Applied Intelligence
https://doi.org/10.1007/s10489-021-02994-z

Spatio-temporal stacking model for skeleton-based action recognition

Yufeng Zhong[1] · Qiuyan Yan[1,2]

Accepted: 26 October 2021

Abstract

Due to the prevalence of affordable depth sensors, skeleton-based action recognition has attracted much attention as a significant computer vision task. The state-of-the-art recognition precision usually comes from the complicated deep learning networks which need a large quantity of training data. On the counterparts, none-deep learning methods are easy to be trained and understood, however, have restricted expressive ability to extract the spatial-temporal features of skeleton data simultaneously. Therefore, it is a challenging problem to use shallow learning architecture to effectively identify complex actions in skeleton data. In this paper, we first combine **T**emporal **H**ierarchy **P**yramid (**THP**) and **S**ymmetric **P**ositive **D**efinite (**SPD**) features to simultaneously capture the temporal relationship of inter-frame and the spatial relationship of intra-frame. Then, to achieve the same learning ability as the deep learning network for a non-linear system, we propose a novel stacking ensemble-based method to effectively identify complex actions in skeleton data. We carry out extensive verification of our method on widely used 3D action recognition datasets. The experiment results indicate that we achieve state-of-the-art performance on all compared datasets.

Keywords Action recognition · Skeleton data · Stacking model · Ensemble learning

1 Introduction

Action recognition is one of the most active research topics concerned by computer vision researchers. There are many wide applications in different domains, such as video surveillance, human-computer interaction, virtual reality, etc. Among the multiple data source used to recognize human action, such as RGB images, infra-red images, and depth images, skeletons data has attracted increasing attention due to its robustness to rotation and scale changes, and many skeleton-based action recognition algorithms have been proposed in recent years.

The human skeleton is composed of a chain of joints, which express associated 3D position information of main body parts. Therefore, human action can be regarded as a successive transformation of joint sequences in 3D space. A natural way to extracting skeleton features is to estimate the spatial geometric relationship of the skeleton, hence many current skeleton-based approaches use either the joint positions or the joint angles to represent a human skeleton [1–4]. However, most geometry-based methods for skeleton-based action recognition assume all of the joints in sequence reflect dynamic variation to the same extent, which fails to focus on those joints more relevant to the action. To address this problem, more and more targeted descriptors [5–7] for action recognition have been proposed. Considering human skeleton sequences are non-Euclidean geometric data, some methods based on Riemannian manifold [8, 9] and Lie group manifold [10] are also widely used. With the thriving of deep learning, the CNN model, RNN model, and GCN model have become predominant in the field of skeleton action recognition [11–13].

The researchers have focused on the ensemble learning technologies like boosting, bagging, and stacking in recent years. In ensemble learning, multiple classifiers are used

✉ Qiuyan Yan
yanqy@cumt.edu.cn

Yufeng Zhong
yfzhong@cumt.edu.cn

[1] School of Computer Science Technology, China University of Mining and Technology, Xuzhou, 221116, Jiangsu, China

[2] Research Center of Innovation on Intelligent Prevention of Disaster and Emergency Rescure, China University of Mining and Technology, Xuzhou, 221116, Jiangsu, China

Published online: 01 February 2022

to solve the same problem, and they are combined to achieve better generalization performance. The boosting method assumes that there is a strong dependence between individual classifiers, and serially trains these classifiers [14–17]. The bagging method assumes that the individual classifiers are independent of each other, and these classifiers are trained in a parallel way [3, 7, 18]. Different from the previous two methods, stacking first trains some first-level classifiers on the initial dataset and then generates a new dataset for training a second-level classifier. The stacking method can learn more advanced and abstract feature representation from the prediction label or probability of the first classifiers, which is more conducive to the recognition of complex skeleton actions.

The state-of-the-art recognition precision usually comes from complicated deep learning networks. However, optimizing the supersized parameters needs a large quantity of training data and computing resources, which is unaffordable for most enterprises and research institutions. Compared with deep learning methods, the pipeline of manual feature extraction combining with classic classifiers is easy to be trained and understood, however, has restricted expressive ability to extract the spatial-temporal features of skeleton data simultaneously.

In this paper, we propose a **S**patial-**T**emporal **S**tacking **M**odel (**STSM**) to further enhance the complicated action recognition accuracy in 3D skeleton data, as shown in Fig. 1. Firstly, to capture the temporal relationship of adjacent skeleton sequences, we split skeleton sequences according to a Temporal Hierarchy Pyramid (THP) structure, which fused the global long-time dependency and the local short-time dependency for the arbitrary size of skeleton sequences. Secondly, since human skeleton sequences are non-Euclidean geometric data, we use Symmetric Positive Definite (SPD) matrix, a Riemannian manifold feature to further extract the spatial information of skeleton data in THP sliding windows. The SPD feature has the excellent capability of expressing the non-Euclidean character of skeleton data distribution and can be approximately transformed into the Euclidean vectors which can be applied in the typical classifier. Lastly, we adopt an ensemble learning method to improve the performance by stacking varied meta-classifiers. Through stacking two layers of meta-classifiers, the learning model stimulates the non-linear structure of neural networks. Because it's easy to understand without adjusting thousands of hyper-parameters, the **STSM** method can be applied to an environment with limited resources, which does not need a large quantity of training data and computing resources to optimize the supersized parameters. We carried out experiments on several public action recognition datasets, and the results reach 100% accuracy on UTKinect-Action3D dataset, 93.57% on MSR Action3D dataset, 99.23% accuracy on UTD-MHAD dataset, 97.28% on Gaming 3D dataset, and 89.03% on Mocap Database HDM05, which demonstrates that the model achieves the state-of-the-art recognition accuracy to this end.

In summary, the main contributions of this paper are as follows:

1. A novel structure, named Temporal Hierarchy Pyramid (THP), combines with the SPD matrixes features for skeleton-based action recognition.
2. A stacking ensemble-based method is designed to stack two layers of meta-classifiers and stimulate the non-linear structure of neural networks.
3. Ablation study is executed on the benchmark dataset, and our proposed THP and stacking modules gain up to a complete improvement compared with the baseline network architecture.

The remaining sections of the article are organized as follows. Some previous works and more details on skeleton-based action recognition are presented in Section 2. We introduce the proposed **S**patial-**T**emporal **S**tacking **M**odel (**STSM**) in Section 3. Then, we give extensive results on common action recognition datasets and explain the effectiveness of our method in Section 4. Finally, we summarize our study and give the prospect of the future research direction in Section 5.

2 Related work

In general, the existing skeleton-based action recognition approach can be mainly classified into two categories: hand-crafted feature-based action recognition and deep learning feature-based action recognition.

