Bidirectional Long Short-Term Memory Variational Autoencoder

Henglin Shi
Henglin.Shi@oulu.fi
Xin Liu
Xin.Liu@oulu.fi
Xiaopeng Hong
Xiaopeng.Hong@oulu.fi
Guoying Zhao*
Guoying.Zhao@oulu.fi

Center for Machine Vision and Signal Analysis
University of Oulu
Oulu, Finland

Abstract

Variational Autoencoder (VAE) has achieved promising success since its emergence. In recent years, its various variants have been developed, especially those works which extend VAE to handle sequential data [1, 2, 5, 7]. However, these works either do not generate sequential latent variables, or encode latent variables only based on inputs from earlier time-steps. We believe that in real-world situations, encoding latent variables at a specific time-step should be based on not only previous observations, but also succeeding samples. In this work, we emphasize such fact and theoretically derive the bidirectional Long Short-Term Memory Variational Autoencoder (*bLSTM-VAE*), a novel variant of VAE whose encoders and decoders are implemented by bidirectional Long Short-Term Memory (bLSTM) networks. The proposed *bLSTM-VAE* can encode sequential inputs as an equal-length sequence of latent variables. A latent variable at a specific time-step is encoded by simultaneously processing observations from the first time-step till current time-step in a forward order and observations from current time-step till the last time-step in a backward order. As a result, we consider that the proposed *bLSTM-VAE* could learn latent variables reliably by mining the contextual information from the whole input sequence. In order to validate the proposed method, we apply it for gesture recognition using 3D skeletal joint data. The evaluation is conducted on the ChaLearn Look at People gesture dataset and NTU RGB+D dataset. The experimental results show that combining with the proposed *bLSTM-VAE*, the classification network performs better than when combining with a standard VAE, and also outperforms several state-of-the-art methods.

1 Introduction

Because of its great success, Variational Autoencoder (VAE) has been extensively adopted in the communities of computer version and natural language processing. One of its major strengths is that it is trained to robustly encode latent variables based on input samples. During training, latent variables are encoded by combining means and variances sampled

* indicates the corresponding author.

from the input data, especially, variances are weighted by a random noise. Thus, the VAE is noise-resistant. During testing, sampled variances are eliminated so that encoding is only based on sampled means. From a signal decomposition point of view, the mean and variance are considered as the "low-frequency" approximation part and "high-frequency" detailed part of the original signal, respectively. Thus, latent variables encoded from sampled means could be considered as a robust generalization of the original data.

However, a critical assumption of VAE is that the input data should be independent and identically distributed (i.i.d.), which impedes its further applications in time series analysis. In recent years, several works have been proposed to eliminate such assumption [1, 2, 5]. For example, in [1, 5], encoders and decoders are implemented by Recurrent Neural Networks (RNNs), and latent variables are encoded based on the output of RNN at the last time-step. However, in this case, only one hidden variable is encoded based on the global context of the given sequence. Moreover, [2] considered encoding sequential latent variables. In [2], latent variables at a specific time-step are encoded based on observations from the first time-step till current time-step. However, sampling hidden variables at a specific time-step only based on early observations may lose useful contextual information from succeeding samples of the sequence. Based on such concern, we derive the bidirectional Long Short-Term Memory VAE (*bLSTM-VAE*) which could encode latent variables by forwardly tracing previous observations and backwardly tracing later observations.

Similar to the standard VAE, the proposed *bLSTM-VAE* also contains a mean encoder, a variance encoder, and a decoder, which are theoretically derived to be implemented by bidirectional Long Short-Term Memory (bLSTM) networks. Thus, *bLSTM-VAE* retains the VAE's feature of encoding robust latent variables based on sampled means from the input data, and further endowed by bLSTM to learn the global contextual information from the whole sequence. As a result, the proposed *bLSTM-VAE* could be an effective tool to extract features from data with high variations, e.g. human 3D skeletal joint data.

