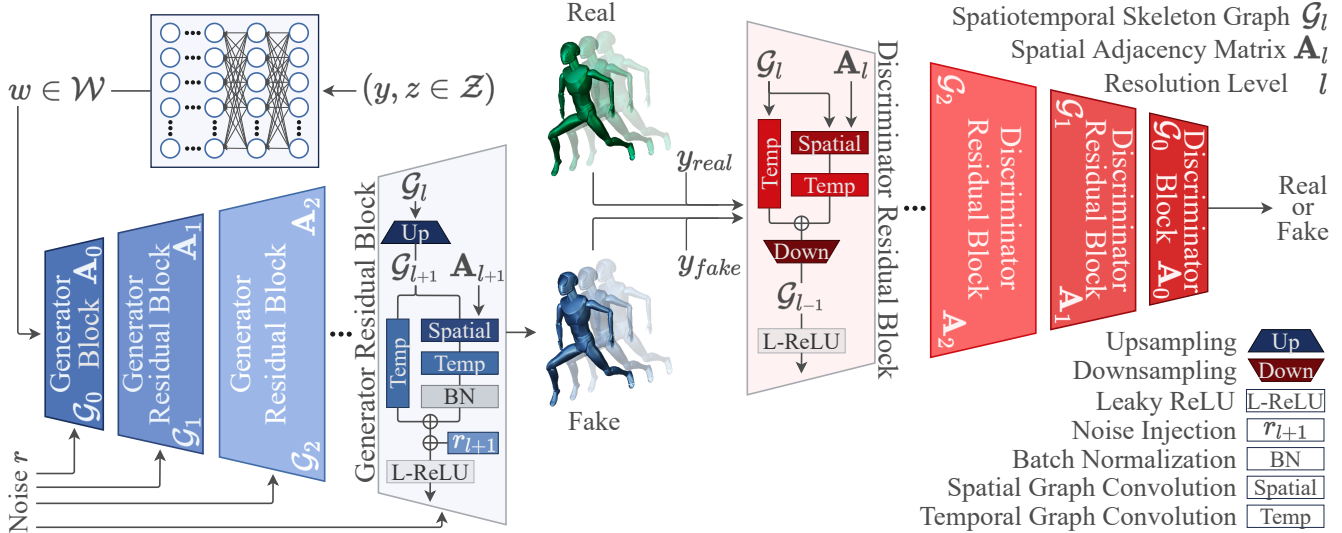
Figure 2: **Cohesive view of the proposed Kinetic-GAN**. The blue and red components belong to the generator and discriminator, respectively. First, both the Gaussian random noise z and the embedded action class representation y are concatenated and mapped to an intermediate latent space $\mathcal{W}$, which is fed to the generator and upsampled spatially and temporally through each resolution level l. The discriminator receives a skeleton graph sequence $\mathcal{G}_l$ together with its embedded class (channel-wise concatenation) and learns to discriminate by downsampling spatially (coarsening) and temporally from level l to 0.

some shortcomings: (1) most approaches still employ manually structured vector sequences to model skeleton data, and (2) they rely on autoregressive techniques and Gaussian processes to solve long-term relationships over the latent space, which greatly limits their scalability in action conditioning.

This paper proposes a Generative Adversarial Graph Convolutional Network (Kinetic-GAN) to address the above-mentioned limitations. Our architecture leverages the benefits of GANs and Graph Convolutional Networks (GCNs), such that we generate conditioned human action sequences directly from the latent space while maintaining the long-term relationships between frames.

Inspired by the generalization of convolutions from images to graphs, we use spatiotemporal graph convolutions in the generator and discriminator to model skeleton data and exploit the skeleton's inherent graph structure, rather than manually structuring them as coordinate vector sequences. Also, the proposed approach improves the state-of-the-art in action conditioning, controlling up to 120 different actions (examples in Fig. 1), whereas previous methods could only control around 10 actions [14, 41, 44]. Additionally, the generation of human actions directly from the latent space may inspire some future directions in human behaviour analysis, such as interpretable latent space directions [32, 39] or style transfer between samples [17, 19, 20]. Fig. 2 illustrates the overview of the proposed framework.

In summary, our main contributions are three-fold: 1) A new scalable Generative Adversarial Graph Convolutional Network architecture to synthesise human actions. 2) An architecture that can be extended to a conditional model, generating desirable actions up to 120 different classes. 3) We perform extensive experiments on three datasets, NTU RGB+D [31], NTU-120 RGB+D [25] and Human3.6M [18], where Kinetic-GAN exceeds the state-of-the-art performance by a significant margin.

2. Related Work

Human action synthesis regards the generation of understandable spatial and temporal kinematics of the human body skeleton. Current methods extract structural and dynamic patterns from either manually structured sequences or graph-based structures. Those representations are then used to synthesise actions via either autoregressive (to learn temporal dependencies) or generative models (to learn a probability distribution).

2.1. Skeleton-based Behaviour Analysis

Body pose estimation is one of the auspicious cues in human behaviour analysis. This semantically rich and very descriptive representation of human dynamics attenuates appearance noises that RGB and depth data contain, driving the learning process solely over human behaviour.