2.1 Hand-crafted feature-based action recognition

A simple method is to use joint distance or joint angle for skeleton-based action recognition. In [1], either the mean or variance of joint angle and the maximum angular velocity of each joint is utilized as features for action recognition. Ding et al. [2] proposed to divide the action into meaningful segments by using motion velocities, the direction of motion, and the curvatures of 3D trajectories. In [3], Halim et al. calculated the angle and distance between any two joints as feature vectors for each pose and trained a two-stage random forest for skeleton action recognition. M. lslam et al. [19] proposed an action recognition descriptor using joint's angle, sine relation, and distance to calculate the spatial and temporal information of the skeleton. In addition, as a more complex topology, the undirected complete labeled graph is used to represent skeleton features in [4]. Using the geometric relationship of

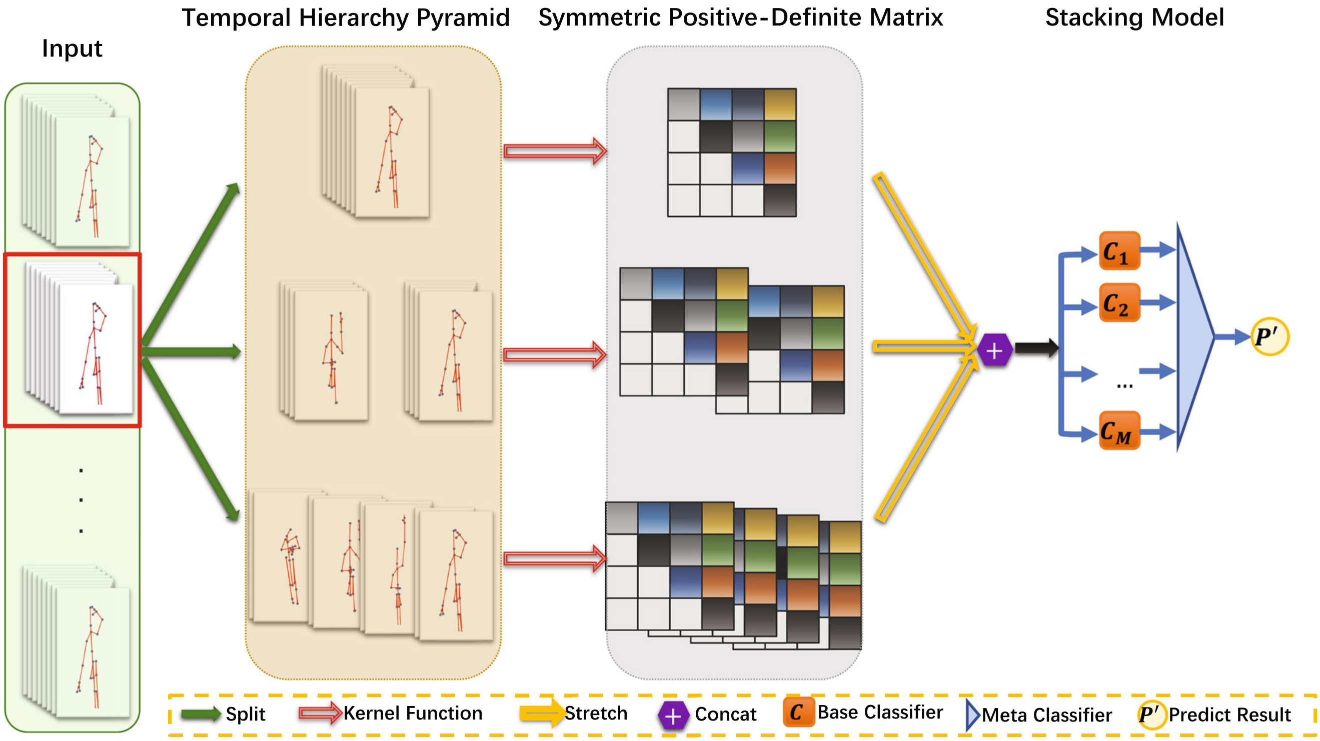


Fig. 1 The pipelines of **S**patio-**T**emporal **S**tacking **M**odel (**STSM**)

skeleton joints as features directly may ignore the inherent logic of the action. Therefore, many targeted descriptors have been proposed and applied to skeleton-based action recognition. Xia et al. [5] calculated the histograms of 3D joint locations (HOJ3D) from skeleton sequences and performed dimensionality reduction to represent the action. In [6], as additional information, depth data is called the Local Occupancy Pattern (LOP) feature, combined with a 3D joint as a skeleton feature. Besides, literature [6] also proposed the Fourier temporal pyramid as a new time pattern representation. Similar to [6], literature [7] proposed a combination of joints information and descriptors effectively based on the random forest method. Zhu et al. [7] extracted local features by the Spatio-Temporal Interest Points (STIPs) method and calculated the frame difference and pairwise distances of skeletons. In [20], O. Oreifej et al. presented a new descriptor (HON4D) to compute the distribution of the 4D normal orientation for each depth sequence by using a histogram. Furthermore, many methods projected skeleton sequences to non-Euclidean space to classify actions. Hussein et al. [8] used the covariance matrix of 3D joint position (Cov3DJ) as a descriptor of skeleton sequences and adopted the temporal hierarchy to add temporal information. In [9], a covariance matrix is extended to Symmetric Positive Definite (SPD) matrices, which have more general Riemannian geometric properties. M. Devanne et al. [21] fitted a human skeleton as a motion trajectory in the action space and expressed them as a point in the Riemannian manifold space. Vemulapalli et al. [10] proposed that human actions can be projected as curves in Lie group manifold, and used Fourier temporal pyramid with dynamic time warping to classify action by linear SVM.

2.2 Deep learning feature-based action recognition

With the thriving of deep learning, the CNN model, RNN model, and GCN model have become predominant in the field of skeleton-based action recognition. In [22], S. Nie et al. proposed a new Restricted Boltzmann Machine (RBMs) to capture both the global dynamics and the local spatial interactions in high-dimensional motion data. Zhu et al. [23] proposed a fully connected deep Long Short-Term Memory (LSTM) network to model skeletons' long-term temporal dependencies and took each joint of the skeleton as the input of LSTM. A. Shahroudy et al. [24] proposed a Part-aware LSTM (P-LSTM) model to model the long-term temporal correlation of the features for each body part. In [25], each skeleton sequence was transformed into three clips, and the CNN features of three clips in the same period were concatenated into a single feature vector, and then a Multi-Task Learning Network (MTLN) was used to process all clips in parallel. Li et al. [26] treated each joint in skeleton sequences as a channel of convolution layer and fused the inter-frame representation for skeleton by a two-stream framework. Hou et al. [11]