Human 3D skeletal joint data is one of the most important modalities for human behavior modeling and analysis because of its various advantages. For example, it is a scene-invariant representation of the body which is the object of the analysis interest. However, there are still two main difficulties for effectively utilizing such data. On one hand, skeletons extracted by Kinect are not always reliable since which could be distorted or even not extracted from the body. On the other hand, as a geometric representation, skeletons have subject variations because different subjects may have different characters of joints (e.g. lengths). Thus, there is a concern that the proposed *bLSTM-VAE* can effectively handle such sequential data with large variations. For verification, we further develop an end-to-end model for skeleton based gesture recognition. The proposed model consists of a *bLSTM-VAE* feature learning network for encoding skeletons into a latent space, and a multi-layer bLSTM gesture classification network whose inputs are encoded latent sequences. This model is evaluated on ChaLearn Look at People (LAP) 2014 gesture dataset [4] and NTU RGB+D action dataset [19].

This work makes following contributions. 1) we derive the *bLSTM-VAE* under the theoretical framework of VAE; 2) we propose a skeleton based gesture recognition method which jointly encodes features using *bLSTM-VAE* and recognizes gestures with a *bLSTM* classifier; 3) we conduct extensive experimental evaluations on two large scale gesture databases.

The rest of the paper is organized as follows. Section 2 reviews recent advances of VAE and related works for skeleton based gesture analysis. Section 3 introduces the proposed *bLSTM-VAE*. Section 4 presents the experimental results of the proposed method on two large scale gesture recognition datasets and comparisons with related methods. Section 5 discusses and concludes the results of this paper.

2 Related Work

Variational Autoencoder [11, 18] is proposed to approximate an intractable true posterior $P(z|x)$ with a constructed distribution $Q(z|x)$ by minimizing the Kullback-Leibler (KL) divergence between them. In computer vision community, VAE has been used in many tasks, such as hand pose estimation [25], action unit intensity estimation [21], and video trajectory prediction [24]. However, VAE assumes input samples should be i.i.d. which cannot be fulfilled by time series data. In recent years, researchers attempted to extend VAE to handle sequential data. In [5] and [1], latent variables are encoded based on the last output of the encoding RNN whose input is a sequence. During the decoding phase, a decoding RNN receives the latent variable at the first time-step and further generates the reconstructed sequence. In such condition, encoded hidden variables are global respect to the whole input sequence, which makes this model inapplicable for encoding frame-wise features. [2] is proposed to sequentially encode time-step specific latent variables, i.e. encoding z_t from observation sequence $x_{1:t}$ using RNNs, which enables us to encode frame-wise features. However, a limitation of this work is that z_t is encoded only based on previous history observations. However, sometimes we can encode more accurate z_t if associating the future observations. Based on such concern, we propose a novel variant of VAE to encode latent variables not only based on previous observations, but also future samples.

Skeleton based gesture analysis is one of most popular tasks in computer vision community. During the past decade, various handcrafted features have been developed [17, 22, 23, 28, 29]. Moreover, deep neural networks have also been widely used in gesture recognition tasks and achieved inspiring successes, for example Restricted Boltzmann Machines (RBMs), Recurrent Neural Networks (RNNs), and LSTMs [3, 13, 14, 15, 20, 26, 27]. However, except work [14] which considered the unreliability of collected skeleton data, most other works were mainly general gesture classification methods. In this work, beside deriving the theoretical background of *bLSTM-VAE*, we also conduct experiments to discuss it from a feature learning point of view to show that *bLSTM-VAE* as an effective tool for encoding skeleton features while coping with the data unreliability and temporal variation caused by different subjects.

3 Proposed method

Figure 1 describes the encoding and decoding processes at time-step t. For a given observation sequence $x_{1:T}$ whose temporal length is T, a latent variable z_t is encoded by forwardly processing samples from x_1 to x_t and backwardly processing samples from x_T to x_t. In this section, firstly we derive the proposed *bLSTM-VAE* as well as its optimization criterion, and describe how the encoder and decoder can be implemented by bLSTM networks. In addition, we introduce the detail of the multi-layer bLSTM network.

