PAPER Special Section on Machine Vision and its Applications

Deep State-Space Model for Noise Tolerant Skeleton-Based Action Recognition

Kazuki KAWAMURA[†a)], ***Nonmember***, **Takashi MATSUBARA**[†b)], ***and*** **Kuniaki UEHARA**[†c)], ***Members***

SUMMARY Action recognition using skeleton data (3D coordinates of human joints) is an attractive topic due to its robustness to the actor's appearance, camera's viewpoint, illumination, and other environmental conditions. However, skeleton data must be measured by a depth sensor or extracted from video data using an estimation algorithm, and doing so risks extraction errors and noise. In this work, for robust skeleton-based action recognition, we propose a deep state-space model (DSSM). The DSSM is a deep generative model of the underlying dynamics of an observable sequence. We applied the proposed DSSM to skeleton data, and the results demonstrate that it improves the classification performance of a baseline method. Moreover, we confirm that feature extraction with the proposed DSSM renders subsequent classifications robust to noise and missing values. In such experimental settings, the proposed DSSM outperforms a state-of-the-art method.

key words: *deep learning, skeleton, action recognition, state-space models*

1. Introduction

Human action recognition is an important topic in computer vision with applications for intelligent video surveillance, robotics, and human–computer interaction [1]. Studies in the past few decades have focused on action recognition based on RGB video recorded with 2D cameras. Because these approaches are sensitive to environmental conditions such as the actor's appearance, camera's viewpoint, and illumination, they require a huge training dataset and considerable computational resources for robust recognition. Action recognition based on skeleton data is thus a promising candidate to overcome this problem, and it has recently garnered considerable attention in the field. Skeleton data is a sequence of 3D locations of human body joints, and it is independent from the aforementioned environmental conditions for RGB video. 3D skeleton data can be obtained using cost-effective depth sensors such as Microsoft Kinect [2] and powerful algorithms to estimate human poses [3]. To analyze 3D skeleton data, several methods of extracting handcrafted features have been proposed [4]–[7]. However, these features often depend on a specific dataset; they are not universal.

Recent progress in deep learning has shown outstanding performance in computer vision, and several excellent methods have been proposed in the field of action recognition by employing recurrent neural networks (RNNs), which excel at modeling time series data [8]–[10]. Most of these models employ discriminative approaches based on handcrafted parts, or they are designed to specialize at handling whole-body skeleton data. Hence, despite their high performance with skeleton-based action recognition, these approaches are vulnerable to noise and missing values [11].

To overcome this issue, we propose a new deep state-space model (DSSM). The DSSM is a generative state-space model that uses deep neural networks (DNNs) for robust feature extraction. The proposed DSSM infers the underlying dynamics of given sequential skeleton data as a sequence of extracted features. Generative models are known to be robust to noise and overfitting, and they can handle missing values, provided that their architecture is appropriate [12], [13]. In particular, generative probabilistic models with latent variables, such as the variational autoencoder (VAE) [14], have been applied in many types of research on noise removal and missing value imputation [15], [16]. As with many other generative models that extend the VAE, the proposed DSSM embeds the input skeleton data into low-dimensional latent space using a probabilistic encoder, and then reconstructs the skeleton data from latent space using a probabilistic decoder. This allows us to learn a robust latent representation of the noise in low-dimensional latent space. Moreover, the proposed DSSM uses the skeleton information from the previous timestep $t-1$ to generate the skeleton at timestep t. Hence, it does not need to encode the body size and orientation based on the subject and camera. For this reason, the extracted features are expected to be robust to the subject's body size, orientation, and noise. Therefore, subsequent classifications are relatively more refined than direct classifications.

The rest of this paper is organized as follows. Section 2 introduces related work on skeleton-based action recognition. In Sect. 3, we illustrate the details of the proposed method. Experimental results and discussions are presented in Sect. 4. Section 5 offers our conclusions.

2. Related Work

2.1 Skeleton-Based Action Recognition

For action recognition, deep learning models such as recurrent neural networks (RNNs) [17], long short-term

Manuscript received August 31, 2019.
Manuscript revised January 31, 2020.
Manuscript publicized March 18, 2020.

[†]The authors are with the Faculty of Engineering and the Graduate School of System Informatics, Kobe University, Kobe-shi, 657–8501 Japan.
a) E-mail: kkawamura@ai.cs.kobe-u.ac.jp
b) E-mail: matsubara@phoenix.kobe-u.ac.jp
c) E-mail: uehara@kobe-u.ac.jp

DOI: 10.1587/transinf.2019MVP0012

Copyright © 2020 The Institute of Electronics, Information and Communication Engineers

memory (LSTM) [18], and convolutional neural networks (CNNs) [19] have been employed to build models of the spatio-temporal dynamics of skeleton data. In this section, we briefly review recent action recognition methods based on RNNs and LSTM.

Du *et al.* [8] proposed a three-step method based on a hierarchical RNN network. First, the whole human skeletal structure is divided into five parts, that is, two arms, two legs, and the trunk, according to human anatomy. These parts are separately fed to the corresponding bidirectional RNN (BiRNN). Second, the five outputs are combined to represent the upper and lower body parts, and each is fed to another RNN. Finally, the two outputs are combined and fed to the last RNN. Then, a fully connected layer and a softmax layer are applied to obtain the action classification to which the given sequence belongs.

Veeriah *et al.* [20] proposed a differential RNN by selecting frames containing distinguishable spatio-temporal information for different actions. Their method employs derivatives of memory states to explore salient behavioral patterns and includes a novel gating mechanism for LSTM.