and P. Wang [27] proposed novel ideas to transform the skeleton sequences to color texture images and used convolutional neural networks to learn feature vectors for action recognition. Huang et al. [28] built a Riemannian network architecture to open up a new direction of applying SPD matrix to a deep neural network. Similarly, Huang et al. [29] integrated the Lie group structure into a deep neural network architecture to learn more suitable Lie group features for action recognition. It can be seen that how to encode skeleton sequences is an important problem for deep neural networks. In [30], Y. Yang et al. proposed a novel multi-task learning model to capture the intrinsic interdependencies between the latent skeletons and action classes. The action and joint configuration were regarded as a bag and its instances in [31], and a discriminative Multi-Instance Multi-Task Learning (MIMTL) framework was used to discover the intrinsic connection between joint configurations and action. In [32], R. Zhao et al. seen model parameters as random variables with designated prior distributions, with learning a set of models by fitting each model to its corresponding type of action, the predictive probability computed by those models are used to determine class label and uncertainty. R. Memmesheimer et al. [33] proposed to transform individual signals of different sensor modalities and represent them as an image. As for skeleton data, each joint and its respective axis are represented as individual signals. Since pose estimation maps provide richer cues for inferring body parts and their movements, M. Liu et al. [34] presented a novel method using pose estimation maps to recognize human action. Considering the topology of the human body inherently lies in a graph-based structure, [35] proposed a Deep Progressive Reinforcement Learning (DPRL) method to extract the most informative frame and eliminate the fuzzy frames in the sequences. In [13], the changes of human skeletons can be considered as dynamic graphs, hence Li et al. proposed a Spatio-Temporal Graph Convolution (STGC) approach. X. Gao et al. [36] proposed a graph regression-based graph convolutional network to capture the spatio-temporal variation in irregular skeleton data. Combined with the LSTM model and attention mechanism, Y. Ding et al. [37] proposed an STA-LSTM model to process long time-series information. In HAMLET [38], a novel multi-modal attention mechanism was designed to disentangle and fuse the salient unimodal spatio-temporal features. In [39], C. Ding et al. designed a novel network architecture to combine the spatial and temporal attention mechanism on Lie group manifold space.

2.3 Ensemble learning

Ensemble learning completes learning tasks by constructing and combining multiple learners, which is also called multi-classifier system, committee-based learning, and so on. In the past, researchers have focused on ensemble learning technology like boosting methods and bagging methods. In [15], a weak classifier was formed based on HMM, and then a multi-class AdaBoost algorithm was used to combine the weak classifiers with strong recognition ability. Literature [16] tested their evaluation framework by using AdaBoost as a real-time action recognition machine learning algorithm. In [18], Bloom et al. combined the discriminative power of random forest for feature selection and performed the classification by AdaBoost. In [7], a fusion scheme was proposed to combine joints information and descriptors effectively based on the random forest method. Bloom et al. [17] adopted the AdaBoost algorithm and achieved good performance in the experiment. Halim et al. [3] trained a two-stage random forest for skeleton action recognition.

These studies inspired us to apply the ensemble learning method to the field of skeletal action recognition to obtain better prediction performance. The stacking method is another ensemble learning technology. Different from the previous two methods, the stacking method first trains some first-level classifiers on the initial dataset and then generates a new dataset for training a second-level classifier. In the new dataset, the output of the first-level classifiers is treated as the input features, while the labels of the initial sample are still used as the sample labels. The choice of the second-level classifier has a great influence on the generalization performance of the stacking model. Literature [40] has shown that it is better to take the output probability of the first classifiers as the input of the second classifier, and adopted Multi-response Linear Regression (MLR) as the second level classifier. Furthermore, using different attribute sets is better in MLR. To some extent, using the stacking model, a kind of ensemble learning method, can be regarded as constructing a simple neural network.

3 Method

In this section, we introduce some details of our **S**patial-**T**emporal **S**tacking **M**odel (**STSM**). First, we illustrate the general process of skeleton sequence segmentation based on THP and theoretically analyze the general structure of THP. Then, we analyze and compare the non-linear spatial feature extraction ability of SPD matrices generated by popular kernel functions. Finally, we introduce the application of the stacking model on SPD features in our method.

3.1 Temporal feature extraction by THP

The proposed THP structure is shown in Fig. 2. We adopt the temporal hierarchical construction introduced in [8] as

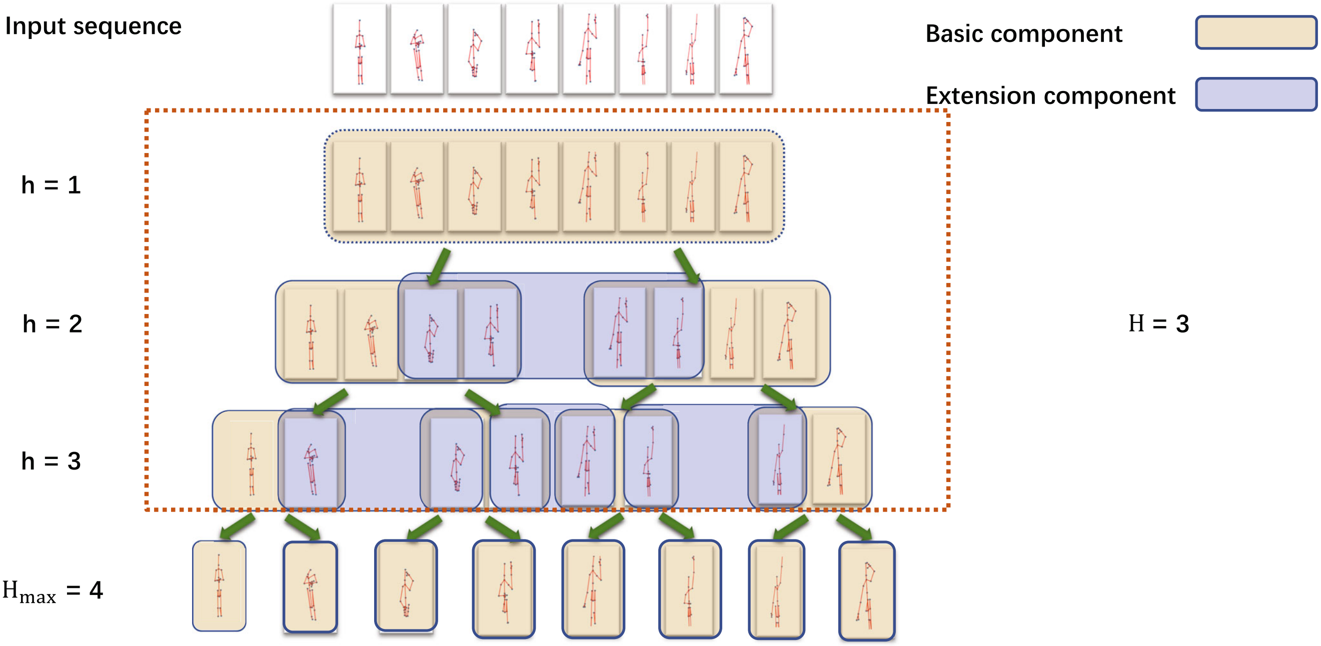


Fig. 2 The general form of temporal hierarchy pyramid

our basic model. Furthermore, we give a detailed description of the general form of THP, which can be regarded as a full binary tree from the topological structure.

The THP is composed of two kinds of components, one is the basic component, and the other is the extension component. The basic component directly splits the input sequence into multiple sub-sequences, while the extension component covers two adjacent basic components. We choose half the length of a basic component as an overlapping step, which has achieved good results in practical applications [8]. Significantly, the extension component is optional, and it will extract tighter temporal features.