Over the last decade, skeleton-based behaviour analysis has evolved from pseudo-images with CNNs [21, 24, 28, 35] and sequence coordinate vectors with RNNs [9, 26, 27, 34, 45], to the solid improvements of GCNs [6, 7, 33, 43, 47, 46] which models skeleton data as a spatiotemporal graph, which better represents the embedded

structural information. Still, most current methods [41, 44] employ manually structured sequence coordinate vectors.

2.2. Autoregressive Models

Inspired by action prediction models, some works [10, 48] employ autoregressive algorithms to generate human actions from several seen frames. Fragkiadaki *et al.* [10] proposed to incorporate an encoder-decoder network pre- and post-LSTM-units, capturing the temporal dependencies directly from the low-dimensional representation of the input skeleton. Zhou *et al.* [48] presented a conditioned LSTM network, in which the generated data were conditioned at regular intervals of the sequence.

2.3. Generative Models

Generative adversarial networks [12] inspired Generative-based human action synthesis methods [23, 41, 42, 44], whereas some approaches are yet attached to autoregressive techniques [23, 41, 44] or Gaussian processes [42], which limits their temporal flexibility, stochastic variation, and action conditioning. Kundu *et al.* [23] proposed a hierarchical feature fusion based on an RNN auto-encoder architecture with a manually structured tree of limb connections. Yu *et al.* [44] suggested using a graph convolutional network on top of an RNN to analyze the latent temporal dependencies. However, this architecture has a limitation regarding temporal length (50 frames) and action conditioning (10 actions). Yan *et al.* [42] proposed a Gaussian process prior to the generator to analyze the dimensions of the latent space one by one (1024 dimensions). Despite reporting temporal long-term latent relations, its heavy computation on latent points correlation limits their diversity and action conditioning. This paper proposes a novel approach that generates long-term human actions without handcrafted procedures (sequence coordinate vectors or Gaussian process priors), enabling us to synthesise far more different actions with significantly more quality even with higher temporal lengths.

3. Proposed Method

Kinetic-GAN aims to improve the generation quality of controllable action samples while increasing intra-action diversity through stochastic variation and latent space disentanglement. This section presents the proposed approach by initially introducing the GCN adopted, followed by network description, and the discussion of the improvements caried out in the generator, discriminator, and adversarial loss.

3.1. Graph Convolutional Networks Preliminaries

Similar to image modelling with GANs, the proposed solution employs a generator and discriminator to perform upsampling and downsampling on samples, respectively.
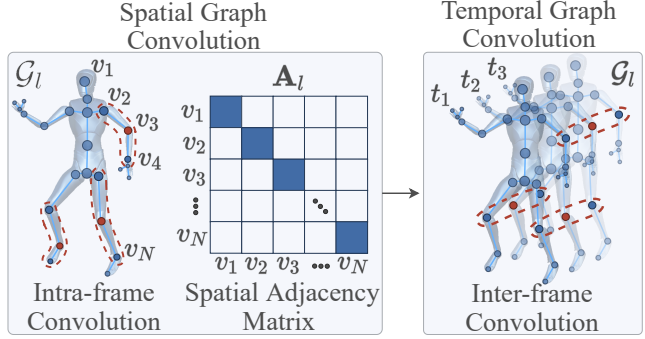

Figure 3: **Spatiotemporal graph convolution used in Kinetic-GAN**. The spatial graph convolution takes as input a skeleton graph $\mathcal{G}_l$ where its corresponding adjacency matrix $\mathbf{A}_l$ handles the intra-frame (spatial) convolution (red dotted line) through the respective root node (red joint) neighbourhood. The temporal (inter-frame) convolution consists of a 1-dimensional convolution performed on the same positional joints across consecutive frames.

However, since we are modelling skeleton human actions, those operations can not be treated as conventional images with CNNs, which would lead to spatial and temporal distortions. Specifically, the conventional convolutional kernels will lose some structural information embedded in the skeleton data since adjacent joints in the pseudo-image are considered as connected joints. Hence, in each skeleton graph's resolution level l, the Kinetic-GAN employs graph convolutions to circumvent this issue due to their ability to exploit the skeleton's inherent graph structure.

In graph convolutional networks (GCNs), a spatiotemporal graph $\mathcal{G}_l = (\mathcal{V}_l, \mathcal{E}_l)$ represents the skeleton data with N_l joints and T_l frames, where $l = \{1, ..., L\}$, and L is the number of levels of the skeleton graph resolution. Therefore, the feature map of the skeleton sequence is represented as $\mathbf{X}_l \in \mathbb{R}^{N_l \times T_l \times C}$, where C is the number of channels, representing the joints coordinates at resolution level L. A GCN consists of both spatial and temporal graph convolutions. Typically, in the spatial dimension, an adjacency matrix $\mathbf{A}_l \in \{0, 1\}^{N_l \times N_l}$ and the corresponding identity matrix $\mathbf{I}_l$ define the intra-body joints connections, which are used to regulate the receptive fields of the convolution. Due to its high-level formulation, a partitioning strategy is defined to represent the neighbours set of each joint for constructing convolution operations, whereas $\mathbf{A}_l$ and $\mathbf{I}_l$ are dismantled into three partitions p (spatial configuration proposed by [43]), so $\mathbf{A}_l + \mathbf{I}_l = \sum_p \mathbf{A}_{l_p}$. For a single frame at resolution level l, the graph convolution can be visualized in Fig. 3 (left), which is computed as:

$$\mathcal{S}(\mathbf{X}_l) = \sum_{i=1}^{p} \mathbf{\Lambda}_{l_i}^{-\frac{1}{2}} \mathbf{A}_{l_i} \mathbf{\Lambda}_{l_i}^{-\frac{1}{2}} \mathbf{X}_l \mathbf{W}_{l_i}, \qquad (1)$$
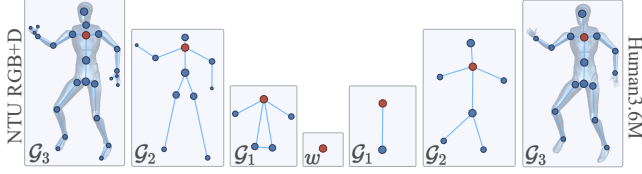
Figure 4: **Graph upsampling and downsampling paths**. Graph pyramids applied by Kinetic-GAN, where the left side refers to NTU RGB+D [31] and NTU-120 RGB+D [25], and the right side shows the Human3.6M [18]. The red nodes represents the root node of the respective dataset.

where the degree matrix $\mathbf{\Lambda}_{l_p}^{ii} = \sum_j (\mathbf{A}_{l_p}^{ij})$ normalizes the adjacency matrix $\mathbf{A}_{l_p}$ through the number of edges attached to each joint node. $\mathbf{W}_{l_p}$ denotes the stacked weight vectors for each partition group p from resolution level l.

Since multiple graph convolutional layers are used, different layers may contain multilevel semantic information [7, 33, 43], and simply using $\mathbf{A}_l$ in Eq. 1 results in the same pre-defined spatial weight configuration to every layer. Hence, we also resort to a learnable weight matrix $\mathbf{M}_l \in \mathbb{R}^{N_l \times N_l}$ (initialized as an all-one matrix) on each layer of both generator and discriminator. Thus, we can adaptively learn to optimize the spatial weight configuration of $\mathbf{A}_l$, and Eq. 1 becomes:

$$\mathcal{S}(\mathbf{X}_l) = \sum_{i=1}^{p} \mathbf{\Lambda}_{l_i}^{-\frac{1}{2}} (\mathbf{A}_{l_i} \odot \mathbf{M}_l) \mathbf{\Lambda}_{l_i}^{-\frac{1}{2}} \mathbf{X}_l \mathbf{W}_{l_i}, \qquad (2)$$

Over the temporal axis, considering that consecutive frames define consecutive skeletons, one-dimensional kernels are used as the temporal graph convolution, which is applied after the spatial graph convolution (Fig. 3). Finally, our spatiotemporal graph convolution at resolution l is given by convolving the positional features joint-wise as:

$$\mathcal{T}(\mathcal{S}(\mathbf{X}_l)) = \mathcal{S}(\mathbf{X}_l) * \mathbf{w}_l, \qquad (3)$$

where $\mathbf{w}_l \in \mathbb{R}^{1 \times t \times C}$ is the temporal kernel at resolution l with t as the number of frames to be convolved in the kernel.

3.2. Generative Adversarial Graph Convolutional Network

Kinetic-GAN consists of a generative adversarial network composed of a graph convolutional-based generator and discriminator. As previously stated, we perform upsampling and downsampling on samples, respectively, in the generator and discriminator (see Fig. 4). Since each spatial graph resolution $\mathcal{G}_l$ has its corresponding adjacency matrix $\mathbf{A}_l$, we compute Eq. 3 at any resolution level l of both upsampling and downsampling streams.

Typically, generative-based action synthesis methods feed a latent vector to the generator and apply auto-regressive techniques [41, 44], or Gaussian processes [42]

to capture the temporal relationships in a latent sequence. From a generative perspective, the multiplicative interactions across the latent sequence provoke the entangling of factors of variation [5, 8]. We depart from this constraint and propose to synthesise human actions from a single latent point z as conventional image generators. However, considering the numerous variation factors in human actions, our generator starts with a non-linear mapping network to transform the latent code $z \in \mathcal{Z}$ to produce an intermediate latent space $\mathcal{W}$. The rationale is to allow less entangled latent factors, as previously confirmed on image modelling [19, 20, 32], and consequently increasing the linearity of factors of variation, where the generation of realistic actions becomes easier for the generator. This phenomenon is verified in the ablation study.