Zhu *et al.* [21] introduced an internal dropout mechanism applied to LSTM gates for stronger regularization. To further regularize learning, a regularization term is added to the cost function of the network, forcing the model to learn the co-occurrence relations among the joints.

Shahroudy *et al.* [9] proposed a part-aware LSTM. The memory cell of the LSTM is divided into subcells corresponding to body parts, and the network is encouraged to learn the context representation of each body part independently. Output gates are shared among the body parts and learned by concatenating multiple memory subcells.

Liu *et al.* [10] proposed a spatio-temporal LSTM. To capture the dependency between joints in the spatial domain, it expresses adjacent joints as a tree structure. In addition, they introduced a new gating mechanism to address noise and occlusion in skeleton data.

When considering real applications, skeleton data is prone to noise and missing values, which occur during the process of generating skeleton data. However, previous studies do not sufficiently verify the robustness of their models to noise and missing values, despite the fact that these discriminative approaches are known to be vulnerable to noise; many previous studies reported that only small perturbations can lead misclassification [11]. By contrast, it is known that generative models, that is, models of the probability distribution $p(\boldsymbol{x})$ of observation data $\boldsymbol{x}$, are resistant to noise and missing values, and are less likely to suffer from overfitting [12]. Given the above, we here propose a new DSSM, a kind of generative state-space model based on deep learning.

2.2 Deep Generative Model

A generative model is a model of the process of generating observable data $p(\boldsymbol{x})$. In particular, a latent variable model is built under the assumption that observation data $\boldsymbol{x}$ is generated from an unobservable latent variable $\boldsymbol{z}$:

$$p_\theta(\boldsymbol{x}) = \int_z p_\theta(\boldsymbol{x}|\boldsymbol{z}) p_\theta(\boldsymbol{z}), \tag{1}$$

where $p(\boldsymbol{x}|\boldsymbol{z})$ and $p(\boldsymbol{z})$ denote the generation process and a prior distribution of the latent variable $\boldsymbol{z}$, respectively. Conventional generative models such as naive Bayes and the Gaussian mixture model are successful on toy benchmark tasks [12], but they encounter difficulty when expressing the generation process $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ of practical data. To leverage the flexibility of deep learning, Kingma *et al.* [14] proposed the variational autoencoder (VAE), where the generation process $p(\boldsymbol{x}|\boldsymbol{z})$ is expressed using a DNN. Using the variational approximation, the model evidence $\log p_\theta(\boldsymbol{x})$ is bounded as

$$\begin{aligned} \log p_\theta(\boldsymbol{x}) &\geq \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right] - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})\|p_\theta(\boldsymbol{z})) \\ &=: \mathcal{L}(\theta, \phi; \boldsymbol{x}) \end{aligned} \tag{2}$$

where $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is a variational approximation of the intractable posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ and $\mathcal{L}(\theta, \phi, \boldsymbol{x})$ is the evidence lower bound (ELBO). The variational posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ corresponds to an encoder of an autoencoder and it is expressed using a DNN. The generative model $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ corresponds to a decoder of an autoencoder and it is also expressed using a DNN.

Many studies have extended VAE for learning time series patterns by combining the elements of RNNs, e.g., STORN [22], VRNN [23], SRNN [24], Z-Forcing [25], and DMM [26]. Furthermore, recent studies [27], [28] have used generative models of sequences that use latent variables for missing data imputation tasks and time series prediction. [27] prepares two latent variables in the model and learns to disentangle the observations and latent dynamics. Unlike this model, the proposed DSSM uses the skeleton information x_{t-1} at the previous time step $t-1$ to generate x_t at time step t. Consequently, despite the fact that our model has only one latent variable at each time step, it is able to extract significant features that are robust to the subject's body size and orientation. As with our model, [28] uses information from the previous time step $t-1$ for generation, by embedding it in the latent variable z. By contrast, our model uses it directly for generation. Furthermore, our goal is to extract useful information for classification, but not to predict time series. Hence, we use all information, including future information, to infer the latent variable at each time step.

3. Proposed Method

3.1 Deep State-Space Model

In this section, we propose a novel skeleton-based action recognition method. The proposed method is composed of two steps, namely, feature extraction and classification, as shown in Fig. 1. Inspired by Krishnan *et al.* [26], we propose a new kind of deep state-space model to extract internal states as features for skeleton-based action recognition. After feature extraction, we apply a bidirectional LSTM (BiLSTM) to the extracted features for action recognition.

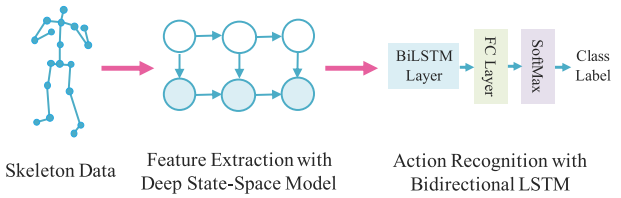
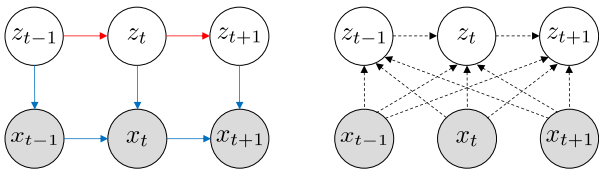


Fig. 1 **Overview of the proposed method:** the proposed method has two major parts. The first is the **deep state-space model** (DSSM), which is used to extract features from skeleton data. The second is **bidirectional long short-term memory** (BiLSTM), which takes a sequence of features extracted by the DSSM as input and classifies it into action classes.