Next, we theoretically analyze the relationship between the height of the THP and the length of the dataset. Given n_{min} as the minimum length of the sequence in this dataset, the maximum hierarchy H_{max} depends on the n_{min}, which is computed as (1). Specifically, the preset level H in THP can be selected from 1 to H_{max}.

$$H_{max} = \lfloor \log_2 n_{min} \rfloor + 1 \tag{1}$$

One advantage of the extension component is that it can more precisely describe the temporal character in a whole action sequence. The other superiority is to increase the number of sub-sequences and to enrich the training samples. The number of subsequences generated by THP can be expressed by (2).

$$s = \begin{cases} 2^H - 1 & \text{basic component} \\ 2^{H+1} - H - 2 & \text{basic \& extension component} \end{cases} \tag{2}$$

Where, s is the number of all sub-sequences. The component at the top of THP can capture the global long-time dependency, while the component at the bottom of THP can capture the local short-time dependency. For the h_{th} layer in THP, $h \in \{1, 2, \ldots, H\}$, the length of sub-sequence is shown in (3).

$$l^h = \frac{k}{2^{h-1}} \tag{3}$$

Where, k is the length of the raw sequence and l^h is the length of sub-sequence.

Specifically, as shown in Fig. 2, The maximum hierarchy H_{max} is 4, and we choose $H = 3$ as the preset level. On the 1_{st} layer, there is only one basic component, covering the entire sequence, and on the 2_{nd} layer, there are two basic components covering 1/2 frames of the sequence without overlapping each other. If we choose the extension component, it will be added to two adjacent basic components and occupies half of each basic component.

3.2 Spatial feature extraction by SPD

The second highlight of our approach is to compare the nonlinear spatial feature extraction ability of SPD matrices generated by different kernel functions. For the convenience of narration, we need to give some definitions first.

Defined the sub-sequence obtained in Section 3.1 as $S_{l^h \times d}$, $S_{l^h \times d}$ can be regarded as a data matrix, where l^h is the length of sub-sequence and $d = 3 * N$. The 3 represents three coordinate axes, and the N represents the number of joints in the skeleton. Each column in d is a feature vector f_i

($i = 1, 2, \dots, d$) and f_i represents the same joint coordinate in different frames. We use $M_{d\times d}$ as a general description of the SPD matrix, which is the result of traversing matrix $\mathrm{S}_{l^h\times d}$ and calculating any two feature vectors with a kernel function.

As mentioned in [9], for covariance matrix, its kernel function can be written as (4). Where (i, j) means an entry of M_{cov}, and (f_i, f_j) is the feature vector in $\mathrm{S}_{l^h\times d}$ corresponding to (i, j). The k_{cov} represents the kernel function used to form the SPD matrix M_{cov}.

$$M_{\mathrm{cov}}(i, j) = k_{\mathrm{cov}}\left(f_i, f_j\right) = \left(\frac{f_i - \bar{f_i}}{\sqrt{l^h - 1}}\right)^{\mathrm{T}} \left(\frac{f_j - \bar{f_j}}{\sqrt{l^h - 1}}\right) \tag{4}$$

However, the covariance matrix is easy to be singular and only calculate linear correlation, which greatly limits its feature extraction ability. Similarly, for modeling linear relationships, the linear kernel function can generate a nonsingular matrix, which can be used as a better descriptor than the covariance matrix. For linear kernel function, it can be expressed as (5).

$$M_{\mathrm{linear}}(i, j) = k_{\mathrm{linear}}\left(f_i, f_j\right) = f_i^{\mathrm{T}} f_j \tag{5}$$

The polynomial kernel, written as (6), is often used in machine learning algorithms to express the similarity between two vectors. Where d is the kernel degree, and γ is known as slope, and c_0 means the intercept. Besides, the polynomial kernel function can project samples to higher dimensional nonlinear space. From a mathematical point of view, the polynomial kernel can not only consider the similarity of vectors in the same dimension but also consider the similarity of vectors across dimensions.

$$M_{\mathrm{poly}}(i, j) = k_{\mathrm{poly}}\left(f_i, f_j\right) = \left(\gamma f_i^{\mathrm{T}} f_j + c_0\right)^d \tag{6}$$

The hyperbolic tangent function is often used as an activation function in the field of neural networks, also known as the sigmoid function, and it is a nonlinear kernel function with good performance. It is defined as (7).

$$M_{\mathrm{sigmoid}}(i, j) = k_{\mathrm{sigmoid}}\left(f_i, f_j\right) = \tanh\left(\gamma f_i^{\mathrm{T}} f_j + c_0\right) \tag{7}$$

The Radial Basis Function (RBF) kernel, also known as Gaussian kernel, is used in various kernel learning algorithms, like Support Vector Machine (SVM). The kernel is defined as (8). Where σ^2 is known as the variance in the Gaussian kernel.

$$M_{rbf}(i, j) = k_{rbf}\left(f_i, f_j\right) = \exp\left(-\frac{\left\| f_i^{\mathrm{T}} - f_j \right\|^2}{\sigma^2}\right) \tag{8}$$

The Laplacian kernel is a variant of Radial Basis Function (RBF) kernel with lower parameter sensitivity, which is defined as (9). Where $\left\| f_i^{\mathrm{T}} - f_j \right\|^1$ is the Manhattan distance between input vectors f_i, f_j.

$$M_{\mathrm{laplace}}(i, j) = k_{\mathrm{laplace}}\left(f_i, f_j\right) = \exp\left(-\frac{\left\| f_i^{\mathrm{T}} - f_j \right\|^1}{\sigma^2}\right) \tag{9}$$

In addition to the above kernel functions, there are a large number of kernel functions, including Chi-squared kernel, Spline kernel, Wavelet kernel, and so on. These kernel functions have a common good property, that is, non-singularity, which ensures the applicability of the Riemannian metric.

3.3 Action recognition by STSM

Different from commonly action recognition methods, which usually select a single classifier to predict the action

