Learning Diverse Stochastic Human-Action Generators by Learning Smooth Latent Transitions

Zhenyi Wang[1], **Ping Yu**[1], **Yang Zhao**[1], **Ruiyi Zhang**[2], **Yufan Zhou**[1], **Junsong Yuan**[1], **Changyou Chen**[1]
[1] State University of New York at Buffalo
[2] Duke University
[1] {zhenyiwa, pingyu, yzhao63, yufanzho, jsyuan, changyou}@buffalo.edu
[2] ryzhang@cs.duke.edu

Abstract

Human-motion generation is a long-standing challenging task due to the requirement of accurately modeling complex and diverse dynamic patterns. Most existing methods adopt sequence models such as RNN to directly model transitions in the original action space. Due to high dimensionality and potential noise, such modeling of action transitions is particularly challenging. In this paper, we focus on skeleton-based action generation and propose to model smooth and diverse transitions on a latent space of action sequences with much lower dimensionality. Conditioned on a latent sequence, actions are generated by a frame-wise decoder shared by all latent action-poses. Specifically, an implicit RNN is defined to model smooth latent sequences, whose randomness (diversity) is controlled by noise from the input. Different from standard action-prediction methods, our model can generate action sequences from pure noise without any conditional action poses. Remarkably, it can also generate unseen actions from mixed classes during training. Our model is learned with a bi-directional generative-adversarial-net framework, which not only can generate diverse action sequences of a particular class or mix classes, but also learns to classify action sequences within the same model. Experimental results show the superiority of our method in both diverse action-sequence generation and classification, relative to existing methods.

Introduction

Human-action generation is an important task for modeling dynamic behavior of human activities, with vast real applications such as video synthesis (Wang et al. 2018a), action classification (Kim and Reiter 2017; Ke et al. 2017; Yan, Xiong, and Lin 2018; Liu et al. 2016; Shahroudy et al. 2016; Du, Wang, and Wang 2015) and action prediction (Martinez, Black, and Romero 2017; Wang et al. 2018b; Barsoum, Kender, and Liu 2017). Directly generating human actions from scratch is particularly challenging due to the complexity and high-dimensionality of natural scenes. One promising workaround is to first generate easier-to-deal-with skeleton-based action sequences, based on which natural sequence are then rendered. This paper thus focuses on skeleton-based action-sequence generation.

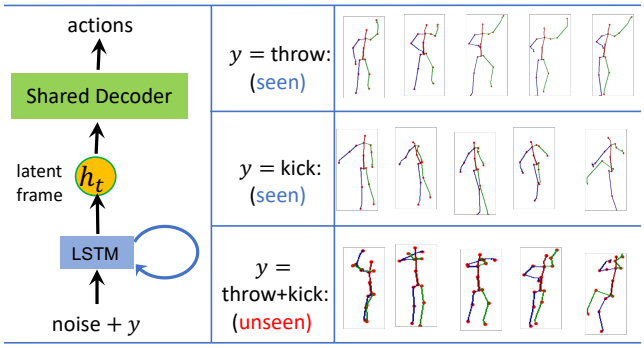


Figure 1: Generating sentences from pure noise, our model can learn smooth latent-frame transitions via an *implicit LSTM-based* RNN, which are then decoded to an action sequence via a shared decoder. The action sequence endows a flexible implicit distribution induced by the input noise. Our model not only can generate actions whose class are seen during training (*e.g.*, throw, kick), but also can generate actions of unseen mixed classes (*e.g.*, throw+kick).

Skeleton-based human action generation can be categorized into action synthesis (also referred to generation) (Kovar, Gleicher, and Pighin 2002) and prediction. Action synthesis refers to synthesizing a whole action sequence from scratch, with controllable label information; whereas action prediction refers to predicting remaining action-poses given a portion of seed frames. These two tasks are closely related, *e.g.*, the latter can be considered as a conditional variant of the former. In general, however, action synthesis is considered more challenging due to little input information available. Existing action-prediction methods can be categorized into deterministic (Martinez, Black, and Romero 2017; Judith Bütepage 2017; Wang et al. 2018b; Harvey and Pal 2018) and stochastic (Barsoum, Kender, and Liu 2017; Kundu, Gor, and Babu 2019; Habibie et al. 2017) approaches. Predicted action sequences in deterministic approaches are not associated with randomness; thus, there is no variance once input sub-sequences are given. By contrast, stochastic approaches can induce probability distributions over predicted sequences. In most cases, stochastic (probabilistic) approaches are preferable as they allow one to generate dif-

ferent action sequences conditioned on the same context.

For diverse action generation, models are required to be stochastic so that the synthesis process can be considered as drawing samples from an action-sequence probability spaces. As a result, one approach for action synthesis is to learn a stochastic generative model, which induces probability distributions over the space of action sequences from which we can easily sample. Once such a model is learned, new actions can be generated by merely sampling from the generative model.

Among various deep generative models, the generative adversarial network (GAN) (Goodfellow et al. 2014) is one of the state-of-the-art methods, with applications on various tasks such as image generation (Ma et al. 2017), characters creation (Jin et al. 2017), video generation (Vondrick, Pirsiavash, and Torralba 2016) and prediction (Denton and Birodkar 2017; Lee et al. 2018; Liang et al. 2017). However, most existing GAN-based methods for action generation consider directly learning frame transitions on the original action space. In other words, these works define action generators with recurrent neural networks (RNNs) that directly produce action sequences (Liang et al. 2017; Habibie et al. 2017; Lee et al. 2018; Wang, Chai, and Xia 2018). However, these models are usually difficult to train due to the complexity and high-dimensionality of the action space.