3.1 Bidirectional Variational Autoencoder

Given a sequential observation $x_{1:T}$ which is generated by a latent sequence $z_{1:T}$. Generally we can define $P(z_t|x_{1:T})$, the probability of sampling z_t at time-step t which could contribute to generate $x_{1:T}$. More concretely, we define the sampling process is performed by two steps: **a).** Forwardly tracing observations from x_1 to x_t; **b).** Backwardly tracing observations from x_T to x_t, where x_T indicates the last frame. Then we can rewrite $P(z_t|x_{1:T})$ as $P(z_t|\overrightarrow{x_{1:t}}, \overleftarrow{x_{t:T}})$.

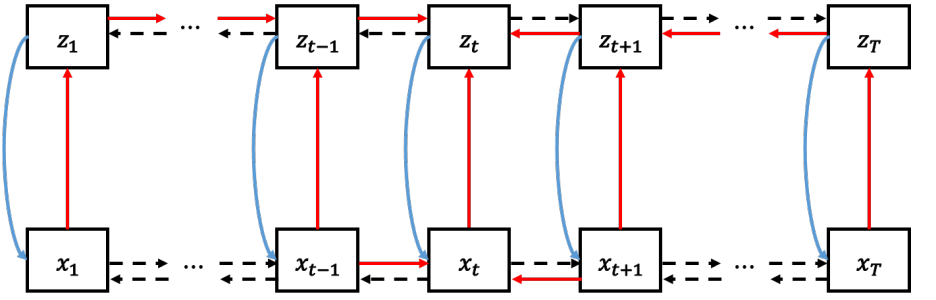

Figure 1: Graphical Representation of the proposed Bidirectional Long Short-Term Memory Variational Autoencoder (*bLSTM-VAE*) for encoding latent variables at the time-step t. Solid red lines denote the encoding process, in which arrows to the right denote the forward processing and arrows to the left denote the backward processing. Solid blue lines describe the decoding process, similar with encoding process, right arrows denote forward processing and left arrows denote backward processing. Both encoding and decoding processes can be implemented by *bLSTM* networks. Besides, arrows with dash lines denote operations are not active when encoding and decoding variables at current time-step.

Here $\overrightarrow{x_{1:t}}$ denotes samples processed in a forward order, and $\overleftarrow{x_{t:T}}$ denotes samples processed in a backward order.

Adopting the philosophy of VAE, one may hope to find a distribution $Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})$ to approximate $P(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})$ by minimizing the Kullback-Leibler (KL) divergence between them. Starting from the definition of KL divergence we can have:

$$\begin{aligned} &D_{KL}[Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})||P(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})] \\ &= \int Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}}) \log \frac{Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})}{P(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})} dz_t \\ &= \int Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}}) \log \frac{Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})P(\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})}{P(\overrightarrow{x_t},\overleftarrow{x_t}|z_t,\overrightarrow{x_{1:t-1}},\overleftarrow{x_{t+1:T}})P(\overrightarrow{x_{1:t-1}},\overleftarrow{x_{t+1:T}}|z_t)P(z_t)} dz_t \end{aligned} \tag{1}$$

From Figure 1 we can see that forwardly generating $x_{1:t-1}$ and backwardly generating $x_{t+1:T}$ do not depend on z_t. Thus $P(\overrightarrow{x_{1:t-1}},\overleftarrow{x_{t+1:T}}|z_t)$ is equal to $P(\overrightarrow{x_{1:t-1}},\overleftarrow{x_{t+1:T}})$ so that Equation (1) can be rewritten to

$$\begin{aligned} &\log \frac{P(\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})}{P(\overrightarrow{x_{1:t-1}},\overleftarrow{x_{t+1:T}})} - D_{KL}[Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})||P(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})] \\ &= E_{z_t\sim Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})} \log P(x_t|z_t,\overrightarrow{x_{t-1}},\overleftarrow{x_{t+1}}) - D_{KL}[Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})||P(z_t)], \end{aligned} \tag{2}$$