(a) Generative model (b) Inference model

Fig. 2 **Graphical model of DSSM:** generative model (decoder) and inference model (encoder) of the DSSM.

A graphical model of the proposed DSSM is shown in Fig. 2 a. Let $\boldsymbol{x}$ denote a sequence of skeleton data $x_1, x_2, \ldots, x_T$, and let $\boldsymbol{z}$ denote a corresponding series of internal states $z_1, z_2, \ldots, z_T$. At time t, the internal state z_t transitions to a new state z_{t+1}. This transition is modeled using an internal transition model, expressed as $p_\theta(z_{t+1}|z_t)$. Moreover, the skeleton data x_t transits to a new state x_{t+1} depending on the latent variable z_{t+1} This model is called a skeleton transition model, expressed as $p_\theta(x_t|z_t, x_{t-1})$.

Compared to the original deep state-space model proposed by Krishnan *et al.* [26], our proposed DSSM has a skeleton transition model $p_\theta(x_t|z_t, x_{t-1})$, rather than an emission model $p_\theta(x_t|z_t)$. The emission model $p_\theta(x_t|z_t)$ depends exclusively on the internal state z_t and must generate the entire body x_t. By contrast, the skeleton transition model $p_\theta(x_t|z_t, x_{t-1})$ can refer to the previous skeleton state x_{t-1} for information regarding the subject's body size and orientation. Hence, the skeleton transition model $p_\theta(x_t|z_t, x_{t-1})$ only refers to the internal state z_t for the subject's motion. Consequently, we can assume that the internal state z_t represents the motion and pose independent from the body size and orientation. This kind of disentanglement of information sources has been found in various studies on generative models. According to the graphical model in Fig. 2 a, the joint distribution $p_\theta(\boldsymbol{x}, \boldsymbol{z})$ is expressed as

$$p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p_\theta(z_1)p_\theta(x_1|z_1)\prod_{t=2}^{T} p_\theta(z_t|z_{t-1})p_\theta(x_t|z_t, x_{t-1}). \tag{3}$$

As only skeleton data $\boldsymbol{x}$ is observable, we must infer the internal state $\boldsymbol{z}$ to optimize the proposed DSSM. The true posterior is

$$p_\theta(\boldsymbol{z}|\boldsymbol{x}) = p_\theta(z_1|\boldsymbol{x})\prod_{t=2}^{T} p_\theta(z_t|z_{t-1}, x_t, x_{t+1}, \ldots, x_T). \tag{4}$$

Instead of the true posterior, we can infer the internal state $\boldsymbol{z}$ using a mean-field approximation as

$$p_\theta(\boldsymbol{z}|\boldsymbol{x}) \simeq p_\theta(z_1|\boldsymbol{x})\prod_{t=2}^{T} p_\theta(z_t|z_{t-1}, \boldsymbol{x}). \tag{5}$$

In both cases, the posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ is intractable because the generative model $p_\theta(\boldsymbol{x}, \boldsymbol{z})$ is modeled using DNNs. Hence, we employ the variational approximation implemented on DNNs [14], [26], which is expressed as

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \prod_{t=1}^{T} q_\phi(z_t|z_{t-1}, \boldsymbol{x}). \tag{6}$$

Then, the model evidence $\log p_\theta(\boldsymbol{x})$ is bounded using the variational posterior q_ϕ as

$$\begin{aligned}
&\log p_\theta(\boldsymbol{x}) \\
&\geq \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_t|\boldsymbol{x})}\left[\log p_\theta(x_t|z_t, x_{t-1})\right] \\
&\quad - D_{KL}(q_\phi(z_1|\boldsymbol{x})\|p_\theta(z_1)) \\
&\quad - \sum_{t=2}^{T} \mathbb{E}_{q_\phi(z_{t-1}|\boldsymbol{x})}\left[D_{KL}(q_\phi(z_t|z_{t-1}, \boldsymbol{x})\|p_\theta(z_t|z_{t-1}))\right] \\
&=: \mathcal{L}(\theta, \phi; \boldsymbol{x}).
\end{aligned} \tag{7}$$

Here, the ELBO $\mathcal{L}(\theta, \phi; \boldsymbol{x})$ is the objective function to be maximized.

3.2 Network Architecture

We implemented the generative model p_θ described in the previous section using DNNs. A conceptual diagram of our implementation is depicted in Fig. 3.

For the internal transition model $p_\theta(z_t|z_{t-1})$, we employed the gated function based on DNNs proposed by Krishnan *et al.* [26]. This model aims at flexibility when selecting either a linear or a non-linear transition for each dimension and step, similar to a gated recurrent unit (GRU) [29]. We adopted the following parameterization, where $\odot$ denotes element-wise multiplication:

$$\begin{aligned}
g_t &= \text{sigmoid}\left(\text{M}(z_{t-1})\right), \\
h_t &= \text{M}(z_{t-1}), \\
\mu_{z_t} &= (1 - g_t) \odot \left(\text{M}\left(z_{t-1}\right)\right) + g_t \odot h_t, \\
\sigma^2_{z_t} &= \text{softplus}\left(\text{M}\left(h_t\right)\right), \\
p_\theta(z_t|z_{t-1}) &= \mathcal{N}(\mu_{z_t}, diag(\sigma^2_{z_t})),
\end{aligned} \tag{8}$$

where M is an affine transformation. We employed Dropout [30] and the rectified linear unit (ReLU) [31] activation function for each hidden layer. The output of the internal transition model $p_\theta(z_t|z_{t-1})$ represents the posterior

distribution (modeled as a Gaussian distribution with a diagonal covariance matrix) of the internal state z_t using the reparameterization trick [14]. For the reparameterization trick, the output units are divided into two groups of the same size; half of the units are followed by the identity function and used as a mean vector μ_{z_t}, and the other half are followed by the exponential function and used as a variance vector $\sigma^2_{z_t}$. The two vectors jointly represent the parameters of the Gaussian posterior $p_\theta(z_t|z_{t-1}) = \mathcal{N}(\mu_{z_{t-1}}, \mathrm{diag}(\sigma^2_{z_{t-1}}))$. This enables us to calculate the Kullback–Leibler divergence in Eq. (7). To model the internal state $p_\theta(z_1)$ at the first time step, we introduce the previous internal state z_0 filled with the default value 0, obtaining $p_\theta(z_1) := p_\theta(z_1|z_0)$.