The proposed generator will gradually increase the resolution of a single intermediate latent point w over the spatial and temporal dimensions. First, spatial graph upsampling is computed by introducing new vertices and assigning the corresponding connected joints' average values. One-dimensional interpolation is performed over the temporal axis to increase the number of frames in the skeleton sequence. The discriminator will distinguish between the synthesised actions from the real ones by gradually coarsening the input skeleton's graph, where the vertices are removed, and corresponding neighbours are reconnected. Following the upsampling and downsampling streams from Fig. 4, the resolution is increased and decreased as:

$$\mathbf{X}_{l+1} = \mathcal{T}\Big(\mathcal{S}\big(Up(\mathbf{X}_l)\big)\Big), \quad \mathbf{X}_{l-1} = Down\Big(\mathcal{T}\big(\mathcal{S}(\mathbf{X}_l)\big)\Big) \\ (4)$$

Additionally, we propose to use residual blocks performing a skip connection with solely a temporal graph convolution to learn the temporal mappings more efficiently, which improves training stability and reduces the appearance of artefacts (we verify this phenomenon in the ablation study). Hence, we define our generator and discriminator residual block as:

$$\begin{aligned} \mathbf{X}_{l+1} &= \mathcal{T}\Big(\mathcal{S}\big(Up(\mathbf{X}_l)\big)\Big) + \mathcal{T}\big(Up(\mathbf{X}_l)\big) \\ \mathbf{X}_{l-1} &= Down\Big(\mathcal{T}\big(\mathcal{S}(\mathbf{X}_l)\big) + \mathcal{T}(\mathbf{X}_l)\Big) \end{aligned} \qquad (5)$$

3.3. Conditional Adversarial Training

The proposed architecture can be extended to a conditional model by feeding additional information about factors that we aim to condition. The generation of desired actions is imperative in human action synthesis; thus, we provide the embedded class information of the action y to both generator and discriminator. Specifically, in the generator, the embedded class representation y is concatenated to the

prior input noise z before being mapped to the intermediate latent space $\mathcal{W}$. The discriminator is fed with the channel-wise concatenation of the skeleton with the embedded class representation y. In this paper, we rely on the WGAN-GP objective formulation [13], which is conditioned as:

$$\min_G \max_D \overbrace{\mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}_r}\left[D(\boldsymbol{x}|\boldsymbol{y})\right] - \mathbb{E}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g}\left[D(\tilde{\boldsymbol{x}}|\boldsymbol{y})\right]}^{\text{Discriminator loss}}$$
$$+ \lambda \underbrace{\mathbb{E}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}}[(\|\nabla_{\hat{\boldsymbol{x}}} D\left(\hat{\boldsymbol{x}}|\boldsymbol{y}\right)\|_2 - 1)^2]}_{\text{Gradient penalty}}, \quad (6)$$

where $\mathbb{P}_r$ is the data distribution and $\mathbb{P}_g$ is the model distribution implicitly defined by $\tilde{\boldsymbol{x}} = G(\boldsymbol{z}, \boldsymbol{y}), \boldsymbol{z} \sim p(\boldsymbol{z})$ (the input z is sampled from a noise distribution p, which is then concatenated with the embedded action class representation y). $\mathbb{P}_{\hat{\boldsymbol{x}}}$ is sampled uniformly along straight lines between pairs of points sampled from the data distribution $\mathbb{P}_r$ and generator distribution $\mathbb{P}_g$. The loss weight λ for gradient penalty is set to 10 in all experiments.

3.4. Improving Quality and Diversity

3.4.1 Stochastic Variation

As previously stated, the current state-of-the-art methods are still attached to autoregressive techniques and Gaussian processes. Such procedures performed over the latent space reduce the ability of variation between samples.

Aside from the non-linear mapping network to attain less entangled latent factors, we propose an individual stochastic variation to circumvent this issue without affecting the skeleton structure in the action sequence itself. Specifically, we add random noise to each joint independently in the generator after each spatiotemporal graph convolution, which can be learned by assigning weights to every channel. Our noise injection operation is introduced in the generator's residual block as:

$$\mathbf{X}_{l+1} = \mathcal{T}\Big(\mathcal{S}\big(Up(\mathbf{X}_l)\big)\Big) + \mathcal{T}\big(Up(\mathbf{X}_l)\big) + \boldsymbol{r}_{l+1}\mathbf{w}_{l+1}, \quad (7)$$