Because $\log \frac{P(\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})}{P(\overrightarrow{x_{1:t-1}},\overleftarrow{x_{t+1:T}})}$ is composed by the likelihood of samples from the observed sequence, minimizing $D_{KL}[Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})||P(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})]$ is equivalent to maximizing the terms right to the equal sign, which is also called as the *Variation Lower Bound* [□]:

$$L(\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}}) = E_{z_t\sim Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})} \log P(x_t|z_t,\overrightarrow{x_{t-1}},\overleftarrow{x_{t+1}}) - D_{KL}[Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})||P(z_t)] \tag{3}$$

The second term of the *Variation Lower Bound* is assumed as a conditional Gaussian distribution whose mean and variance are $\mu_t = \mu(\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})$ and $\sigma_t = \sigma(\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})$, respectively.

$\mu(\cdot)$ and $\sigma(\cdot)$ can be implemented by any functions, and we use bidirectional LSTM in this study. Thus we have

$$Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}}) = N(z_t;\mu_t,\sigma_t). \tag{4}$$

By assuming $P(z_t) = N(z_t;0,I)$, the analytical solution of $D_{KL}[Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})||P(z_t)]$ is given according to [10]:

$$D_{KL}[Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})||P(z_t)] = -\frac{1}{2}\sum_{j=1}^{J}(1+\log(\sigma_t^j)^2-(\mu_t^j)^2-(\sigma_t^j)^2), \tag{5}$$

where J indicates the number of dimension of μ_t and σ_t.

Additionally, the first term in Eq. (3) can be solved by sampling L samples for z_t such that

$$E_{z_t\sim Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})}\log P(x_t|z_t,\overrightarrow{x_{t-1}},\overleftarrow{x_{t+1}}) \approx \frac{1}{L}\sum_{l=1}^{L}\log P(x_t|z_t^l,\overrightarrow{x_{t-1}},\overleftarrow{x_{t+1}}), \tag{6}$$

z_t^l is sampled from $Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})$ by $z_t^l = \mu_t + \sigma_t \circ \varepsilon^l$, where $\varepsilon^l \sim N(0,I)$ and $\circ$ denotes the element-wise product. In our case, z_t and x_t are real valued variables, so $P(x_t|z_t^l,\overrightarrow{x_{t-1}},\overleftarrow{x_{t+1}})$ can be assumed as a conditional Gaussian whose mean is $f_{dec}(z_t^l,\overrightarrow{x_{t-1}},\overleftarrow{x_{t+1}})$ and variance is I. Thus the analytical form of first term of the *VariationalLowerBound* can be given as:

$$E_{z_t\sim Q(z_t|\overrightarrow{x_{1:t}},\overleftarrow{x_{t:T}})}\log P(x_t|z_t,\overrightarrow{x_{t-1}},\overleftarrow{x_{t+1}}) \approx -\frac{1}{L}\sum_{l=1}^{L}\left\|x_t - f_{dec}(z_t^l,\overrightarrow{x_{t-1}},\overleftarrow{x_{t+1}})\right\|^2 + c \tag{7}$$

Finally, based on the analytical form of the two terms, the *Variation Lower Bound* can be optimized by stochastic optimization methods, such as Adam [10]. Each of the three functions $\mu(\cdot)$, $\sigma(\cdot)$, and $f_{dec}(\cdot)$ can be implemented by a bidirectional LSTM network which is illustrated in next section.

3.2 Multi-layer bidirectional Long Short-Term Memory

In this study, the bidirectional LSTM is constructed by two multi-layer LSTM networks and one fully connected layer. One LSTM is a forward LSTM which processes samples from x_1 to x_T. The other one is a backward LSTM which processes samples from the inversed order. Finally, the outputs of the two LSTM networks are concatenated together and fed into the fully connected layer.