As the skeleton transition model $p_\theta(x_t|z_t, x_{t-1})$, we also employed a gated function. This differs from the above insofar as it accepts the internal state z_t in addition to the previous skeleton state x_{t-1}. We adopted the following parameterization:

$$\begin{aligned} g_t &= \mathrm{sigmoid}\left(\mathrm{M}(z_t, x_{t-1})\right), \\ h_t &= \mathrm{M}(z_t), \\ \mu_{x_t} &= (1 - g_t) \odot \left(\mathrm{M}\left(x_{t-1}\right)\right) + g_t \odot h_t, \\ \sigma^2_{x_t} &= \mathrm{softplus}\left(\mathrm{M}\left(h_t\right)\right), \\ p_\theta(x_t|z_t, x_{t-1}) &= \mathcal{N}(\mu_{x_t}, diag(\sigma^2_{x_t})). \end{aligned} \tag{9}$$

The output also employs the reparameterization trick, enabling us to calculate the log-likelihood $p_\theta(x_t|z_t, x_{t-1})$ in Eq. (7). Again, to model the skeleton state $p_\theta(x_1|z_1)$ at the first time step, we introduce the previous skeleton data x_0 filled with the default value 0, obtaining $p_\theta(x_1|z_1) := p_\theta(x_1|x_0, z_1)$.

The inference model $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is implemented using a bidirectional GRU, which has two outputs from forward and backward paths. After an affine transformation, each output is represented as a Gaussian distribution $q^{(f)}_\theta(z_t|\boldsymbol{x})$ and $q^{(b)}_\theta(z_t|\boldsymbol{x})$ using the reparameterization trick, and the posterior is defined as their product: $q_\theta(z_t|\boldsymbol{x}) \propto q^{(f)}_\theta(z_t|\boldsymbol{x}) \times q^{(b)}_\theta(z_t|\boldsymbol{x})$. Then, the Kullback–Leibler divergence $D_{KL}\left(q_\phi(z_t|\boldsymbol{x}) \| p_\theta(z_t|z_{t-1})\right)$ in Eq. (7) is easily calculated.

To recognize actions, we apply a BiLSTM of two layers to the sequence $\boldsymbol{z}$ of internal states inferred by the DSSM. A fully connected layer and a softmax function are applied to the pair of outputs, resulting in the posterior probability $q(y|\boldsymbol{x})$ that the skeleton data $\boldsymbol{x}$ belongs to the action class y.

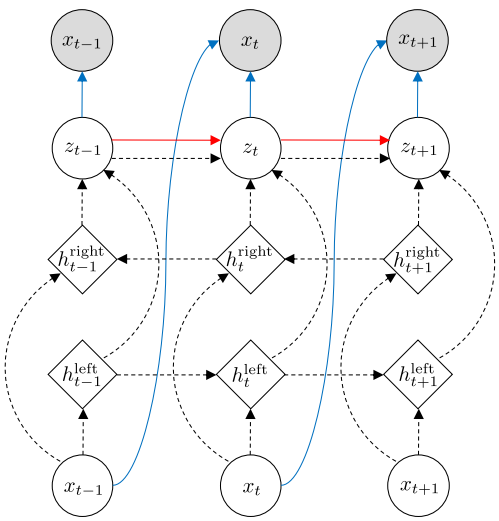


Fig. 3 **Architecture of DSSM:** the blue and red solid lines denote the generative model, and the black dashed lines denote the inference model.

4. Experiments

In this section, we evaluate our model and compare it to baseline methods and several recent works on the NTU RGB+D [9] benchmark dataset.

4.1 Datasets

NTU RGB+D is one of the largest and most common benchmark datasets among skeleton-based human action datasets. As shown in Fig. 4, each person's skeleton data consists of 3D coordinates of 25 major joints. The dataset contains 60 classes of actions collected by 40 subjects using Microsoft Kinect v2. These actions were captured from different locations and viewpoints using three cameras. In addition, the dataset has two standard evaluation protocols. Following previous studies, we evaluated the proposed method with two experimental settings. The cross-subject (CS) evaluation separates 40 subjects into training and test subsets. Subjects 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35, and 38 are used for training, and the remaining subjects are used for testing. The cross-view (CV) evaluation separates the dataset with respect to the camera; samples captured by Cameras 2 and 3 are used for training, and the remaining samples are used for testing.

4.2 Implementation Details

We trained the network using the Adam [32] algorithm with a learning rate of $\alpha = 10^{-4}$, and hyperparameters $\beta_1 = 0.9$

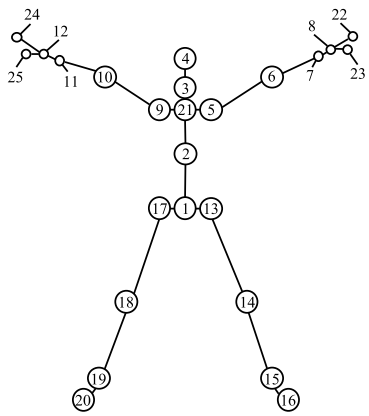


Fig. 4 Structure of skeleton data.