where $\boldsymbol{r}_{l+1}$ denotes the Gaussian random noise for each joint to be added at resolution level $l + 1$, and $\mathbf{w}_{l+1}$ is the respective weight vector for each channel. The rationale is to provide a second input to the generator, which is handled to produce variation between samples without forcing the generator to use earlier activations from the latent space to generate random noise. Moreover, each layer has a corresponding per-channel weight $\mathbf{w}_l$ and receives a new random noise $\boldsymbol{r}_l$, giving the flexibility needed to adaptively learn to produce stochastic variation without compromising the skeleton structure of the action (this phenomenon is verified in the ablation study).
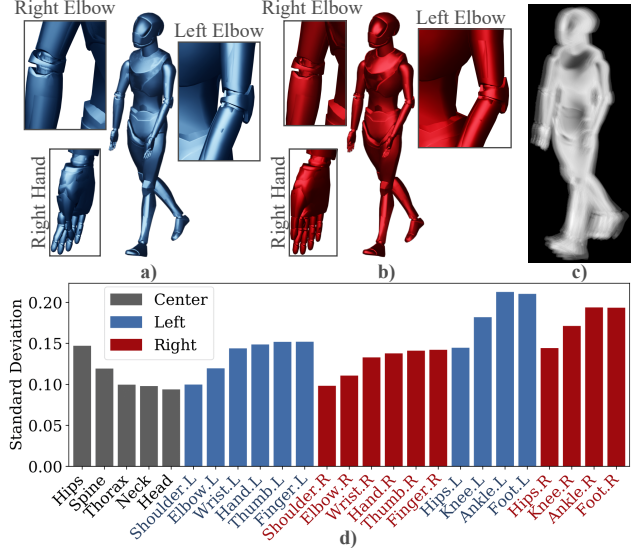

Figure 5: **Examples of stochastic variation**. a) and b) Two human skeletons at the same frame from two action sequences (walking) generated by the same latent point. Zoom-in areas correspond to the respective coloured skeleton. c) Standard deviation over 100 different realizations from the respective frame, highlighting the skeleton parts affected by the noise. d) Cumulative standard deviation of each joint w.r.t. c). A cohesive behaviour is reproduced, where naturally the edges of limbs (hands, fingers, heels, feet) have a higher deviation than their parent joints.

Fig. 5 illustrates the effect of stochastic variation on a walking sequence generated from the same latent point with different noise realizations. 5 a) and b) show how noise affects different parts of the skeletons at the same frame, and c) highlights the standard deviation from 100 different realizations at the same frame, where most affected areas are the legs and arms (w.r.t. the walking action). The lower plot d) comprises a quantitative view of the standard deviation of each joint in c). It can be seen that the noise injector correctly learned the intrinsic natural variation of limbs since it affects more the edges, such as hands and feet than their corresponding parent joints (knees and elbows). Moreover, left limbs (blue) are slightly higher than their corresponding right ones (red) since the respective frame of the skeleton is moving the left leg forward, so naturally, its deviation will be higher than the right leg. For instance, in a throwing action, the arm that throws the object will always have more variation than the other arm. See accompanying video.

3.4.2 Reducing Spatial Artefacts

Typically, the increased quality of generated samples leads to unpleasant artefacts, which is a phenomenon already well-known in image modelling [3, 11, 20, 40]. Those are

often the result of normalization methods over the generator, which improves training stability by eliminating covariate shift. Even so, the feature map's normalization omits any information concerning the individual feature's magnitude. It is assumed that the generator magnifies those magnitudes, and through the normalization process, it becomes unnoticed by the discriminator [20, 40].

This hypothesis is supported by the fact that removing these normalizations from our generator eradicated the appearance of artefacts. Still, it disabled our ability to manipulate human actions. Therefore, we found that reducing the number of layers that employ batch normalization led to better stability in action conditioning while completely removing artefacts. Fig. 6 illustrates an artefact example in action synthesis.
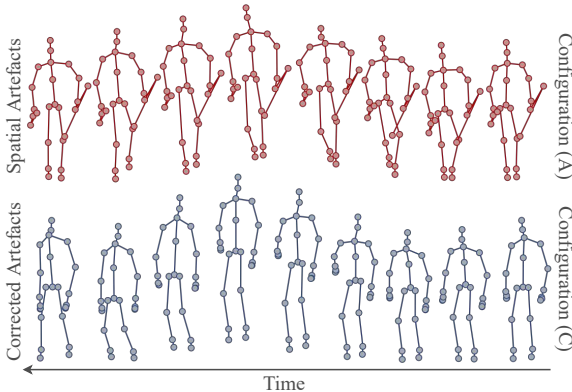

Figure 6: **Spatial artefacts in action synthesis**. First row shows an occurrence of spatial artefacts on a jumping action sequence generated by configuration (A) in Table 1. Similar to image modelling, the remaining information is unaffected as we can identify a human skeleton jumping. Second row shows a jumping action sequence with corrected artefacts by configuration (C) in Table 1.

3.4.3 Truncation Trick on $\mathcal{W}$

As described, the Kinetic-GAN samples z from $\mathcal{N}(0, 1)$ and maps it an intermediate representation $\mathcal{W}$ of that distribution, which is then fed to the generator. So naturally, ranges of low density (in the training data) are not well represented, becoming difficult to learn for the generator, resulting in an important open problem in generative algorithms.