Consider a N-layer bidirectional LSTM, the forward pass of the n^{th} layer of forward network and backward network are respectively explained by Equations (8)-(10) and (11)-(13). $h_t^{(n-1)}$ represents the output from the $(n-1)^{th}$ (previous) layer at time-step t, and $h_t^{(n-1)}$ equals to the raw input when $n = 1$.

$$c_{f,t}^{(n)} = f_{f,t}^{(n)} \circ c_{f,t-1}^{(n)} + i_{f,t}^{(n)} \circ g_{f,t}^{(n)} \tag{8}$$

$$h_{f,t}^{(n)} = o_{f,t}^{(n)} \circ tanh\left(c_{f,t}^{(n)}\right) \tag{9}$$

$$\begin{bmatrix} i_{f,t}^{(n)} \\ f_{f,t}^{(n)} \\ o_{f,t}^{(n)} \\ g_{f,t}^{(n)} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \begin{bmatrix} W_{f,i,x}^{(n)} & W_{f,i,h}^{(n)} \\ W_{f,f,x}^{(n)} & W_{f,f,h}^{(n)} \\ W_{f,o,x}^{(n)} & W_{f,o,h}^{(n)} \\ W_{f,g,x}^{(n)} & W_{f,g,h}^{(n)} \end{bmatrix} \begin{bmatrix} h_t^{(n-1)} \\ h_{t-1}^{(n)} \end{bmatrix} \tag{10}$$

$$c_{b,t}^{(n)} = f_{b,t}^{(n)} \circ c_{b,t+1}^{(n)} + i_{b,t}^{(n)} \circ g_{b,t}^{(n)} \tag{11}$$

$$h_{b,t}^{(n)} = o_{b,t}^{(n)} \circ tanh\left(c_{b,t}^{(n)}\right) \tag{12}$$

$$\begin{bmatrix} i_{b,t}^{(n)} \\ f_{b,t}^{(n)} \\ o_{b,t}^{(n)} \\ g_{b,t}^{(n)} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \begin{bmatrix} W_{b,i,x}^{(n)} & W_{b,i,h}^{(n)} \\ W_{b,f,x}^{(n)} & W_{b,f,h}^{(n)} \\ W_{b,o,x}^{(n)} & W_{b,o,h}^{(n)} \\ W_{b,g,x}^{(n)} & W_{b,g,h}^{(n)} \end{bmatrix} \begin{bmatrix} h_t^{(n-1)} \\ h_{t+1}^{(n)} \end{bmatrix} \tag{13}$$

The final output y_{b-LSTM} of the bidirectional LSTM at each time-step t will only rely on the output of the last layer of both forward and backward networks as explained by Equation (14), where $h_{f,t}^{(N)}$ and $h_{b,t}^{(N)}$ represent the output from the last (the N^{th}) layer of forward network and backward network at time-step t, respectively.

$$y_{b-LSTM} = ReLU(W \begin{bmatrix} h_{f,t}^{(N)} \\ h_{b,t}^{(N)} \end{bmatrix} + b) \tag{14}$$

4 Experimental results

In this section, firstly we introduce selected datasets in our experiments. Since deep learning based methods require large amount of data, we use two large scale gesture datasets: the ChaLearn Looking at People (LAP) gesture dataset [4] and the NTU RGB+D action recognition dataset [19]. Moreover, we develop a skeleton based gesture recognition model which integrates the proposed *bLSTM-VAE* and a multi-layer bLSTM classification network, and make it as our target method. We name our target method as *bLSTM-VAE+bLSTM*. In order to validate the effectiveness of the proposed *bLSTM-VAE*, we also design two baseline models for comparison: 1) *bLSTM*, and 2) *FC-VAE+bLSTM* which is based on the standard VAE. Implementation details of our target method and the two baseline methods are presented in section 4.2. Lastly, the experimental results are discussed in section 4.3.

4.1 Datasets

4.1.1 ChaLearn LAP gesture dataset

The ChaLearn LAP gesture recognition dataset is a multi-modal dataset designed for gesture detection (or spotting) and recognition. This dataset provides RGB videos, depth maps, and skeletal joint positions. The resolution of RGB videos and depth maps is 640×480. For skeletons, each frame records the position as well as rotation in the real world space and the position in the screen space of 20 joints including *head*, *left(L)/center(C)/right(R) shoulders*, *spine*, *L/R elbows*, *L/R wrists*, *L/R hands*, *L/C/R hips*, *L/R knees*, *L/R ankles*, and *L/R feet*.