In this paper, we overcome this issue by breaking the generator into two components: a smooth-latent-transition component (SLTC) and a global skeleton-decoder component (GSDC). Figure 1 illustrates the key features of and some results from our model. Specifically,

- The SLTC is responsible for generating smooth latent frames, each of which corresponds to the latent representation of a generated action-pose. The SLTC is modeled by an *implicit LSTM*, which takes a sequence of independent noise plus a one-hot class vector as input, and outputs a latent-frame sequence. Our method inherits the advantage of RNNs, which could generate diverse length sequences but on a much lower-dimensional latent space, an advantage over existing methods such as (Cai et al. 2018).
- The GSDC is responsible for decoding each latent frame to an output action pose, via a shared (global) decoder implemented by a deep neural network (DNN). Note that at this stage, only a mapping from a single latent frame to an action-pose needs to be learned, *i.e.*, no sequence modeling is needed, making generation much simpler.

Our model is learned by adopting the bi-directional GAN framework (Dumoulin et al. 2017; Donahue, Krähenbühl, and Darrell 2017), consisting of a stochastic action generator, an action classifier and an action discriminator. These three networks compete with each other adversarially. At equilibrium, the generator can learn to generate diverse action sequences that match the training data. In addition, the classifier is able to learn to classify both real and synthesized action sequences. Our contributions are summarized as follows:

- We propose a novel stochastic action sequence generator architecture, which benefits from an ability to learn smooth latent transitions. The proposed architecture eases the training of the RNN-based generator for sequence generation, and at the same time can learn to generate much higher-quality and diverse actions.
- We propose to learn an action-sequence classifier simultaneously within the bi-directional GAN framework, achieving both action generation and classification.
- Extensive experiments are conducted, demonstrating the superiority of our proposed framework.

Related Works

Skeleton-based action prediction has been studied for years. One of the most popular methods for human motion prediction (conditioned on a portion of seed action-poses) is based on recurrent neural networks (Martinez, Black, and Romero 2017; Wang et al. 2018b; Kundu, Gor, and Babu 2019). For skeleton-based action generation, switching linear models (Pavlović, Rehg, and MacCormick 2001; Bissacco 2005; Oh et al. 2005) were proposed to model stochastic dynamics of human motions. However, it is difficult to select a suitable number of switching states for best modeling. Furthermore, it usually requires a large amount of training data due to the large model size. Restricted Boltzmann Machine (RBM) also has been applied for motion generation (Taylor, Hinton, and Roweis 2007; Sutskever et al. 2008; Taylor and Hinton 2009). However, inference for RBM is known to be particularly challenging. Gaussian-process latent variable models (Wang, Fleet, and Hertzmann 2008; Urtasun et al. 2008) and its variants (Wang, Fleet, and Hertzmann 2007) have been applied for this task. One problem with such methods, however, is that they are not scalable enough to deal with large-scale data.

For deep-learning-based methods, RNNs are probably one of the most successful models (Fragkiadaki et al. 2015). However, most existing models assume output distributions as Gaussian or Gaussian mixture. Different from our *implicit representation*, these methods are not expressive enough to capture the diversity of human actions.

In contrast to action prediction, limited work has been done for diverse action generation, apart from some preliminary work. Specifically, the motion graph approach (Min and Chai 2012) needs to extract motion primitives from prerecorded data; the diversity and quality of action will be restricted by way of defining the primitives and transitions between the primitives. Variational autoencoder and GAN have also been applied in (Habibie et al. 2017; Wang, Chai, and Xia 2018; Kiasari, Moirangthem, and Lee 2018) for motion generation. However, these methods directly learn motion transitions with an RNN, and the error of current frame will be accumulate into the next frame, thus making it inapplicable to generate long action sequences, especially for aperiodic motions such as eating and drinking.

Another distinction between our model and existing methods is that the latter typically require some seed action-frames as input to a generator (Xue et al. 2016; Barsoum, Kender, and Liu 2017; Wang, Chai, and Xia 2018; Kiasari, Moirangthem, and Lee 2018), which is learned based on the GAN framework; whereas our model is designed to generate action sequence from scratch, and learned based on

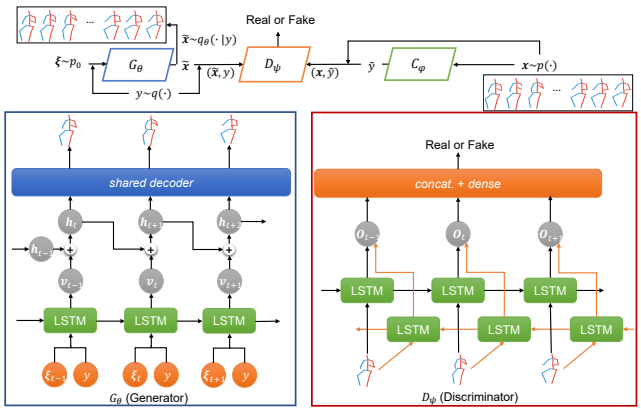


Figure 2: The proposed action-generation model (top), with detailed structures of action sequence generator ($G_{\boldsymbol{\theta}}$), discriminator ($D_{\boldsymbol{\psi}}$). The classifier ($C_{\boldsymbol{\phi}}$) is the same as $D_{\boldsymbol{\psi}}$ except that it outputs a class label instead of a binary value.

the bi-direction GAN framework in order to achieve simultaneous action generation and classification.

The Proposed Model

We first illustrate our whole model in Figure 2, followed by detailed descriptions of specific components.

Problem Setup and Challenges

Our training dataset is represented as $\mathbf{X} \triangleq \{(\mathbf{x}_1^{(i)}, \cdots, \mathbf{x}_{T_i}^{(i)}, \mathbf{y}^{(i)})\}_i$, where $\mathbf{x}_t^{(i)} \in \mathbb{R}^d$ represents one action-pose of dimension d; T_i is the length of the sequence; and $\mathbf{y}^{(i)}$ is the corresponding one-hot label vector of the sequence*. Our basic goal is to train a stochastic sequence generator $G_{\boldsymbol{\theta}}$ using a DNN parametrized by $\boldsymbol{\theta}$. Hence, given a label $\mathbf{y}$ and a sequence of random noises $\boldsymbol{\xi}$ (specified latter), $G_{\boldsymbol{\theta}}$ is supposed to generate a new action sequence following

$$(\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_T) = G_{\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\xi}) \,, \tag{1}$$

where T is the length of the sequence that can be specified flexibly in $G_{\boldsymbol{\theta}}$.