As previously confirmed, sample quality can be improved from truncated [3, 19, 29], or shrunken [22] sampling spaces. Thus, despite some variation losses, we follow the same approach to balance sample quality and diversity. During inference time, we scale the deviation of a given intermediate latent point w from the centre mass of $\mathcal{W}$ as:

$$w' = \mathbb{E}_{z \sim \mathbb{P}_z}[f(z)] + \psi(w - \mathbb{E}_{z \sim \mathbb{P}_z}[f(z)]), \quad (8)$$
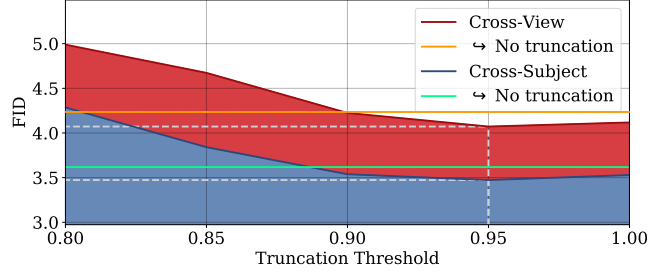

Figure 7: **Improvements through the "truncation trick"**. The horizontal lines correspond to the respective FID obtained in Table 1 without truncations. All action sequences illustrated in this paper uses $\psi = 0.95$.

where $\psi \leq 1$, $f(\cdot)$ denote our mapping network, and $\mathbb{P}_z$ is the latent space distribution from 1000 points. Despite Brock *et al.* [3] reporting that only a subset of networks is manageable to such truncations even when orthogonal regularization is used, truncation in our intermediate space $\mathcal{W}$ successfully works even without adjustments to the loss function. As shown in Fig. 7, we can increase the generation quality in both benchmarks of NTU RGB+D [31]. However, for $\psi \leq 0.9$, the variation starts to decrease and, consequently, the FID starts to increase.

4. Experiments and Discussion

In this section, the experiments are split into local and global movement settings concerning if the 3D locations were positionally normalized into 2D space or not. The ablation studies verify the efficacy of the proposed model's properties over global movement. Then, our best performing model is compared to the state-of-the-art approaches over four datasets: NTU RGB+D [31] and NTU-120 RGB+D [25] (global movement) and Human3.6M [18] and NTU-2D RGB+D [31] (local movement).

4.1. Datasets and evaluation metrics

NTU RGB+D [31]. This dataset is composed of 56,880 video samples with 60 action classes. 3D skeleton data from 40 volunteers are provided for each action sample, with 25 joints for each skeleton. Authors recommend two benchmarks: 1) cross-subject, where models are trained with 20 subjects and tested with the remaining ones; and 2) cross-view, where models are trained with camera views 2 and 3 and tested on camera view 1. A curated version of this dataset is also used (2D joints locations, global to local movement normalization and selected samples) for a fair comparison in Table 4, denoted by NTU-2D RGB+D.

NTU-120 RGB+D [25]. This dataset is an extended version of its predecessor, comprising 114,480 video samples with 120 action classes. Samples were captured with three camera views in 32 different setups and 106 volunteers with

25 body joints for each skeleton. Authors recommend two benchmarks: 1) cross-subject, where models are trained with 53 subjects and tested with the remaining ones, and 2) cross-setup, where models are trained from samples with even setup IDs and tested on odd setup IDs.

Human3.6M [18]. This dataset is a more simplistic set with 2D human motions and 15 body joints for each skeleton. For a fair comparison between the state-of-the-art, the same pre-processing and settings as [41, 44] were followed with corresponding 10 action classes.

Evaluation metrics. Two evaluation metrics are used for estimating the quality of synthetic samples. Similar to image modelling, we use the Fréchet Inception Distance (FID) [15], measuring the distance between the real data distribution and the synthesised one, considering the output activations of a specific layer from an InceptionV3 network [37]. Additionally, we measure the Maximum Mean Discrepancy (MMD) between real and synthetic samples based on a two-sample test to measure the discrepancy of both distributions. The MMD over motion dynamics corresponds to the average MMD across each frame, denoted by MMD_a, and the MMD over whole sequences are indicated as MMD_s. For joints over 2D space, only the MMD is used.

4.2. Global Movement Settings

4.2.1 Ablation study

Before diving into state-of-the-art comparisons, we first demonstrate experimentally that Kinetic-GAN properties improve sample quality considerably.

Method	FID	MMD_a	MMD_s	FID	MMD_a	MMD_s
		Cross-Subject			Cross-View	
c-GAN [30]	27.480	0.919	0.975	31.875	0.993	1.088
Baseline [42]	6.030	0.873	0.954	7.114	0.910	0.991
A Proposed	5.621	0.836	0.927	6.528	0.883	0.953
B + No Residual	19.723	0.892	0.961	21.331	0.971	1.030
C + Regular BN	4.751	0.815	0.917	5.328	0.867	0.940
D + Noise Inject	4.698	0.811	0.895	5.102	0.851	0.933
E + Mapping Net	**3.618**	**0.772**	**0.871**	**4.235**	**0.824**	**0.913**

(rows grouped under **NTU RGB+D**)

Table 1: **Evaluating different generator designs**. The FID and MMD scores (lower is better) between real and synthetic samples generated under global body movement.