Table 1 Experimental results on the NTU RGB+D dataset with baselines

Method	CS Accuracy (%)	CV Accuracy (%)
BiLSTM	64.62	77.54
VAE+BiLSTM	65.19	77.72
DSSM+BiLSTM (**proposed**)	67.45	80.68

and $\beta_2 = 0.999$. The batch size was set to 256. In addition, we applied a weight decay coefficient of 10^{-4} and a dropout probability of 0.5.

The dataset contains skeleton data taken from three viewpoints. Generally, to reduce the influence of noise and the environment, normalization is performed on the position, size, rotation, etc. Many comparative models have employed the preprocessing procedure proposed by [9] to minimize differences in the imaging environment. Specifically, the coordinates of the skeleton data are transformed according to a specific joint and normalized such that the size of the body is fixed. We found that such preprocessing did not improve the performance of the proposed method, indicating its robustness to body size and orientation.

4.3 Baselines

For comparison, we introduce two baseline methods. To emphasize the characteristics of our proposed method, we refer to it as **DSSM+BiLSTM**.

BiLSTM: In this baseline, we did not employ the DSSM, but rather directly applied the BiLSTM to the skeleton data $\boldsymbol{x}$. This baseline was used to show the benefits of the DSSM when extracting features of the skeleton sequences.

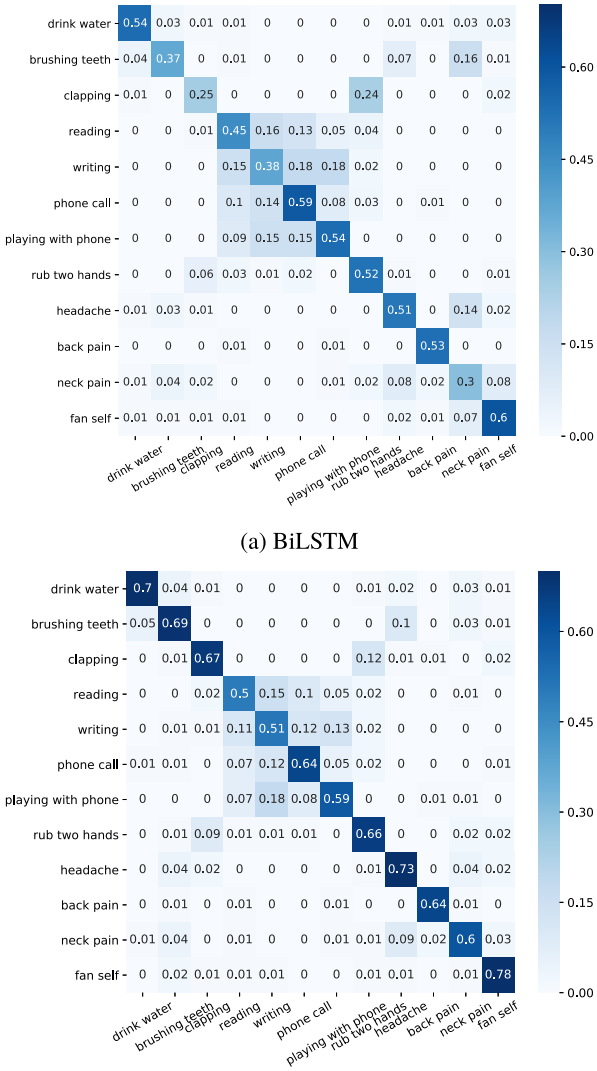
VAE+BiLSTM: In this baseline, we used a VAE as the feature extraction method, rather than the DSSM. As it does not take the temporal dynamics of the latent variables into account, we can highlight the effectiveness of the temporal modeling achieved by the DSSM.

4.4 Classification Results

The experimental results on NTU RGB+D are summarized in Table 1. DSSM+BiLSTM outperformed the baseline methods, BiLSTM and VAE+BiLSTM, by a considerable margin, which verifies the importance of the temporal feature extraction by DSSM.

To demonstrate the contribution of DSSM, we show confusion matrices of BiLSTM with and without feature extraction by DSSM in Fig. 5. Among the 60 classes in NTU-RGB+D, we focused on 11 classes, for which the BiLSTM was the least accurate. DSSM+BiLSTM improved the accuracy for most classes, especially "brushing teeth", "clapping", "headache", and "neck pain". DSSM also suppressed the confusion between similar classes, such as "reading" and "writing".

We also compared DSSM+BiLSTM to recent state-of-the-art methods, as detailed in Table 2. The Lie Group [7] and FTP Dynamic Skeletons [33] are conventional feature learning methods. The other methods are sequence classifiers based on RNNs or LSTM. DSSM+BiLSTM outperformed these feature learning methods and the typical LSTM methods examined by [9].



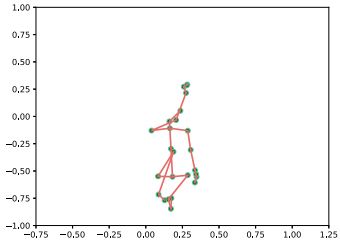
(a) BiLSTM

(b) DSSM+BiLSTM

Fig. 5 Confusion matrices of DSSM.