Remark 1 *Similar to implicit generative models such as GAN, we call the generator form* (1) *an* implicit generator, *in the sense that the generated sequence* $(\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_T)$ *is a random sequence endowing an implicit probability distribution with an unknown density function induced by the random noise* ξ. *A traditional way of modeling action sequences usually defines* $G_{\boldsymbol{\theta}}$ *as an RNN, which typically defines a Gaussian distribution (explicit) for* $\tilde{\mathbf{x}}_t$, *referred to as explicit modeling. An implicit distribution is typically much more flexible than explicit distributions as the density is not restricted to a particular distribution class.*

*Our model can also be applied to data without labels by simply removing $\mathbf{y}^{(i)}$ from the generator. We focus on the one with labels.

Challenges There are a few challenges. The first one relates to how to define an expressive-enough generator for diverse action generation. We adopt an implicit model of action sequences without explicit form assumption; Thus, the generator benefits from better representation power to generate more sophisticated, higher-quality and diverse action sequences. The second challenge is to find an appropriate generator structure. One straightforward way is to define $G_{\boldsymbol{\theta}}$ as an RNN that outputs action sequences directly, similar to (Habibie et al. 2017; Wang, Chai, and Xia 2018). However, it is well known that an RNN with high-dimensional outputs is challenging to train (Pascanu, Mikolov, and Bengio 2013). In recent years, attention (Bahdanau, Cho, and Bengio 2015) and the Transformer model (Vaswani et al. 2017) have been developed to enhance/replace RNN-based models. Attention and Transformer are used for addressing the long-term dependency problem in seq2seq-based models. They are not directly applicable to our setting because our model is not a simple seq2seq model, as our inputs are purely random noise. To this end, we propose a novel generator structure, where smooth *latent-action transitions* are first inferred via an RNN, which are then fed into a shared frame-wise decoder (non-sequential) to map all latent poses to their corresponding action-poses. The detailed structure of the generator is illustrated in Figure 2 and described below.

Stochastic Action-Sequence Generator

Our action-sequence generator consists of two components, SLTC and GSDC. The detailed structure of the generator is illustrated as $G_{\boldsymbol{\theta}}$ in Figure 2.

Learning smooth latent transitions Instead of directly modeling sequential transitions in the action space, we propose to model them in a latent action-sequence space. To this end, we decompose $G_{\boldsymbol{\theta}}$ as compositions of an *implicit LSTM* and a shared frame-wise decoder. The LSTM generator (a.k.a. SLTC) models smooth latent action transitions, and the shared decoder (a.k.a. GSDC) models frame-wise mapping from latent space to action space. Specifically, denote $\mathbf{h}_t$ to be the latent representation of an action-pose $\mathbf{x}_t$. We define $(\mathbf{h}_1, \cdots, \mathbf{h}_T)$ to be outputs of an *implicit LSTM*, written as:

$$(\mathbf{h}_1, \cdots, \mathbf{h}_T) = \mathsf{LSTM}\,(\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_T, \mathbf{y}; \boldsymbol{\theta}_1) \,, \tag{2}$$

where $(\boldsymbol{\xi}_t, \mathbf{y})$ is the input of the LSTM at time t (the noise $\boldsymbol{\xi}_t$'s are independent of each other for all t); and $\boldsymbol{\theta}_1 \subset \boldsymbol{\theta}$ is the parameter of the LSTM network. We called (2) *implicit LSTM* because its input consists of independent noise $\boldsymbol{\xi}_t$ at each time in both training and testing (generation) stages (please see the generator graph in Figure 2). This generator is different from standard LSTM where the output of previous time will be used as input of current time in the testing stage. In addition, the noise in the input would induce a much more flexible implicit distribution on $\mathbf{h}_t$; whereas standard LSTM defines an explicit distribution such as Gaussian, restricting the representation power. Another advantage of adopting an implicit LSTM as a latent-frame generator is that the length of a generated action sequence could be induced from the latent space instead of the action space. In general, the dimension

of $\mathbf{h}_t$ is much smaller than action poses, making the training of the LSTM easier. Finally, modeling latent representations with an LSTM also allows latent transitions to be smooth, that is the desired property of action sequence generation.

To further ease the training of LSTM, we propose a variant whose outputs are defined as the residual latent sequences. That is, instead of modeling as in (2), we define the following generating process:

$$\begin{aligned}(\mathbf{v}_1, \cdots, \mathbf{v}_T) &= \mathsf{LSTM}(\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_T, \mathbf{y}; \boldsymbol{\theta}_1) \\ \mathbf{h}_{t+1} &= \mathbf{h}_t + \mathbf{v}_t \ . \end{aligned} \tag{3}$$

The shared frame-wise decoder The second component, GSDC, is a shared frame-wise decoder mapping one latent frame $\mathbf{h}_t$ to the corresponding action pose $\mathbf{x}_t$. Specifically, given $\mathbf{h}_t$ from SLTC, we have, for all t,

$$\tilde{\mathbf{x}}_t = \mathsf{Dec}(\mathbf{h}_t, \mathbf{y}; \boldsymbol{\theta}_2) \ ,$$

where $\mathsf{Dec}(\cdot, \cdot; \boldsymbol{\theta}_2)$ represents a decoder implemented as any DNN with parameter $\boldsymbol{\theta}_2 \subset \boldsymbol{\theta}$, mapping an input latent frame $\mathbf{h}_t$ to an output action pose $\tilde{\mathbf{x}}_t$. In experiments, we use a simple MLP structure for Dec.