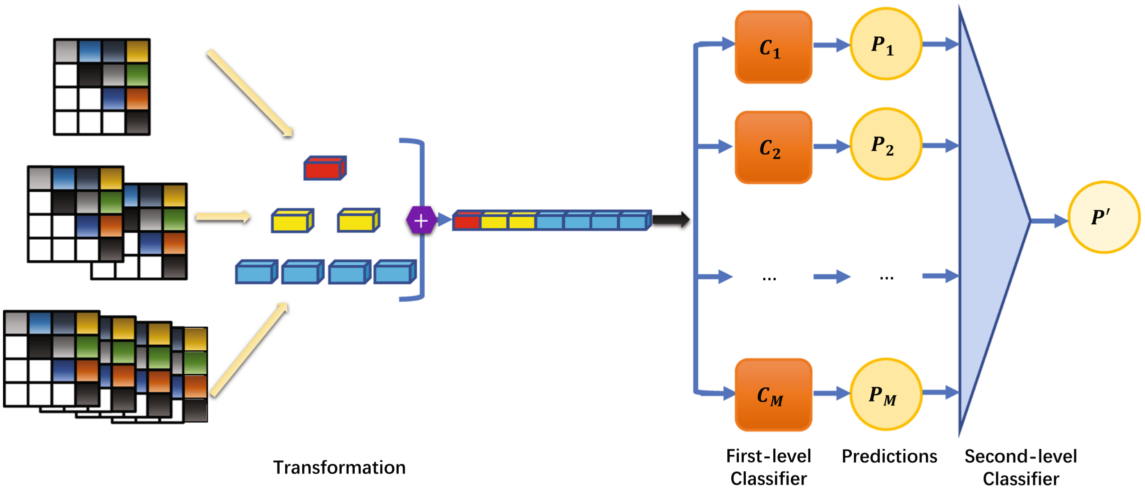


Fig. 3 The transformation on SPD matrix and the application of **STSM**

label, we adopt an ensemble learning model called stacking, which learns a classifier based on the output of multiple basic classifiers. To apply the stacking model to the SPD matrix, we need to do some transformation on the SPD matrix obtained in Section 3.2. Our algorithm is shown in Fig. 3.

Specifically, half of the elements in the SPD matrix are repetitive, so they need to be removed. Moreover, different SPD matrices in the same layer of THP should be fused to a comprehensive representation. The stacking model learns a second-level classifier on top of basic classifiers, which are called the first-level classifiers. The general procedure is illustrated in Algorithm 1 and some implementation details of our scheme are shown in the following steps.

Algorithm 1 Stacking on SPD matrices.

```
Input:
A set of SPD matrices, the number of which is $s$;
first-level learning algorithm $L_1$, $L_2$,...,$L_m$;
second-level learning algorithm $L'$.
Output:
The second-level classifier $C'$s predicted probability $P'$.
for i ← 1 to s do
    Straighten the upper triangular matrix of $M_{d\times d}$ to vector.
for h ← 1 to H do
    Concatenate the feature vectors as vector $x$ according to their layer.
Construct the first level training dataset $D$,
where $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$.
for m ← 1 to M do
    Learn a base classifier $C_m = L_m(D)$.
for m ← 1 to M do
    Construct the second level training dataset $D'$ that contains $(x_i', y_i)$.
    The output of first-level classifiers are $P_{1i} = C_1(x_i)$, $P_{2i} = C_2(x_i), \ldots, P_{mi} = C_m(x_i)$.
    Where ${x_i}' = \{P_{1i}, P_{2i}, \ldots, P_{mi}\}$.
Learn a new classifier $C' = L'(D')$.
The second-level classifier $C'$ predicted probability $P'$=$C'$ $(P_1, P_2 \ \ldots, P_m)$.
return $P'$.
```

Step 1. To transform the SPD matrix to a feature vector. Considering the symmetry, we only use the upper triangular matrix of $M_{d\times d}$ and straighten it to a one-dimensional vector at each level. In this way, we obtain a set of feature vectors, and each feature vector represents the spatial information of action over a period of time. To integrate the global and local time dependency, we concatenate the set of feature vectors to a complete feature vector in order of hierarchy.

Step 2. To fit first-level classifiers on the train set. There are several choices for learning basic classifiers: 1) By adjusting parameters in a learning algorithm to generate various base classifiers, like select different kernel functions or penalty coefficient for SVM. 2) We can simultaneously adopt different classification methods (such as SVM, HMM, neural network, decision tree, and so on) as basic classifiers. 3) We can even plug else ensemble learning models, such as Bagging and Boosting, to generate basic classifiers.

Step 3. To construct a new dataset using the output classification probability by the first-level classifiers. Here, the predicted probability is considered as new attribute of the dataset. Specifically, let each samples in the first-level dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, and the first-level classifiers are C_1, C_2, ..., C_m, then the output of first-level classifiers are $P_{1i} = C_1(x_i)$, $P_{2i} = C_2(x_i)$, ..., $P_{mi} = C_m(x_i)$. The constructed dataset is (x_i', y_i), where $x_i' = \{P_{1i}, P_{2i}, \ldots, P_{mi}\}$.

Step 4. To learn a second-level classifier based on new dataset. Most machine learning algorithms, including ensemble learning algorithms, can be used as second-level learning algorithm. For a second-level classifier C', its relationship with the first-level classifiers' predicted probability $P_{1i}, P_{2i}, \ldots, P_{mi}$ is as (10).

$$P_i' = C' \left(P_{1i}, P_{2i}, \ldots, P_{mi}\right) \tag{10}$$

4 Experiment

In this section, we compare our proposed **STSM** performance results with the state-of-the-art methods on five widely used datasets. We test our method on UTKinect-Action3D dataset, MSR Action3D dataset, UTD-MHAD, Gaming 3D Dataset, and Mocap Database HDM05, comparing our method with various methods, including hand-crafted feature-based methods and deep learning feature-based methods. It can be seen that we have reached state-of-the-art performance on all compared datasets, which demonstrates the effectiveness of our method.

4.1 Datasets

For the datasets we used in the experiment, the main statistics of five datasets in brief and the corresponding relationship of the maximum hierarchy H_{max} are shown in Table 1. We reveal the relationship between the maximum hierarchy H_{max} and the minimum length of sequence n_{min} in a certain dataset, as shown in (1). That means that for different datasets, the maximum hierarchy H_{max} that can be selected is different, instead of simply increasing the number of hierarchies.

Table 1 The main statistics of five datasets

Dataset	n_{min}	n_{avg}	$\lfloor \log_2 n_{min} \rfloor$	H_{max}
UTK	5	30	2	3
MSR	13	41	3	4
UTD	41	67	5	6
G3D	31	32	4	5
HDM	14	262	3	4

UTKinect-Action3D dataset The UTKinect-Action3D [5] dataset consists of 10 different actions: walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands, clap hands, which was captured by a single stationary Kinect. The dataset was performed by 10 subjects, and each subject was repeated twice. Each body skeleton consists of 20 joints. It is abbreviated as **UTK**.

MSR Action3D dataset The MSR Action3D [41] dataset is recorded with a depth sensor similar to the Kinect. There are 20 actions and 10 subjects, each subject performs each action 2 or 3 times. Similarly, each body skeleton consists of 20 joints. It is abbreviated as **MSR**.

UTD-MHAD UTD-MHAD [42] is collected by a Microsoft Kinect sensor and a wearable inertial sensor. This dataset contains 27 actions performed by 8 subjects (4 females and 4 males), each subject repeated each action 4 times. The body skeleton consists of 20 joints too. It is abbreviated as **UTD**.

Gaming 3D dataset Gaming 3D [16] dataset containing 663 action sequences, ranging across 20 gaming motion classes, is captured by Microsoft Kinect. The skeleton sequences in Gaming 3D dataset are represented with 20 joints. All actions in Gaming 3D dataset are performed by 10 actors. It is abbreviated as **G3D**.