Table 2 Experimental results on NTU RGB+D dataset with state-of-the-art methods

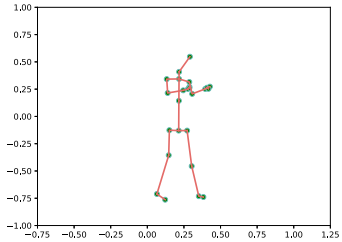
Method	CS Accuracy (%)	CV Accuracy (%)
Lie Group [7]	50.08	63.97
FTP Dynamic Skeletons [33]	60.23	65.22
HBRNN-L [8]	59.07	52.76
Deep RNN [9]	56.29	64.09
Deep LSTM [9]	60.69	67.29
P-LSTM [9]	62.93	70.27
ST-LSTM [10]	69.20	77.70
STA-LSTM [34]	73.40	81.20
VA-LSTM [35]	79.40	87.60
ST-GCN [36]	81.50	88.30
DSSM+BiLSTM (**proposed**)	67.45	80.68



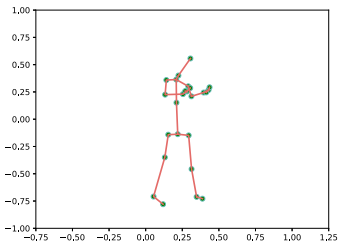
(a) Skeleton data included in the dataset



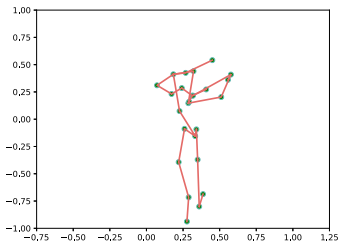
(b) Corresponding RGB image



(c) Skeleton data manually annotated by us



(d) Noisy data ($\sigma = 10^{-2}$)



(e) Noisy data ($\sigma = 10^{-1}$)

Fig. 6 **Example skeleton data:** (a) shows an example of skeleton data. (b) shows the corresponding RGB image, where the skeleton data should denote the person on the left. Obviously, the skeleton data is incorrect. (c) shows skeleton data that we annotated manually according to the RGB image. (d) and (e) are noise-added skeleton data after Gaussian noise of $\sigma = 10^{-2}$ and $\sigma = 10^{-1}$ was respectively added to (c). This example demonstrates that our experimental setting is inside the practical assumption.

However, its performance was still inferior to LSTM with specially designed architectures (e.g., [34]) and graph convolution [36]. However, each of these methods employs prior knowledge regarding the structure of human joints to build its network structure. By contrast, the proposed DSSM has a general architecture for spatio-temporal data. A DSSM adjusted for the structure of human joints will be explored in future work.

4.5 Verification of Robustness to Noise and Missing Values

In this section, we evaluate the robustness of our proposed method to noise and missing values. Indeed, NTU RGB+D contains incomplete or missing skeleton data. Figure 6 shows an example of this. Figure 6 (a) provides an example of the skeleton data included in the dataset. The skeleton data was annotated by Microsoft Kinect v2 [2]. The corresponding RGB image is the one shown in Fig. 6 (b), where the skeleton data should denote the person on the left. Obviously, the skeleton data is incorrect, and this type of incorrect annotation persists for multiple frames. We manually annotated correct skeleton data, as shown in Fig. 6 (c). We also added Gaussian noise of $\sigma = 10^{-2}$ and $\sigma = 10^{-1}$ to Fig. 6 (c) and obtained Figs. 6 (d) and (e), respectively. The differences in Fig. 6 (e) to Fig. 6 (c) are smaller than those in Fig. 6 (a). Hence, we consider that the Gaussian noise of $\sigma = 10^{-1}$ is inside the practical assumption. Further, several frames are missing; in the most severe case, 50% of the sequence is missing. The author of the dataset recommended that sequences with missing frames or incomplete skeleton data should be removed from the training and testing procedures.

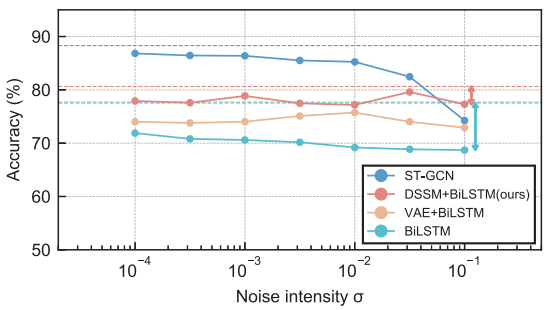


Fig. 7 Classification accuracy under noise.

In the dataset, the magnitude of noise and the amount of missing data vary across subjects and actions, and the former is difficult to measure. This indicates that it is difficult to evaluate the robustness of models in a controlled situation. Therefore, we added noise and removed frames artificially.

We compared our proposed method, DSSM+BiLSTM, to BiLSTM, VAE+BiLSTM, and a state-of-the-art method, ST-GCN. First, we added Gaussian noise $\mathcal{N}\left(0, \sigma^2\right)$ with zero mean and a different standard deviation σ to the skeleton data $\boldsymbol{x}$. The results are summarized in Fig. 7. The performance of ST-GCN and BiLSTM gradually decreases with an increase in noise. By contrast, the performance of the proposed DSSM+BiLSTM and VAE+BiLSTM is mostly constant, and DSSM+BiLSTM always outperforms VAE+BiLSTM. Specifically, ST-GCN is inferior to the proposed DSSM+BiLSTM when the noise increases beyond

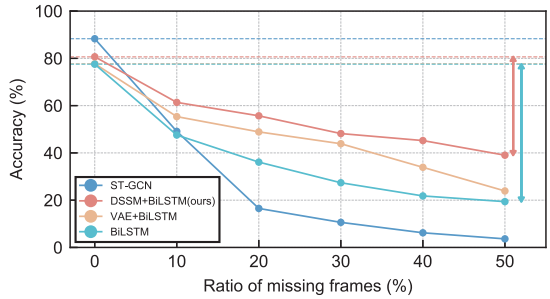


Fig. 8 Classification accuracy with missing values.