The whole action sequence generator Stacking the above two components constitutes our implicit action generator. To further enforce smooth transitions, we penalize the generated latent action poses by the changes of consecutive frames, *i.e.*, with the following regularizer, similar to (Cai et al. 2018):

$$\Omega(\{\mathbf{h}_t\}, \{\tilde{\mathbf{x}}_t\}) \triangleq \sum_{t=2}^{T} \left(\sigma_1 \| \mathbf{h}_t - \mathbf{h}_{t-1} \|^2 + \sigma_2 \|\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}\|^2\right) \tag{4}$$

where σ_1 and σ_2 control the relative importance of the corresponding regularizer term.

Action-Sequence Classifier

Modeling action-sequence generation and classification simultaneously enables information sharing between the generator and classifier, thus it is expected to be able to boost model performance. As a result, we define a classifier C with a bi-directional LSTM (Schuster and Paliwal 1997), whose outputs are further appended with a fully connected layer and a softmax layer for classification. The purpose of adopting the bi-directional LSTM is to effectively model frame-wise relation from two directions, which has been shown more effective than single direction modeling in sequence models (Huang, Xu, and Yu 2015; Sundermeyer et al. 2014; Graves 2012). Please refer to $C_{\boldsymbol{\phi}}$ in Figure 2 for a detailed structure of our sequence classifier.

Action Discriminator and Model Training

Bi-directional GAN based training The proposed action-sequence generator and classifier constitute a pair of networks that can translate between each other, *i.e.*, inverting the classifier achieves the same goal of the generator. To train these two networks effectively, we borrow ideas from bi-directional GAN, and define a discriminator to play adversarial games with the generator and classifier. Specifically, the action-label pairs come from two sources: one starts from a random label and then generates an action sequence via the generator

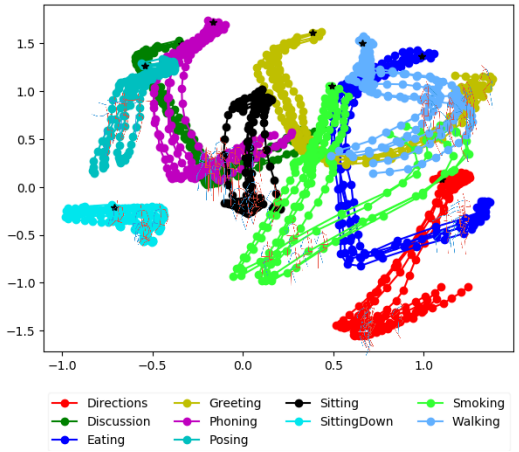


Figure 3: Latent space with dimension = 2. The trajectories intercept with each other due to some similar frames in different action sequences.

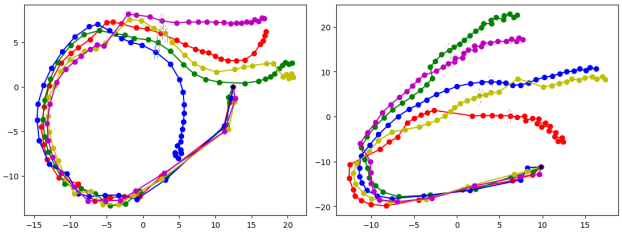


Figure 4: Action diversity of generated latent trajectories. Left: Greeting; Right: Posing.

$G_{\boldsymbol{\theta}}$; the other starts from a randomly-sampled training action sequence and then generates a label via the classifier $C_{\boldsymbol{\phi}}$. Let $q(\mathbf{y})$ be a prior distribution over labels[†]; $q_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}|\,\mathbf{y})$ be the implicit distribution induced by the generator; $p(\mathbf{x})$ be the empirical action-sequence distribution of the training data; and $p_{\boldsymbol{\phi}}(\tilde{\mathbf{y}}|\,\mathbf{x})$ be the conditional label distribution induced by the classifier given a sequence $\mathbf{x}$. Our model updates the generator $G_{\boldsymbol{\theta}}$, the classifier $C_{\boldsymbol{\phi}}$, and the discriminator $D_{\boldsymbol{\psi}}$ alternatively by following the GAN training procedure. Similar to the classifier, the discriminator is also defined by a bidirectional LSTM. The bi-directional GAN is trained to match the joint distributions $q(y)q_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}|y)$ and $p(\mathbf{x})p_{\boldsymbol{\phi}}(\tilde{\mathbf{y}}|\,\mathbf{x})$, via the following min-max game:

$$\begin{aligned}\min_{G_{\boldsymbol{\theta}}, C_{\boldsymbol{\phi}}} \max_{D_{\boldsymbol{\psi}}} V(G_{\boldsymbol{\theta}}, C_{\boldsymbol{\phi}}, D_{\boldsymbol{\psi}}) &= \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}), \tilde{\mathbf{y}}\sim p_{\boldsymbol{\phi}}(\cdot|\,\mathbf{x})}\left[\log D_{\boldsymbol{\psi}}(\mathbf{x}, \tilde{\mathbf{y}})\right] \\ &+ \mathbb{E}_{y\sim q(y), \tilde{\mathbf{x}}\sim q_{\boldsymbol{\theta}}(\cdot|\,\mathbf{y})}\left[\log(1 - D_{\boldsymbol{\psi}}(\tilde{\mathbf{x}}, \mathbf{y}))\right] \ . \end{aligned} \tag{5}$$

In addition, motivated by CycleGAN (Zhu et al. 2017) and ALICE (Li et al. 2017), a cycle-consistency loss is introduced:

$$\mathcal{L}_c \triangleq H(C_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\xi})), \mathbf{y}) \ , \tag{6}$$

where $H(\cdot, \cdot)$ denotes the cross entropy between two distributions. Combining (4), (5) and (6) constitutes the final loss of our model.