Different generator designs. In Table 1, we compare the FID and MMD for various generator architectures in both benchmarks of NTU RGB+D, evaluating each distribution under the respective settings of each method. While the baseline (CSGN [42]) generates local movement in 3D space, we generate global movement and still exceed their performance by a significant margin. In CSGN [42], the application of Gaussian processes limits their generation diversity due to the multiplicative interactions across each di-

mension of the latent space, which emulate the entangling of factors of variation. Additionally, the application of such processes has a high computational cost.

We start with our proposed configuration (A) with conditional sampling applying temporal skip connections and batch normalization over each generator layer. We then confirm the importance of residual blocks in our method by removing them from the generator (B), which clearly reduces sample quality, mainly due to training instability and, consequently, the appearance of artefacts. Since configuration (A) also generates occasional artefacts and eradicating batch normalization sacrificed our ability to generate desirable actions, we propose to regularize the use of batch normalization (C) by removing it when spatial upsampling is performed. This allows us to eliminate artefacts and have complete control to produce desirable actions while still improving sample quality compared to (A). We also introduce noise injection (D) that further improves the results by adding even more diversity. Finally, we also include our mapping network (E), where we distinctly overcome previous baselines and configurations due to the better disentangled latent space.

Method	FID	MMD_a	MMD_s	FID	MMD_a	MMD_s
		Cross-Subject			Cross-View	
D Mapping Net ∅	4.698	0.811	0.895	5.102	0.851	0.933
Mapping Net 1	4.049	0.792	0.892	4.792	0.847	0.931
Mapping Net 2	3.987	0.780	0.885	4.698	0.839	0.929
E Mapping Net 4	**3.618**	**0.772**	**0.871**	4.473	0.831	0.926
E Mapping Net 6	3.849	0.801	0.889	**4.235**	**0.824**	**0.913**
Mapping Net 8	4.396	0.805	0.891	4.610	0.837	0.919

(rows grouped under **NTU RGB+D**)

Table 2: **Importance of the mapping network**. The number in method indicates the depth of the mapping network.

Mapping network effectiveness. The importance of the mapping network is presented in Table 2, where we compare configuration (D) (not using a mapping network) with the increased number of layers in the mapping network. The cross-view benchmark required a deeper network than the cross-subject to attain the optimal performance. We justify this phenomenon due to the increasing number of different subjects in the data, which results in a more complex latent space representation. This indeed confirms the importance of the mapping network since naturally, if there is a greater number of different people, the variations will be higher.

4.2.2 Synthesising 120 different actions

Apart from our ablation studies in Section 4.2.1 which were produced considering global movement settings, Table 3 also presents the performance obtained by Kinetic-GAN over the NTU-120 RGB+D with 120 different actions, being currently the most extensive and challenging set with 3D

joints annotations. Despite achieving a significantly better quality than previous methods, the ability to generate 120 different actions under global movement settings is a substantial improvement in comparison with previous state-of-the-art where they could only generate under local movement settings, and just 10 different actions [14, 41, 44].

	Method	FID	MMD$_a$	MMD$_s$	FID	MMD$_a$	MMD$_s$
NTU-120			*Cross-Subject*			*Cross-Setup*	
	c-GAN [30]	54.403	1.037	1.104	58.531	1.082	1.141
	Kinetic-GAN	**5.967**	**0.819**	**0.906**	**6.751**	**0.847**	**0.934**

Table 3: **Global movement generation results** on NTU-120 RGB+D with 120 different actions.

4.3. Local Movement Settings

Since most state-of-the-art methods were still limited to local movement settings, we also present the results under the same conditions for a fair comparison, where 3D locations were projected into 2D space and normalized positions. We follow the same settings as previous methods [14, 41, 44] and report the results on Table 4. On both datasets, we verify the superiority of our model, where we achieve state-of-the-art performance with a large margin.

	Method	MMD$_a$		MMD$_s$	
Human3.6M	E2E [38]	0.991		0.805	
	EPVA [38]	0.996		0.806	
	adv-EPVA [38]	0.977		0.792	
	SkeletonVAE [14]	0.452		0.467	
	SkeletonGAN [4]	0.419		0.419	
	c-SkeletonGAN [41]	0.195		0.218	
	c-GAN [30]	0.161		0.187	
	SA-GCN [44]	0.146		0.134	
	Kinetic-GAN	**0.071**		**0.082**	
		MMD$_a$	MMD$_s$	MMD$_a$	MMD$_s$
NTU-2D RGB+D		*Cross-Subject*		*Cross-View*	
	SkeletonVAE [14]	0.992	1.136	1.079	1.205
	SkeletonGAN [4]	0.698	0.788	0.999	1.311
	c-SkeletonGAN [41]	0.338	0.402	0.371	0.398
	c-GAN [30]	0.334	0.354	0.365	0.373
	SA-GCN [44]	0.285	0.299	0.316	0.335
	Kinetic-GAN	**0.256**	**0.273**	**0.295**	**0.310**

Table 4: **Local movement generation results**. The MMD scores (lower is better) between real and synthetic samples generated over Human3.6M and NTU-2D RGB+D.