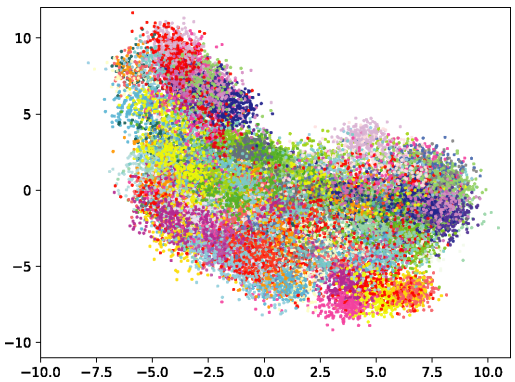


Fig. 9 Distribution of latent variables of all classes.

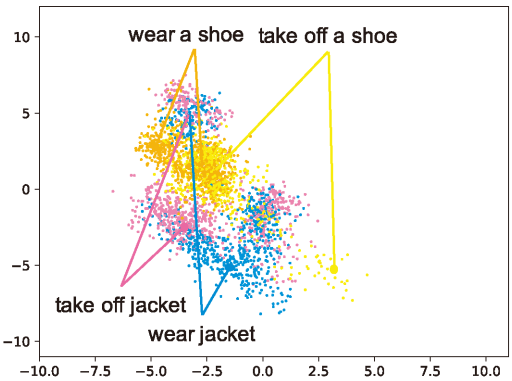
10^{-1}. Moreover, the accuracy of BiLSTM reduces by 8.84% when $\sigma = 10^{-1}$ is added, compared to when no noise is added, whereas the accuracy of DSSM + BiLSTM reduces by only 3.39%.

Next, we evaluated the robustness to missing values. We randomly erased frames of the skeleton data $\boldsymbol{x}$, and the results are summarized in Fig. 8. The performance of ST-GCN rapidly decreases with an increase in the number of missing frames: it suffices to remove 10% of the frames to render ST-GCN inferior to the proposed DSSM+BiLSTM. The performance of BiLSTM and VAE+BiLSTM decreases more slowly than that of ST-GCN, although the performance of the proposed DSSM+BiLSTM decreases the slowest among those models and still works at a certain level, even when 50% of the frames are missing. Moreover, the accuracy of BiLSTM reduces by 58.16% when 50% of the frames are missing, whereas the accuracy of DSSM+BiLSTM reduces by only 41.65%.

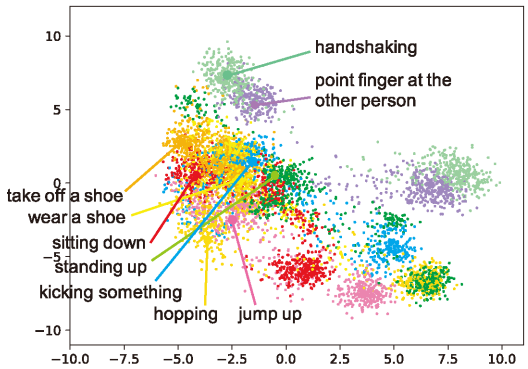
4.6 Visualization of Latent Variables

In this section, we examine in more detail how the internal state z is distributed in the latent space. To visualize the latent space, we set the dimension number of the internal state z to two and depict each internal state z with colors depending on actions, as shown in Fig. 9.

Figure 10 (a) shows pairs of reverse actions, namely "wear a shoe" and "take off a shoe", and "take off jacket" and "wear jacket". Each action has an internal state similar to its reverse action and can be discriminated using the dynamics. We found similar relationships for the pair "wear on glasses" and "take off glasses", and for the pair "put on a hat" and "take off a hat". We pick up eight actions in Fig. 10 (b). Six of them are full-body actions, and the other two are hand actions. The former are distributed over the lower part of the latent space, and the latter are distributed over the upper part. Hence, we can conclude that the DSSM embeds given sequences of skeleton data according to their semantic relations and provides useful representations.



(a) Latent variables of classes of mutually contradictory actions.



(b) Latent variables of classes mainly using feet.

Fig. 10 Distributions of latent variables.

Given a sequence of skeleton data, the latent variable also forms a sequence, and the following classifier (LSTM) classifies the sequence of the latent variables into actions. Hence, a snapshot of the latent variable does not need to contain the detailed information about actions, although its sequence should. The fact that reverse actions have similar latent variables indicates that the latent variable is devoted to capturing poses rather than movement. Nevertheless, if the latent variable were to capture more movement, the classification performance would improve; this will be considered in future work.

5. Conclusion

In this paper, we proposed the DSSM for skeleton-based ac-

tion recognition. The proposed model is a deep generative model of temporal dynamics, and it provides representations that are robust to noise and errors that occur during the process of generating skeleton data. Our experimental results demonstrate that the proposed DSSM improves the performance of subsequent classifications. Moreover, the DSSM provides robust representations and outperforms a state-of-the-art method under noise and missing values.

Acknowledgments

This study was partially supported by the MIC/SCOPE #172107101 and JSPS KAKENHI (19H04172, 19K20344).