Shared frame-wise decoder pretraining It is useful to pretrain the shared frame-wise decoder with the training data.

[†]We adopt a uniform distribution in our method.

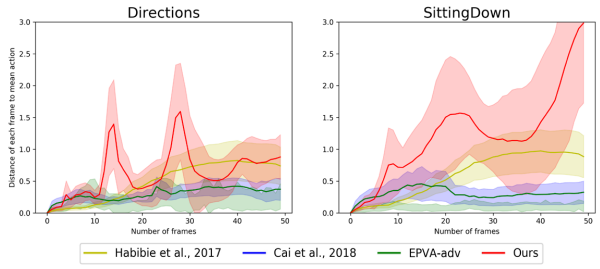


Figure 5: Diversity of generated action sequences.

To this end, we use the conditioned WGAN-GP model (Gulrajani et al. 2017) to train a generator to generate independent action poses from a given label. The generator, denoted as $\bar{G}_{\boldsymbol{\theta}_2}(\cdot)$, corresponds to the shared decoder in our model. To match the input with our frame decoder, we replace the original input $\mathbf{h}_t$ with a random sample from a simple distribution $p_h(\cdot)$, *e.g.*, the standard Gaussian distribution. The discriminator, denoted as $\bar{D}(\cdot)$, is an auxiliary network to be discarded after pretraining. The objective function is defined as:

$$\min_{\bar{G}_{\boldsymbol{\theta}_2}} \max_{\bar{D}} \bar{V}(\bar{D}, \bar{G}_{\boldsymbol{\theta}_2}) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\bar{D}(\mathbf{x})] - \tag{7}$$

$$\mathbb{E}_{\mathbf{h}\sim p_h(\mathbf{h})}[\bar{D}(\bar{G}_{\boldsymbol{\theta}_2}(\mathbf{h}))] + \lambda \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[(\| \bigtriangledown_{\mathbf{x}} \bar{D}(\mathbf{x}) \|_2 - 1)^2]$$

where p_{data} denotes the frame distribution of training data; and λ controls the magnitude of the gradient penalty to enforce the Lipschitz constraint.

Finally, the whole training procedure of our model is described in Algorithm (See Appendix).

Experiments

We evaluate our proposed model for diverse action generation on two datasets, in terms of both action-sequence quality and diversity. We also conduct extensive human evaluation for the results generated by different models. Ablation study, implementation details and more results are provided in the Appendix. Code is also made available[‡].

Datasets & Baselines

Datasets We adopt the human-3.6m dataset (Catalin Ionescu and Sminchisescu 2014) and the NTU dataset (Shahroudy et al. 2016). The human-3.6m is a large scale dataset for human activity recognition and analysis. Following (Cai et al. 2018), we subsample the video frames to 16 fps to obtain more significant action variations. Our model is trained on 10 classes of actions, including *Directions*, *Discussion*, *Eating*, *Greeting*, *Phoning*, *Posing*, *Sitting*, *SittingDown*, *Walking*, and *Smoking*.

The NTU RGB+D is a large action dataset collected with Microsoft Kinect v.2 cameras (Shahroudy et al. 2016). For our purpose, we only use the 2D skeleton locations of 25 major body joints in the corresponding depth/IR frame data. Similar to the human3.6m dataset, we also sample 10 action classes for training and testing, including *drinking water*, *throw*, *sitting down*, *wear jacket*, *standing up*, *hand waving*, *kicking something*, *jump up*, *make a phone call* and *cross hands in front*. We follow the evaluation protocols of previous literature on this dataset to adopt cross-subject and cross-view recognition accuracy. For the cross-subject evaluation, sequences for training (20 subjects) and testing (20 subjects) come from different subjects. For the cross-view evaluation, the training dataset consists of action sequences collected by two cameras, and the test dataset consists of the remaining data. After splitting and cleaning missing or incomplete sequences, there are 2260 and 1070 action sequences for training and testing, respectively, for cross-subject evaluation; and there are 2213 and 1117 action sequences for training and testing, respectively, for cross-view evaluation.

Baselines Generating action sequences from scratch is a relatively less explored field. The most related models to ours we found are the recently proposed generative models EPVA-adv in (Wichers et al. 2018), action generator trained with VAE (Habibie et al. 2017) as well as the model proposed in (Cai et al. 2018) for the human-action generation. In the experiments, we will compare our model with these three, as well as other specific baselines.

Detailed Results

Latent frame transitions To show the effectiveness of our proposed latent-transition mechanism, we visualize the learned latent representations for selected classes.

Latent trajectories of different classes on the Human-3.6 dataset are plotted in Figure 3, with a latent-frame dimension of 2. More results with higher dimensionalities are provided in the Appendix. It is interesting to observe that for the 2-dimensional-latent-space case, some latent trajectories intercept with each other. This is reasonable because action poses in different action categories might be similar, *e.g.*, smoking (green) and eating (blue) in Figure 3 (left).

To demonstrate the diversity of the learned action generator, we plot multiple latent trajectories for selected classes in Figure 4, all starting from the same initial point. It is clear that as time goes on, the generated latent frames become more diverse, a distinct property lacking in deterministic generators in most action-prediction models such as (Martinez, Black, and Romero 2017; Wang et al. 2018b).

To better illustrate the diversity of the action sequences, we compare our model with the three recently proposed models (Habibie et al. 2017; Cai et al. 2018; Wichers et al. 2018). The mean and variance of each action pose along time for a set of trajectories are plotted in Figure 5. It is remarkable to find that trajectories from our model are much more diverse than the baselines. For a quantitative comparison, the standard derivations for different action classes are listed in the Appendix. More diverse action sequences generation results are provided in the Appendix. All these results indicate the superiority of our model in generating diverse action sequences.