4.3.1 Increasing temporal action length

In addition to generation quality, action conditioning (desirable actions) and local/global movement, modelling long-term human actions is also a significant concept in action synthesis. However, NTU RGB+D could not be employed for such a study since the action execution average is 64
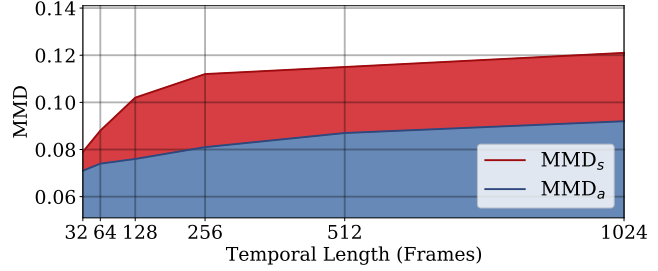

Figure 8: **Increasing action length** on Human3.6M.

frames, where the remaining frames (maximum 300) are all set to 0 for normalization purposes. For this reason, such normalization can disturb the learning process of an action synthesis algorithm. So, we performed long-term experiments over Human3.6M, Fig. 8. Considering that our model can perform *bidirectional temporal dependency*, mainly due to generating a whole human action sequence altogether, we can generate up to 1024 frames (34 seconds). Some autoregressive models [10, 48] gradually freeze poses during the sequence as a result of losing temporal dependencies for such sequence lengths. It can also be observed that its generation quality will stabilize over lengths > 256 frames, which can also be positively regarded.

5. Conclusions and Further Work

This paper introduced a novel Generative Adversarial Graph Convolutional Network for human action synthesis. By generating a human action in a holistic way directly from the latent space, we are able to better disentangle the variation factors through a mapping network, obtaining better representations in the latent space. Furthermore, the introduction of learnable noise injection modules facilitates the generation of variety without compromising the skeleton structure. As a result, we can generate up to 120 different complex actions, which, to the best of our knowledge, were particularly challenging for previous approaches under global movement settings. The proposed method was evaluated on three well known datasets (NTU RGB+D, NTU-120 RGB+D and Human3.6M), advancing the state-of-the-art performance metrics by a significant margin.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1418–1427, 2018.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[4] Haoye Cai, Chunyan Bai, Yu-Wing Tai, and Chi-Keung Tang. Deep video generation, prediction and completion of human action sequences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–382, 2018.

[5] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

[6] Zhan Chen, Sicheng Li, Bing Yang, Qinghan Li, and Hong Liu. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1113–1122, 2021.

[7] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 183–192, 2020.

[8] Guillaume Desjardins, Aaron Courville, and Yoshua Bengio. Disentangling factors of variation via generative entangling. *arXiv preprint arXiv:1210.5474*, 2012.

[9] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[10] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015.

[11] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. Deep generative adversarial compression artifact removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4826–4835, 2017.

[12] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[14] Ikhsanul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. A recurrent variational autoencoder for human motion synthesis. In *28th British Machine Vision Conference*, 2017.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[18] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 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.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[21] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid. A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3288–3297, 2017.

[22] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

[23] Jogendra Nath Kundu, Maharshi Gor, Phani Krishna Uppala, and Venkatesh Babu Radhakrishnan. Unsupervised feature learning of human actions as trajectories in pose embedding manifold. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1459–1467. IEEE, 2019.

[24] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 786–792, 2018.

[25] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019.

[26] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European conference on computer vision*, pages 816–833. Springer, 2016.

[27] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. Global context-aware attention lstm networks for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1647–1656, 2017.

[28] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

[29] Marco Marchesi. Megapixel size image creation using generative adversarial networks. *arXiv preprint arXiv:1706.00082*, 2017.

[30] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[31] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[32] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020.

[33] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019.

[34] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[35] Tae Soo Kim and Austin Reiter. Interpretable 3d human action analysis with temporal convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–28, 2017.

[36] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

[37] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[38] Ruben Villegas, Dumitru Erhan, Honglak Lee, et al. Hierarchical long-term video prediction without supervision. In *International Conference on Machine Learning*, pages 6038–6046. PMLR, 2018.

[39] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020.

[40] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[41] Zhenyi Wang, Ping Yu, Yang Zhao, Ruiyi Zhang, Yufan Zhou, Junsong Yuan, and Changyou Chen. Learning diverse stochastic human-action generators by learning smooth latent transitions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12281–12288, 2020.

[42] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. Convolutional sequence generation for skeleton-based action synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4394–4402, 2019.

[43] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[44] Ping Yu, Yang Zhao, Chunyuan Li, Junsong Yuan, and Changyou Chen. Structure-aware human-action generation. In *European Conference on Computer Vision*, pages 18–34. Springer, 2020.

[45] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2117–2126, 2017.

[46] Pengfei Zhang, Cuiling Lan, Wenjun Zeng, Junliang Xing, Jianru Xue, and Nanning Zheng. Semantics-guided neural networks for efficient skeleton-based human action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1112–1121, 2020.

[47] Xikun Zhang, Chang Xu, and Dacheng Tao. Context aware graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14333–14342, 2020.

[48] Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. In *International Conference on Learning Representations*, 2018.