References

[1] L.L. Presti and M.L. Cascia, "3D skeleton-based human action classi fi cation: A survey," Pattern Recognition, vol.53, pp.130–147, 2016.
[2] Z. Zhang, "Microsoft kinect sensor and its effect," IEEE Multimedia Mag., vol.19, no.2, pp.4–10, 2012.
[3] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-Time Human Pose Recognition in Parts from Single Depth Images," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
[4] L. Xia, C.-C. Chen, and J.K. Aggarwal, "View invariant human action recognition using histograms of 3d joints," CVPRW, 2012.
[5] X. Yang and Y. Tian, "Effective 3D action recognition using EigenJoints," J Vis Commun Image Represent, vol.25, no.1, pp.2–11, 2014.
[6] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," International Conference on Pattern Recognition (ICPR), pp.4513–4518, 2014.
[7] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.588–595, 2014.
[8] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1110–1118, 2015.
[9] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1010–1019, 2016.
[10] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," ECCV, vol.9907, pp.816–833, 2016.
[11] I.J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," International Conference on Learning Representations (ICLR), 2015.
[12] A.Y. Ng, M.I. Jordan, A.Y.N. Jordan, and M. I., "On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes," Advances in Neural Information Processing Systems (NIPS), pp.841–848, 2001.
[13] Y. Li, J. Bradshaw, and Y. Sharma, "Are Generative Classifiers More Robust to Adversarial Attacks?," ICMLW, 2018.
[14] D.P. Kingma and M. Welling, "Auto-encoding variational Bayes," International Conference on Learning Representations (ICLR), 2014.
[15] D.J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," Proceedings of the 31st International Conference on Machine Learning (ICML), pp.1278–1286, PMLR, 2014.
[16] D. Im, S. Ahn, R. Memisevic, and Y. Bengio, "Denoising criterion for variational auto-encoding framework," pp.2059–2065, 11 2017.
[17] J.L. Elman, "Finding Structure in Time," Cognitive Science, vol.14, no.2, pp.179–211, 1990.
[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol.9, no.8, pp.1735–1780, 1997.
[19] K. Fukushima and S. Miyake, "Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position," Biological Cybernetics, vol.15, no.6, pp.455–469, 1982.
[20] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential Recurrent Neural Networks for Action Recognition," International Conference on Computer Vision (ICCV), 2015.
[21] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks," AAAI Conference on Artificial Intelligence (AAAI), 2016.
[22] J. Bayer and C. Osendorfer, "Learning stochastic recurrent networks," arXiv preprint, 2015.
[23] C. Junyoung, K. Kyle, D. Laurent, G. Kratarth, C. Aaron, and B. Yoshua, "A Recurrent Latent Variable Model for Sequential Data," Advances in Neural Information Processing Systems (NIPS), pp.2962–2970, 2015.
[24] M. Fraccaro, S.K. Sønderby, U. Paquet, and O. Winther, "Sequential Neural Models with Stochastic Layers," Advances in Neural Information Processing Systems (NIPS), pp.2199–2207, 2016.
[25] A. Goyal, A. Sordoni, and N.R. Ke, "Z-Forcing: Training Stochastic Recurrent Networks," Advances in Neural Information Processing Systems (NIPS), pp.6713–6723, 2017.
[26] R.G. Krishnan, U. Shalit, and D. Sontag, "Structured Inference Networks for Nonlinear State Space Models," AAAI Conference on Artificial Intelligence (AAAI), pp.2101–2109, 2017.
[27] M. Fraccaro, S. Kamronn, U. Paquet, and O. Winther, "A disentangled recognition and nonlinear dynamics model for unsupervised learning," in Advances in Neural Information Processing Systems (NIPS), pp.3601–3610, 2017.
[28] S.S. Rangapuram, M.W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," in Advances in Neural Information Processing Systems 31 (NIPS), ed. S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, pp.7785–7794, 2018.
[29] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1724–1734, 2014.
[30] G. Hinton, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research (JMLR), pp.1929–1958, 2014.
[31] X. Glorot and A. Bordes, "Deep Sparse Rectifier Neural Networks," International Conference on Artificial Intelligence and Statistics (AISTATS), pp.315–323, 2011.
[32] D.P. Kingma and J.L. Ba, "Adam: A method for stochastic optimization," International Conference on Learning Representations (ICLR), 2015.
[33] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly Learning Heterogeneous Features for RGB-D Activity Recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.5344–5352, 2015.
[34] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An End-to-End Spatio-Temporal Attention Model for Human Action Recognition," AAAI Conference on Artificial Intelligence (AAAI), 2017.
[35] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition from Skeleton Data," International Conference on Computer Vision (ICCV), pp.2117–2126, 2017.
[36] S. Yan, Y. Xiong, and D. Lin, "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition," AAAI Conference on Artificial Intelligence (AAAI), 2018.



Kazuki Kawamura was an undergraduate student of the Faculty of Engineering, Kobe University, Hyogo, Japan. He received his B.E. degree from Kobe University in 2019. He is intrested in machine learning for structured data.



Takashi Matsubara received his B.E., M.E., and Ph.D. in engineering degrees from Osaka University, Osaka, Japan, in 2011, 2013, and 2015, respectively. He is currently an assistant professor at the Graduate School of System Informatics, Kobe University, Hyogo, Japan. His research interests are in model-based machine learning and computational intelligence.



Kuniaki Uehara received his B.E., M.E., and D.E. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1978, 1980 and 1984, respectively. From 1984 to 1990, he was with the Institute of Scientific and Industrial Research, Osaka University as an Assistant Professor. From 1990 to 1997, he was an Associate Professor with Department of Computer and Systems Engineering of Kobe University. From 1997 to 2002, he was a Professor with the Research Center for Urban Safety and Security of Kobe University. Currently he is a Professor with Graduate School of System Informatics of Kobe University.