Quality of generated action sequences We adopt two metrics to measure the quality of the generated actions:

[‡]https://github.com/zheshiyige/Learning-Diverse-Stochastic-Human-Action-Generators-by-Learning-Smooth-Latent-Transitions

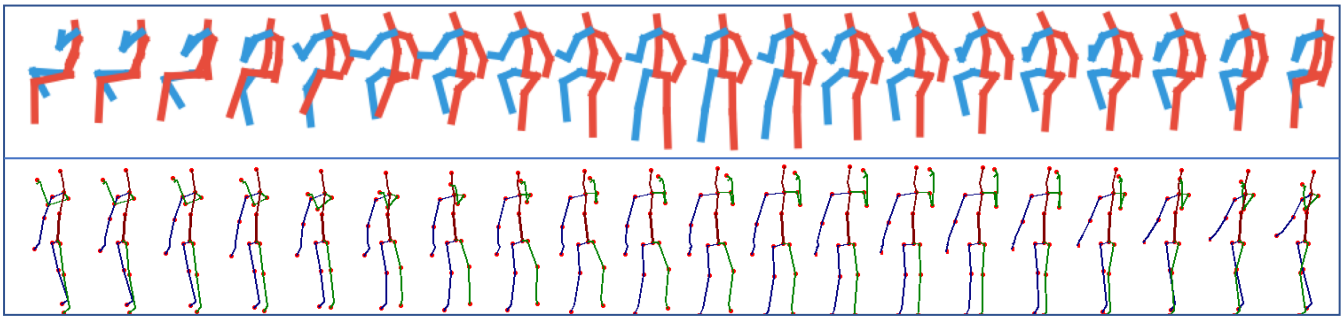
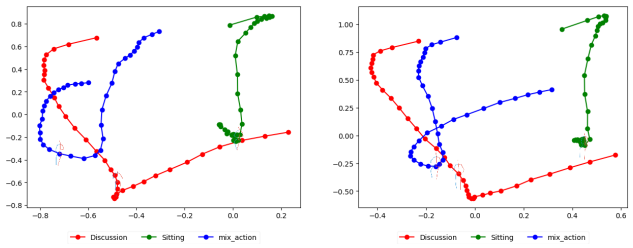


Figure 6: Randomly selected action sequences generated on human3.6 dataset (First row: Smoking) and NTU RGBD dataset (Last row: Drinking)

.

Table 1: Comparisons of our model with (Habibie et al. 2017; Cai et al. 2018; Wichers et al. 2018) in terms of Maximum Mean Discrepancy. (The lower the better.)

	With seed motion			Without seed motion					
	E2E	EPVA	EPVA-adv	E2E	EPVA	EPVA-adv	Habibie et al., 2017	Cai et al., 2018	Ours
MMD_{avg}	0.304	0.305	0.339	0.991	0.996	0.977	0.452	0.419	**0.195**
MMD_{seq}	0.305	0.326	0.335	0.805	0.806	0.792	0.467	0.436	**0.218**



(a) p_1=0.5, p_2=0.5 (b) p_1=0.3, p_2=0.7

Figure 7: Latent space of mixed classes of actions with different mixing proportions p_1 and p_2.

action-classification accuracy and the maximum mean discrepancy (MMD) between generated and real action sequences. The latter metric is adapted from measuring the quality of generated images of GAN-based model, and has been used in (Walker et al. 2017) for measuring the quality of action sequences. For action classification, we adopt the trained classifier from our model to classify both real (testing) actions and a set of randomly generated actions from our model. A good generator should generate action sequences that endow similar or better classification accuracies than real data. We compare our model with a baseline, which uses the same classifier structure but is trained independently on the training data. For every action class, we randomly sample 100 sequences for testing. The classification accuracies are shown in Table 2. It is seen that our model achieves comparable performance on real and generated action sequences, which outperforms the baseline model to a large margin in general. In some cases, the accuracy on generated action sequence is higher than that of real data is because both the real data and the generated data are involved in the training of classifier under the bidirectional GAN framework (Donahue, Krähenbühl, and Darrell 2017). The cross-subject and cross-view classification accuracies across different classes are 0.824 and 0.885 for our model respectively.

The MMD measures the discrepancy of two distributions based on their samples (the generated and real action sequences in our case). Since our data are represented as sequences, we proposed two sequence-level MMD metrics (more details in Appendix **??**). Following (Walker et al. 2017), we vary the bandwidth from 10^{-4} to 10^{9} and report the maximum value computed. We compared our model with (Habibie et al. 2017; Cai et al. 2018; Wichers et al. 2018). (Wichers et al. 2018) contains three models: E2E, EPVA and EPVA-adv. These models require a few seed frames as inputs to the generator. To adapt these model to our setting, we define two variants: 1) given labels and only the first frame as seed input to their models; 2) given only labels but no frames as seed input. The results are reported in Table 1. It is interesting to see that without the need of a seed action-pose, our model obtains much lower MMD scores than the baselines with no seed action-poses. Figure 6 further plots some examples of generated action sequences on the two datasets, which further demonstrates the high quality of our generated actions.