This dataset includes 940 sequences for each modality and each sequence contains 10 to 20 Italian cultural gestures. Totally there are 13585 gesture instances from 20 classes. This dataset officially provides a standard evaluation protocol in which 470 sequences are pre-allocated for training, 230 for validation and 240 sequences for testing. We follow this provided protocol in this evaluation.

4.1.2 NTU RGB+D action recognition dataset

We also use the NTU RGB+D action recognition dataset to evaluate the proposed method. NTU RGB+D dataset is a large scale multi-modal and multi-view datasets which provides four modalities including RGB videos, depth maps, infrared frames, and skeletal joint positions. The RGB video is full HD with the resolution of 1920×1080. Additionally, depth maps and infrared frames are collected with the resolution of 512×424. Besides, skeleton data is collected from 25 joints. Compared with ChaLearn LAP gesture dataset, NTU has two more joints (HAND TIP, and THUMB) for each hand and one more joint at the neck.

In total, this dataset collects 56880 action sequences from 40 subjects, which are annotated into 60 classes. The provided evaluation protocol is cross-subject evaluation. In this part of the experiment, we adopted the protocol from [19] which defines action sequences performed by subjects $1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35, 38$ are training data and the rest are testing data.

4.2 Experiment Implementation

4.2.1 Network specification

Firstly we introduce the target model *bLSTM-VAE+bLSTM* which consists of two parts: a *bLSTM-VAE* for feature extraction; and a 3-layer bLSTM gesture classification network which receives the learned latent variable from the *bLSTM-VAE* as input. Additionally, the baseline *FC-VAE+bLSTM* is implemented in the same way as the target model, but the *bLSTM-VAE* part is replaced by a standard VAE whose encoder and decoder are implemented by fully connected layers in order to compare the difference between *bLSTM-VAE* and standard VAE. Besides, the other baseline model *bLSTM* is implemented by the same 3-layer bLSTM gesture classification network for comparing the gesture classification result between using the raw input and using the feature extracted by two different VAEs.

Since the numbers of selected joints are different for the two datasets, the numbers of neurons for each component of each model are also different. The input dimension of ChaLearn is 36 for 12 selected joints, and the input dimension of NTU RGB+D is 75 since all 25 joints are selected. As a result, the neuron number of the network when evaluating the NTU RGB+D is set to be higher than the one used for evaluating the ChaLearn LAP dataset. Specifically, the dimension of encoders and decoders for evaluating ChaLearn and NTU are set to 128 and 256, respectively. The neuron number of the 3-layer bLSTM gesture classification network for each dataset is equally set to 512.

4.2.2 Training Procedures

Training the baseline *bLSTM* is straightforward. We just simply feed the skeleton sequence and minimize the classification loss. The learning rate is 0.0005 with a decay factor of 0.999 for each 5 training epochs. The network is trained till the loss converges such that the training loss and accuracy tend to be stable.

Moreover, training the target model *bLSTM-VAE+bLSTM* and the baseline model *FC-VAE+bLSTM* requires two steps. Firstly, the VAE feature learning network is solely pretrained under an unsupervised condition (labels are not involved), such that all samples are involved in this phase. The optimization criteria are derived KL divergence and the reconstruction error. In this step, the learning rate is 0.0005 with a decay factor of 0.999 for each 5 training epochs. After the network is sufficiently converged, the latent variable of VAE is

Table 1: Recognition Accuracy on ChaLearn LAP Dataset. Accuracies of methods 4 and 6 marked with * have used at least RGB frames and skeletons. Moreover, accuracies of methods 4 and 5 marked with dagger were developed for gesture detection task which are more difficult than solely classifying gestures. Besides, accuracy marked with underline is the best result among selected methods.