Mocap database HDM05 Mocap Database HDM05 [43] containing 2337 action sequences, ranging across 130 motion classes, is captured by an optical marker-based Vicon system. Different from the dataset captured by Microsoft Kinect, the skeleton sequences in HDM05 dataset are represented with 31 joints. All actions in HDM05 dataset are performed by 5 actors. As we know, HDM05 dataset has the largest number of motion classes. It is abbreviated as **HDM**.

4.2 Implementation details

We set different H of THP on five datasets. The THP is composed of the basic component and the extension component. We consider that THP with the extension components can extract more compact and extended temporal features. So we add the extension components on all five datasets. As mentioned in Section 3.1, for different datasets, the maximum hierarchy H_{max} is different, so the optimal H of each dataset is also different. Another point worth noting is that the larger the selected H is, the longer the temporal features length is generated, and the more difficult the classifier training is.

We compare six kernel function matrixes mentioned in Section 3.2. Specifically, we compare the covariance matrix and the linear kernel matrix for generating linear features, the poly kernel matrix and the sigmoid kernel matrix for generating nonlinear features, and two variants of Gaussian kernel, the RBF kernel matrix and the Laplace kernel matrix for generating gaussian features. The results verify the advantages of gaussian features and demonstrate that the Laplace kernel function matrix has the best effect.

The second layer of the stacking model is a logistic regression classifier. We have done a lot of experiments on how to choose the first and second level classifiers in the stacking model. We fully consider different classifiers, like Bayes, decision tree, KNN, perceptron, logistic regression, and SVM. We also try different parameter combinations of the same classifier, such as choosing different kernel functions or penalty factors for SVM. Finally, we retain the classifiers with good performance as the first-level classifiers and select the logistic regression classifier as the second-level classifier.

We use the Python programming language as an actual coding language and implement the model with an open-source machine learning framework Scikit-Learn [44]. All experiments are carried out on a Linux server equipped with an Intel Xeon Gold 6266C CPU without any GPUs. We adopt a 10-fold cross-validation strategy to assess the performance of different methods in this work. In each fold, half samples are used to train and the other half is used for testing. The reported results are all averaged over ten different combinations of training and testing subsets.

4.3 Ablation study

To verify the effectiveness of our **S**patial-**T**emporal **S**tacking **M**odel (**STSM**), we incrementally evaluate each module on five widely used datasets.

To evaluate the effectiveness of our proposed THP and the benefits of the extension component, we have carried out some experiments. In order to eliminate the interference of choosing different SPD matrices, we use the SPD matrices generated by the Laplace kernel function in the experiments. The logistic regression classifier is used for action recognition. The experimental results show that for the small datasets with the small number of frames, **UTK**, **MSR**, and **UTD**, it is better to choose a smaller hierarchy,

Table 2 The results of THP with basic component and extension component

Dataset	1	2	3	4	5	6
UTK	71.67	**100.00**	96.67	–	–	–
MSR	91.23	**92.40**	86.55	84.21	–	–
UTD	95.37	**98.84**	98.07	97.30	96.14	96.14
G3D	93.20	96.37	96.37	96.60	**97.05**	–
HDM	71.37	82.91	85.47	**88.46**	–	–

Bold numbers represent the maximum accuracy of their row

and for large datasets with the large number of frames, **G3D** and **HDM**, it is better to choose a larger hierarchy. The experiment results are shown in Table 2.

To compare the performance of several commonly used kernel functions in detail, including covariance matrix, linear kernel function matrix, polynomial kernel function matrix, sigmoid kernel function matrix, the RBF kernel function matrix, and the Laplace kernel function matrix, we have carried out some experiments. In order to ensure the fairness of comparison, we keep the best THP settings as shown in Table 2 on each dataset, and all parameters in experiments, including the parameters in each kernel function and the parameters of logistic regression classifier, are only tuned through cross-validation on the training set. The experiment results are shown in Table 3. As pointed out in [9], the covariance matrix has the disadvantages of singularity, which limited its modeling ability for complex feature relations, so it obtains the lowest accuracy and can't train on **G3D** dataset. As a variant of the RBF kernel function, the Laplace kernel is different from the RBF kernel function in that it is less sensitive about parameters. It can be concluded that the nonlinear feature extraction ability, noise immunity, and wider range are the reasons for the success of the Laplace kernel function.

Finally, to verify the improvement effect of our proposed stacking model on the baseline, on the basis of the previous two experiments, we conduct a comparative experiment. The experiment configurations represented as follows. Where, **STSM**(baseline) shows the performance of the best SPD matrix, the Laplace kernel function matrix. **STSM**(+THP) shows the performance of the best hierarchy in THP based on the Laplace kernel function matrix. **STSM**(+THP+stacking) selects a variety of classifiers mentioned in Section 3.3 as the first and second level classifiers and retains the best combination of classifiers. We keep the same experimental settings in Section 4.4.

Table 3 Comparison of different kernel function matrices

Dataset	Cov	Linear	Poly	Sigmoid	RBF	Laplace
UTK	78.33	91.67	93.33	91.67	93.33	**100.00**
MSR	83.04	83.04	84.80	83.63	89.47	**92.40**
UTD	91.12	94.98	95.75	94.98	96.91	**98.84**
G3D	–	95.01	95.24	94.78	96.15	**97.05**
HDM	85.33	85.33	88.32	85.90	85.47	**88.46**

Bold numbers represent the maximum accuracy of their row

- **STSM**(baseline) is our basic scheme based on Symmetric Positive Definite (SPD) matrices without any extra module.
- **STSM**(+THP) is the scheme with our proposed Temporal Hierarchy Pyramid (THP) applied to baseline.
- **STSM**(+THP+stacking) is the scheme with the Temporal Hierarchy Pyramid (THP) and our proposed stacking model is applied to baseline.

Figure 4 shows the ablation study results, from which several observations could be drawn. On the **MSR**, **UTD**, and **G3D**, the **STSM**(baseline) only using SPD matrices without THP structure achieves good performance. This outcome indicates that the Riemannian manifold feature can further express the spatial information of skeleton data. By combining SPD matrices and THP structure, **STSM**(+THP) significantly outperforms the baseline on both **UTK** and **HDM**. This validates the effectiveness of our proposed THP structure and demonstrates their complementary power over utilizing either alone and their combination leads to further improvements. By stacking varied meta classifiers, **STSM**(+THP+stacking) displays further boosts. For example, on **MSR**, **STSM**(+THP+stacking) outperforms the **STSM**(baseline) counterpart by a margin of 2.34%, and it outperforms **STSM**(+THP) counterpart by 1.17%.

4.4 Comparison of state-of-the-arts

The following describes the details of the experiment results compared with state-of-the-art approaches. To make the comparison more intuitive, we summarize the explicit information about the comparison methods in Table 4, such as the mainly used feature representation technology, the venue, and the year of publication.