Novel action generation by mixing action classes Another distinct feature of our model is its ability to generate novel action sequences by specifying the input class variable $\mathbf{y}$. One interesting way for this is to mix several action classes such that the elements satisfy $\mathbf{y}_k \geq 0$ and $\sum_k \mathbf{y}_k = 1$. When feeding such a soft-mixed label into our model, due to the smoothness of the learned latent space, one would expect the generated sequence contains all features from the mixing classes. To demonstrate this, we consider mixing two classes. Figure 7 plots the latent trajectories for different mixing coefficients. As expected, the trajectories of mixing classes smoothly interpolate between the original trajectories. For better visualization, let the first two classes correspond to *walking* and *phoning*. We set $\mathbf{y} \triangleq (0.5, 0.5, 0, \cdots, 0)$ and

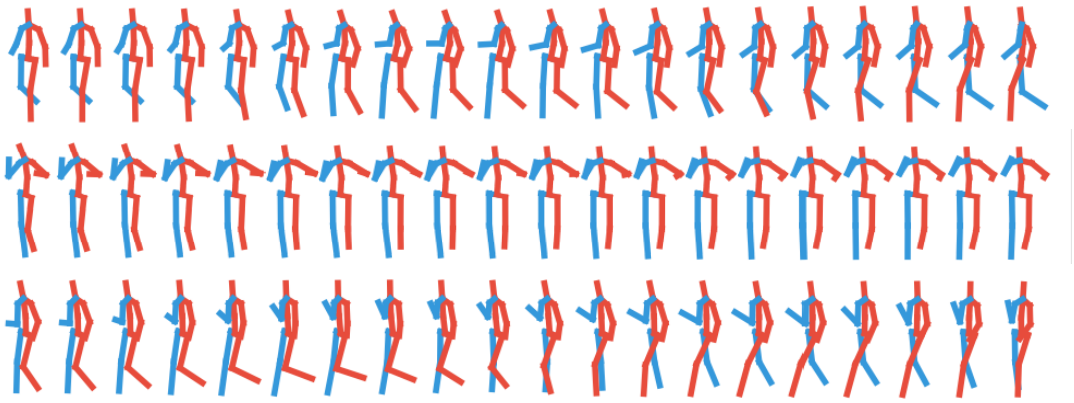


Figure 8: Novel mixed action sequences generated on human3.6 dataset. First row: generated sequence of "Walking"; Second row: generated sequence of "Phoning"; Third row: generated sequence of mixed "Walking" + "Phoning".

Table 2: Cross-view and cross-subject evaluation of classification accuracies on NTU-RGB dataset. Baseline_T means the independently trained classifier testing on real data, and Baseline_G means the independently trained classifier testing on generated sequences. Ours_T means our model testing on real data, and Ours_G means our model testing on generated sequences.

Split	Data	drinking	throw	sitting down	wear jacket	standing up	hand waving	kick	jump	phoning	cross hands
cross-view	Baseline_T	0.71	0.90	0.92	0.94	0.93	0.73	0.85	0.76	0.87	**0.92**
	Baseline_G	0.65	0.21	0.25	0.38	0.75	0.61	0.66	0.46	0.75	0.13
	Ours_T	0.82	0.92	0.94	**0.96**	0.95	0.76	0.86	**0.93**	0.84	0.88
	Ours_G	**0.87**	**0.94**	**0.98**	0.93	**0.95**	**0.81**	**0.97**	0.79	**0.91**	0.82
cross-sub	Baseline_T	0.76	0.80	0.90	0.90	0.92	0.73	0.6	0.67	**0.86**	0.74
	Baseline_G	0.60	0.15	0.12	0.57	0.74	0.67	0.34	0.26	0.59	0.21
	Ours_T	0.82	0.65	**0.93**	**0.95**	**0.94**	0.83	0.74	0.69	0.82	**0.81**
	Ours_G	**0.83**	**0.86**	0.92	0.91	0.90	**0.87**	**0.86**	**0.82**	0.81	0.76

generate the corresponding action sequences by feeding it into the generator. The generated sequences are shown in Figure 8. It is interesting to see that the sequence with mixing classes indeed contains actions with both hands and legs, which correspond to walking and phoning, respectively.

Human evaluation We run perceptual studies on Amazon Mechanical Turk (AMT) to assess the realism of generated actions. There were 120 participants in this test for three-round evaluations. Every worker was assigned some collection of evaluation tasks, each of which consists of four videos generated by one of the four models, including (Habibie et al. 2017; Wichers et al. 2018; Cai et al. 2018) and ours. The worker was asked to evaluate each group of videos with a scale from 1 to 5. The higher the score, the more realistic of an action. Table 3 summarizes the results, which clearly shows the superiority of our method over others. Scoring standard and detailed experiment design are provided in the Appendix.

Table 3: Human evaluations.

Model	Average Score
(Habibie et al. 2017)	2.445
(Wichers et al. 2018)	2.387
(Cai et al. 2018)	2.847
Ours	**3.378**

Ablation study We conduct extensive ablation study to better understand each component of our model, including smoothness term, cycle consistency loss, and residual latent sequence prediction. More details are described in the Appendix.

Conclusion

We propose a new framework for stochastic diverse action generation, which induces flexible implicit distributions on generated action sequences. Different from existing action-prediction methods, our model does not require conditional action poses in the generation process, although it can be easily generalized to this setting. Within the core is a latent-action generator that learns smooth latent transitions, which are then fed to a shared decoder to generate final action sequences. Our model is formulated within the bi-directional GAN framework, which contains a sequence classifier that simultaneously learns action classification. Experiments are conducted on two accessible datasets, demonstrating the effectiveness of the proposed model, and obtaining better results compared to related baseline models.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Barsoum, E.; Kender, J.; and Liu, Z. 2017. Hp-gan: Probabilistic 3d human motion prediction via gan.

Bissacco, A. 2005. Modeling and learning contact dynamics in human motion. In *CVPR*.

Cai, H.; Bai, C.; Tai, Y.-W.; and Tang, C.-K. 2018. Deep video generation, prediction and completion of human action sequences. In *ECCV*.

Catalin Ionescu, Dragos Papava, V. O., and Sminchisescu, C. 2014. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE TPAMI*.

Denton, E., and Birodkar, V. 2017. Unsupervised learning of disentangled representations from video. In *NIPS*.

Donahue, J.; Krähenbühl, P.; and Darrell, T. 2017. Adversarial feature learning. In *ICLR*.

Du, Y.; Wang, W.; and Wang, L. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*.