Methods	*Accuracies*
1. Lie Group [23]	79.20%
2. Deep LSTM [8]	82.00%
3. HMM-DBN [26]	83.60%
4. HMM-DBN-ext [27]	86.40%*†
5. ModDrop [16]	<u>93.10%</u>*
6. Baseline 1: bLSTM [6]	90.80%†
7. Baseline 2: FC-VAE [10] + bLSTM	92.46%
8. **bLSTM-VAE + bLSTM (ours)**	**92.88%**

treated as extracted features and further fed into the bLSTM classification network for joint fine-tuning. Now the optimization criterion is only the classification loss. In this stage, only training samples are involved since the fine-tuning is supervised. The learning rate during fine-tuning is 0.0002 with a decay factor of 0.999 for each 2 training epochs.

4.3 Experimental Results

Table 1 presents the evaluation results on the ChaLearn LAP dataset. In addition to the proposed target model and two baseline models, we also select five comparison methods. Lie Group [23], HMM-DBN [26], and HMM-DBN-ext (multi-modal version) [27] are selected because they are feature learning related methods which are similar to this work. Moreover, [8] is compared because LSTM is closely related to our work. Lastly, ModDrop [16] is selected for comparison because it achieves currently the highest performance on this dataset based on the authors' best knowledge.

According to Table 1, the proposed method achieves the recognition accuracy of 92.88% which outperforms most of listed methods, except the ModDrop [16] which achieves the highest 93.10%. However, compared with the proposed method which use only the skeleton data, ModDrop used multi-modal cues including RGB frames and skeletons. Compared with baseline 2, *FC-VAE + bLSTM* which achieves 92.46%, the proposed method does not provide significant improvement. The main reason could be that the evaluation protocol is not a subject independent one, so that there are no challenges from the temporal variation of different subjects. Then different subjects can be learned by the bLSTM classification network, which dilutes the effectiveness of *bLSTM-VAE*. Moreover, if comparing with baseline model 2, *bLSTM* which achieves 90.80%, both two VAE based methods can provide obvious improvements. As a result, VAE, especially the proposed *bLSTM-VAE* can effectively perform a feature learning task and improve the classification model.

Experimental results on NTU RGB+D dataset are presented in Table 2, associated with as list of selected comparison methods. Firstly, Lie Group [23] and Lie Net [9] are compared because they were also conducted for feature learning. Moreover, HBRNN-L [3], Deep LSTM [8], Part-aware LSTM [19], ST-LSTM+TG [14] and Temporal Sliding LSTM networks [12] are compared because they have also used LSTMs.

Table 2: Recognition Accuracy on NTU RGB+D Dataset. Accuracy marked with underline is the best result among selected methods.

Methods	*Accuracies*
1. Lie Group [23], reported [19]	50.08%
2. HBRNN-L [8], reported by [19]	59.07%
3. Deep LSTM [8], reported by [19]	60.69%
4. LieNet [9]	61.37%
5. Part-aware LSTM (dataset baseline) [19]	62.93%
6. ST-LSTM+TG [12]	69.20%
7. Temporal Sliding LSTM networks [12]	<u>74.60%</u>
8. Baseline 1: bLSTM [6]	65.22%
9. Baseline 2: FC-VAE [11] + bLSTM	67.42%
10. **bLSTM-VAE + bLSTM (ours)**	**71.60%**

The experiment results show that our method achieves the recognition accuracy of 71.60%, which outperforms most of listed methods, except the Temporal Sliding LSTM network [12] which achieves the accuracy of 74.60%. However, compared with our method which takes the raw joints as the input and perform end-to-end gesture classification, the Temporal Sliding LSTM network used a series of processing tasks. Firstly it involved a data preprocessing step to transform skeletons for aligning scales, rotations, as well as translations. Moreover, rather than using the skeleton joint as the network input, it used the salient motion feature extracted from the preprocessed data. Unlike the Temporal Sliding LSTM network, our model is more convenient and can also provide comparable result. Moreover, the proposed method significantly outperforms the baseline model 2, *FC-VAE + bLSTM*. This might because the evaluation protocol is cross-subject so that temporal variations from subjects is not visible to the classification network. However, the *bLSTM-VAE* can model the temporal variations, which is difficult for *FC-VAE*. Furthermore, the proposed method also significantly outperforms the baseline model 1, a sole *bLSTM* classification network. Thus, the experimental results show that the proposed *bLSTM-VAE* is more effective to encode features from data with high variations and can improve the classification model.