Table 5 presents the comparison results with the state-of-the-arts on the **UTK**. The experiment results illustrate that our **STSM** model outperforms the existing approaches. Among these methods, STA-LSTM [47] achieved 79.26% as minimum accuracy, while our **STSM** model can more efficiently capture spatial and temporal information. Compared with the Lie group-based methods Lie Group [10] and STA-DeepLG [39], the **STSM** model can better characterize the topological structure of joints information depend on SPD matrices. Besides, GR-GCN [36] and DPRL [35] obtain good performance in a graph-based structure, while our **STSM** model can not only extract the spatial information of skeleton data but also fuse the global long-time dependency and the local short-time dependency depended on the THP structure. Moreover, our

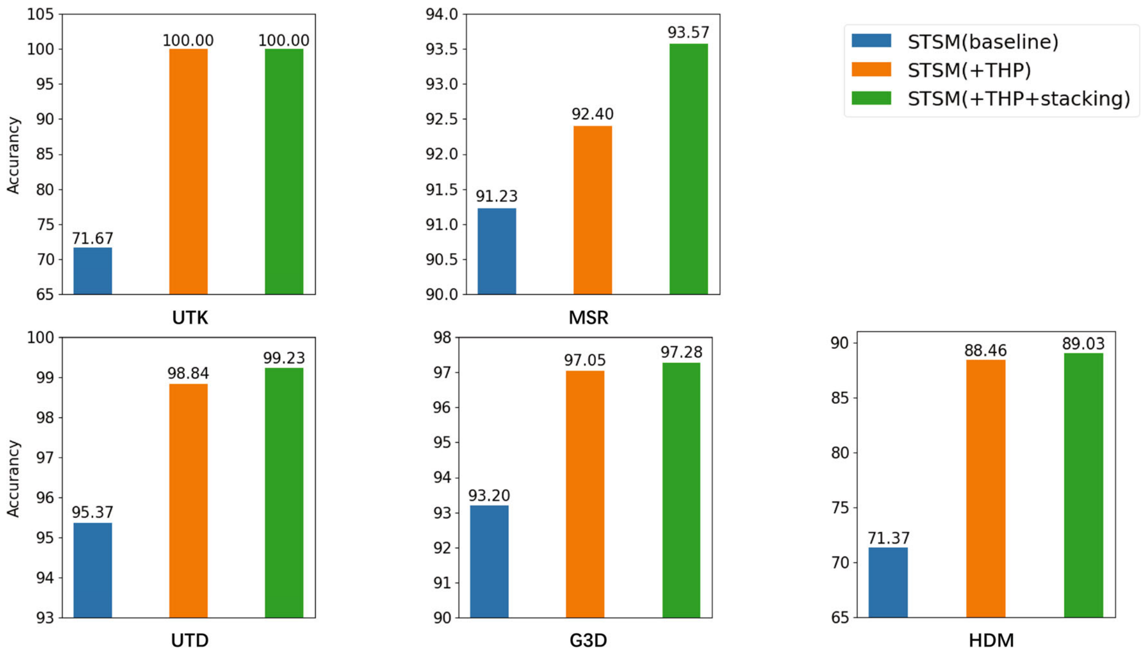


Fig. 4 Ablation study of **S**patial-**T**emporal **S**tacking **M**odel (**STSM**)

Table 4 Summary of the state-of-the-arts methods used in comparison

Methods	Technology	Venue	Year
Actionlets [6]	descriptor	CVPR	2012
HON4D [20]	descriptor	CVPR	2013
Lie Group [10]	Lie group manifold	CVPR	2014
MT [21]	Riemannian manifold	IEEE	2015
Ker-RP-RBF [9]	Riemannian manifold	ICCV	2015
LRBM [22]	Boltzmann	Elsevier	2015
P-LSTM [24]	LSTM	CVPR	2016
MIMTL[31]	multi-instance+multi-task	IEEE	2016
LM^3TL [45]	multi-task	IEEE	2017
SPDNet [28]	Riemannian manifold+neural network	AAAI	2017
Lie Group Net [29]	Lie group manifold+neural network	CVPR	2017
CNN [27]	convolutional neural network	Elsevier	2018
SOS [11]	convolutional neural network	Elsevier	2018
PEM [34]	pose estimation maps+neural network	CVPR	2018
DPRL [35]	graph+reinforcement learning	CVPR	2018
Deep STGCK [13]	graph convolution network	AAAI	2018
GR-GCN [36]	graph convolution network+graph regression	ACM	2019
RA-GCN [46]	multi-stream+graph convolutional network	ICIP	2019
HDM-BG [32]	Bayesian	CVPR	2019
STA-LSTM [47]	LSTM+attention	IEEE	2019
MSM [33]	multi-modal	IROS	2020
HAMLET [38]	multi-modal+attention	IROS	2020
ASD-R [19]	joint's angle+sine relation+distance	Springer	2021
STA-DeepLG [39]	Lie group manifold+LSTM+attention	Springer	2021

Table 5 Comparison with the state-of-the-arts methods on the **UTK**

Methods	Accuracy(%)
STA-LSTM [47]	79.26
HDM-BG [32]	87.50
RA-GCN [46]	89.23
Lie Group [10]	97.20
HAMLET [38]	97.45
STA-DeepLG [39]	97.70
GR-GCN [36]	98.50
DPRL [35]	98.50
MIMTL [31]	99.19
STSM(baseline)	71.67
STSM(+THP)	**100.00**
STSM(+THP+stacking)	**100.00**

Bold numbers represent the maximum accuracy of their columns

Table 7 Comparison with the state-of-the-arts methods on the **UTD**

Methods	Accuracy(%)
RA-GCN [46]	79.26
STA-LSTM [47]	79.56
Lie Group [10]	92.36
HDM-BG [32]	92.80
MSM [33]	93.30
PEM [34]	94.51
HAMLET [38]	95.12
ASD-R [19]	96.00
STSM(baseline)	95.37
STSM(+THP)	98.84
STSM(+THP+stacking)	**99.23**

Bold numbers represent the maximum accuracy of their columns

method of combining multiple classifiers is superior to the multimodal-based method HAMLET [38] and Multi-instance Multitask-based method MIMTL [31].

Table 6 illustrates our proposed **STSM** model's comparison with the previous approaches against the accuracy index of **MSR**. In these methods, Bayesian-based HDM-BG [32] achieved 86.10% as minimum accuracy, whereas Riemannian manifold-based MT [21] obtained a maximum of 92.10% accuracy among compared approaches. The conventional approaches Actionlets [6] and HON4D [20] achieve good performance by using a descriptor to extract spatial motion information. By comparison, our proposed **STSM** model attained the best performance of 93.57% with stacking ensemble-based method to effectively identify complex actions in skeleton data, which improved 1.47% of the previous maximum accuracy of MT [21] model using Riemannian manifold.

In Table 7, we compare the results of the proposed **STSM** model with other state-of-the-art approaches on the **UTD**. According to given values, the Lie Group [10] model achieved 92.36% at least accuracy with the combination of projecting skeleton as curves in Lie group manifold and using Fourier temporal pyramid with dynamic time warping. The HAMLET [38] model achieves better performance by combining multi-modal information and attention mechanism to fuse the salient unimodal spatio-temporal features than the MSM [33] only transforms individual signals of different sensor modalities and represents them as an image. Our proposed **STSM** model attained 99.23% and improved 3.23% of the state-of-the-art ASD-R [19] model, which uses joint's angle, sine relation, and distance to calculate the spatial and temporal information of skeleton.