Dumoulin, V.; Belghazi, I.; Poole, B.; Mastropietro, O.; Lamb, A.; Arjovsky, M.; and Courville, A. 2017. Adversarially learned inference. In *ICLR*.

Fragkiadaki, K.; Levine, S.; Felsen, P.; and Malik, J. 2015. Recurrent network models for human dynamics. In *ICCV*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

Graves, A. 2012. Sequence transduction with recurrent neural networks. In *ICML Representation Learning Workshop*.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *NIPS*.

Habibie, I.; Holden, D.; Schwarz, J.; Yearsley, J.; and Komura, T. 2017. A recurrent variational autoencoder for human motion synthesis. In *BMVC*.

Harvey, F. G., and Pal, C. 2018. Recurrent transition networks for character locomotion. In *SIGGRAPH Asia*.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional lstm-crf models for sequence tagging.

Jin, Y.; Zhang, J.; Li, M.; Tian, Y.; and Zhu, H. 2017. Towards the automatic anime characters creation with generative adversarial networks.

Judith Bütepage, Michael Black, D. K. H. K. 2017. Deep representation learning for human motion prediction and classification. In *IEEE CVPR*.

Ke, Q.; Bennamoun, M.; An, S.; Sohel, F.; and Boussaid, F. 2017. A new representation of skeleton sequences for 3d action recognition. In *CVPR*.

Kiasari, M. A.; Moirangthem, D. S.; and Lee, M. 2018. Human action generation with generative adversarial networks.

Kim, T. S., and Reiter, A. 2017. Interpretable 3d human action analysis with temporal convolutional networks. In *BNMW CVPR*.

Kovar, L.; Gleicher, M.; and Pighin, F. 2002. Motion graphs. In *SIGGRAPH*.

Kundu, J. N.; Gor, M.; and Babu, R. V. 2019. Bihmp-gan: Bidirectional 3d human motion prediction gan. In *AAAI*.

Lee, A. X.; Zhang, R.; Ebert, F.; Abbeel, P.; Finn, C.; and Levine, S. 2018. Stochastic adversarial video prediction.

Li, C.; Liu, H.; Chen, C.; Pu, Y.; Chen, L.; Henao, R.; and Carin, L. 2017. Alice: Towards understanding adversarial learning for joint distribution matching. In *NIPS*.

Liang, X.; Lee, L.; Dai, W.; and Xing, E. P. 2017. Dual motion gan for future-flow embedded video prediction. In *ICCV*.

Liu, J.; Shahroudy, A.; Xu, D.; and Wang, G. 2016. Spatio-temporal lstm with trust gates for 3d human action recognition. In *ECCV*.

Ma, L.; Jia, X.; Sun, Q.; Schiele, B.; Tuytelaars, T.; and Van Gool, L. 2017. Pose guided person image generation. In *NIPS*.

Martinez, J.; Black, M. J.; and Romero, J. 2017. On human motion prediction using recurrent neural networks. In *CVPR*.

Min, J., and Chai, J. 2012. Motiongraphs++: A compact generative model for semantic motion analysis and synthesis. In *ACM TOG*.

Oh, S. M.; Rehg, J. M.; Balch, T.; and Dellaert, F. 2005. Learning and inference in parametric switching linear dynamical systems. In *ICCV*.

Pascanu, R.; Mikolov, T.; and Bengio, Y. 2013. On the difficulty of training recurrent neural networks. In *ICML*.

Pavlović, V.; Rehg, J. M.; and MacCormick, J. 2001. Learning switching linear models of human motion. In *NIPS*.

Schuster, M., and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE TSP*.

Shahroudy, A.; Liu, J.; Ng, T.-T.; and Wang, G. 2016. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *CVPR*.

Sundermeyer, M.; Alkhouli, T.; Wuebker, J.; and Ney, H. 2014. Translation modeling with bidirectional recurrent neural networks. In *EMNLP*.

Sutskever, I.; Hinton, G.; ; and Taylor, G. 2008. The recurrent temporal restricted boltzmann machine. In *NIPS*.

Taylor, G. W., and Hinton, G. E. 2009. Factored conditional restricted boltzmann machines for modeling motion style. In *ICML*.

Taylor, G. W.; Hinton, G. E.; and Roweis, S. 2007. Modeling human motion using binary latent variables. In *NIPS*. MIT Press.

Urtasun, R.; Fleet, D. J.; Geiger, A.; Popović, J.; Darrell, T. J.; and Lawrence, N. D. 2008. Topologically-constrained latent variable models. In *ICML*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016. Generating videos with scene dynamics. In *NIPS*.

Walker, J.; Marino, K.; Gupta, A.; and Hebert, M. 2017. The pose knows: Video forecasting by generating pose futures. In *ICCV*.

Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Liu, G.; Tao, A.; Kautz, J.; and Catanzaro, B. 2018a. Video-to-video synthesis. In *NeurIPS*.

Wang, Y.; Gui, L.-Y.; Liang, X.; and Moura, J. M. F. 2018b. Adversarial geometry-aware human motion prediction. In *ECCV*.

Wang, Z.; Chai, J.; and Xia, S. 2018. Combining recurrent neural networks and adversarial training for human motion modelling, synthesis and control. In *https://arxiv.org/pdf/1806.08666.pdf*.

Wang, J. M.; Fleet, D. J.; and Hertzmann, A. 2007. Multifactor gaussian process models for style-content separation. In *ICML*.

Wang, J. M.; Fleet, D. J.; and Hertzmann, A. 2008. Gaussian process dynamical models for human motion. *IEEE TPAMI*.

Wichers, N.; Villegas, R.; Erhan, D.; and Lee, H. 2018. Hierarchical long-term video prediction without supervision. In *ICML*.

Xue, T.; Wu, J.; Bouman, K. L.; and Freeman, W. T. 2016. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*. MIT Press.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal gcns for skeleton-based action recognition. In *AAAI*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.