5 Conclusion and future work

In this paper, we derive a variant of VAE, which is called *bLSTM-VAE*. Compared with the standard VAE, the proposed one can encode latent sequences based on sequential observations rather than i.i.d variables. Endowed by the bLSTM, the encoding and decoding processes can fully utilize the contextual information from the corresponding sequence. Based on the evaluation results on two large scale gesture datasets, the proposed *bLSTM-VAE* is effective for feature learning from data with high variations.

In the future, we will extend our work to investigate the multi-modal data interaction under the architecture of Conditional Variational Autoencoder (CVAE), in which the input sample and the reconstruction are not needed to be the same.

Acknowledgement. This work was supported by the Academy of Finland, Tekes Fidipro program (Grant No. 1849/31/2015) and Business Finland project (Grant No. 3116/31/2017), Infotech Oulu, and Nokia visting professor grant. The authors wish to acknowledge CSC – IT Center for Science, Finland, for computational resources.

References

[1] S. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. In *CoNLL*, 2016.

[2] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. In *NIPS*, 2015.

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

[4] S. Escalera, X. Baró, J. Gonzalez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon. Chalearn looking at people challenge 2014: Dataset and results. In *ECCVW*, 2014.

[5] O. Fabius and J. R. van Amersfoort. Variational recurrent auto-encoders. In *ICLRW*, 2015.

[6] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.

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

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

[9] Z. Huang, C. Wan, T. Probst, and L. Van Gool. Deep learning on Lie groups for skeleton-based action recognition. In *CVPR*, 2017.

[10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[11] D. P. Kingma and M. Welling. Variational recurrent auto-encoders. In *ICLR*, 2014.

[12] I. Lee, D. Kim, S. Kang, and S. Lee. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *ICCV*, 2017.

[13] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu. Online human action detection using joint classification-regression recurrent neural networks. In *ECCV*, 2016.

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

[15] J. Liu, G. Wang, P. Hu, L. Duan, and A. C. Kot. Global context-aware attention lstm networks for 3D action recognition. In *CVPR*, 2017.

[16] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout. Moddrop: adaptive multi-modal gesture recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(8):1692–1706, 2016.

[17] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. *Journal of Vis. Commun. Image Represent.*, 25(1):24–38, 2014.

[18] Danilo J. R., Shakir M., and Daan W. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[19] A. Shahroudy, J. Liu, T. Ng, and G. Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *CVPR*, 2016.

[20] G. W. Taylor, G. E. Hinton, and S. Roweis. Modeling human motion using binary latent variables. In *NIPS*, 2007.

[21] D. L. Tran, R. Walecki, O. Rudovic, S. Eleftheriadis, B. Schuller, and M. Pantic. Deepcoder: Semi-parametric variational autoencoders for automatic facial action coding. In *ICCV*, 2017.

[22] R. Vemulapalli and R. Chellapa. Rolling rotations for recognizing human actions from 3D skeletal data. In *CVPR*, 2016.

[23] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3D skeletons as points in a Lie group. In *CVPR*, 2014.

[24] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016.

[25] C. Wan, T. Probst, L. Van Gool, and A. Yao. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *CVPR*, 2017.

[26] D. Wu and L. Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In *CVPR*, 2014.

[27] D. Wu, L. Pigou, P. Kindermans, N. D. Le, L. Shao, J. Dambre, and J. Odobez. Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(8):1583–1597, 2016.

[28] L. Xia, C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *CVPRW*, 2012.

[29] X. Yang and Y. Tian. Eigenjoints-based action recognition using naive-Bayes-nearest-neighbor. In *CVPRW*, 2012.