As shown in Table 8, we compare the effectiveness of our **STSM** model with other state-of-the-art methods on the **G3D**. The STA-DeepLG [39] model combines the spatial and temporal attention mechanism on Lie group manifold space, which boosts the performance from 80.60% to 90.30% on the raw Lie group manifold-based method Lie Group [10] representation. Compared with the deep learning methods SOS [11] and CNN [27], which transform

Table 6 Comparison with the state-of-the-arts methods on the **MSR**

Methods	Accuracy(%)
RA-GCN [46]	56.74
STA-LSTM [47]	81.16
HDM-BG [32]	86.10
Actionlets [6]	88.20
HON4D [20]	88.89
Lie Group [10]	90.37
LM^3TL [45]	90.53
MT [21]	92.10
STSM(baseline)	91.23
STSM(+THP)	92.40
STSM(+THP+stacking)	**93.57**

Bold numbers represent the maximum accuracy of their columns

Table 8 Comparison with the state-of-the-arts methods on the **G3D**

Methods	Accuracy(%)
STA-LSTM [47]	75.60
RA-GCN [46]	78.60
Lie Group [10]	80.60
STA-DeepLG [39]	90.30
LRBM [22]	90.50
HDM-BG [32]	92.00
SOS [11]	95.45
CNN [27]	96.02
STSM(baseline)	93.20
STSM(+THP)	97.05
STSM(+THP+stacking)	**97.28**

Bold numbers represent the maximum accuracy of their columns

Table 9 Comparison with the state-of-the-arts methods on the **HDM**

Methods	Accuracy(%)
SPDNet [28]	62.57
Ker-RP-RBF [9]	66.20
P-LSTM [24]	75.47
Lie Group Net [29]	78.04
STA-DeepLG [39]	84.59
Deep STGCK [13]	87.42
STSM(baseline)	71.37
STSM(+THP)	88.46
STSM(+THP+stacking)	**89.03**

Bold numbers represent the maximum accuracy of their columns

the skeleton sequences to color texture images and use convolutional neural networks to learn feature vectors for action recognition, the **STSM** model can make full use of the temporal relationship of inter-frame and the spatial relationship of intra-frame by SPD matrices and THP structure, thus obtaining a higher recognition accuracy.

In Table 9, we compare the recognition results of our proposed method with different methods on the **HDM**. Among these methods, the Riemannian manifold-based approaches SPDNet [28] and Ker-RP-RBF [9] achieve good performance in using the SPD matrices of 3D joint position as a descriptor of skeleton sequences. The P-LSTM [24] model which groups the human skeleton into five parts and models the long-term temporal correlation of the features for each body part achieves better performance. Our proposed **STSM** model outperforms the state-of-the-art Deep STGCK [13] model, which considers human skeletons as dynamic graphs and proposes a spatio-temporal graph convolution (STGC) approach.

4.5 Statistical analysis

In order to compare the results of different methods, we use the statistical analysis method to analyze several representative methods. The selected comparison methods include Bayesian-based HDM-BG [32], Lie Group [10], attention-based STA-LSTM [47] and graph convolutional network-based RA-GCN [46]. We adopt the Friedman test [48] to reject the null hypothesis, i.e., the measured rank may not differ from the mean rank, which states that all considered methods have the equivalent performance. Once we check for the statistically significant differences in the classification performance, we perform the Nemenyi test [49] as a post-hoc test to compare the methods with each other.

First, we implement the Friedman test. Friedman test ranks comparison methods on all datasets. The best method gets rank 1, the second-best method gets rank 2, and so on. The average rank of a method is obtained by averaging the ranks on all datasets. Ranks of the competing methods obtained by the Friedman test procedure are showing in Table 10.

The Friedman statistic value follows a χ_F^2 distribution with degree of freedom equals to $(k-1)$, as (11). Where N is the number of datasets, k is the number of methods and R_i is the average rank of methods on all datasets. In this paper, N = number of datasets considered = 4 and k = number of comparison methods = 5.

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_{i=1}^{k} R_i^2 - \frac{k(k+1)^2}{4}\right] \tag{11}$$

In our case, $\chi_F^2 = 13.00$. Iman and Davenport [50] found that this statistic value is undesirable conservative, and proposed a corrected one, which is distributed following an F distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom, as (12).

$$F_F = \frac{(N-1)\chi_f^2}{N(k-1)-\chi_f^2} \tag{12}$$

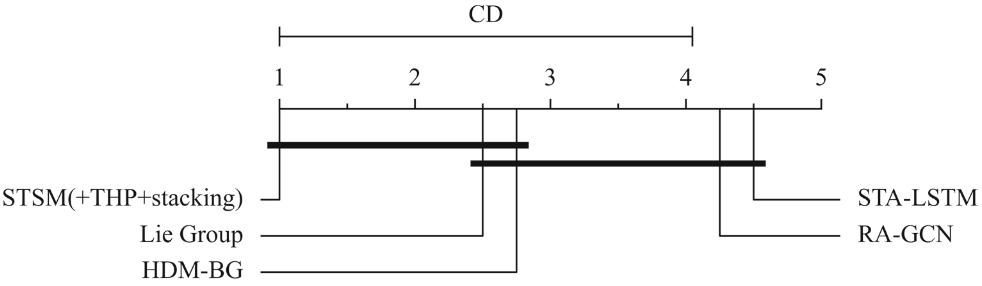
Applying this correction, we obtain $F_F = 13.00$. The critical value of $F(4, 12)$ for $\alpha = 0.05$ is 3.259. As the value of $F_F = 13.00$ is higher than 3.259, we can reject the null hypothesis, which means there are significant differences between rival methods by using the Iman-Davenport test.

Then, we adopt the Nemenyi test to compare rival methods with each other. The Nemenyi statistic value is obtained as (13). If the average rank of the two methods is

Table 10 Ranks of comparison methods for Friedman test

Datasets	STSM(+THP+stacking)	HDM-BG	Lie Group	STA-LSTM	RA-GCN
UTK	1	4	2	5	3
MSR	1	3	2	4	5
UTD	1	2	3	4	5
G3D	1	2	3	5	4
Average ranks	1.00	2.75	2.50	4.50	4.25

Fig. 5 Statistical analysis results of Nemenyi test



at least different from the critical difference value (CD), the two methods are significantly different.

$$CD = q_{\alpha}\sqrt{\frac{k(k+1)}{6N}} \tag{13}$$

In our case, with $k = 5$ and $N = 4$ tests, the critical value for a 95% of confidence ($q_{\alpha} = 2.728$) is $CD = 3.05$. The results of the Nemenyi test are shown in Fig. 5.

Figure 5 shows the result of the Nemenyi test on comparison methods. The comparison methods are sorted by their average rank, which lower is better. We connect the groups of methods not significantly different in performance indicators by a horizontal line. The analysis reveals that STSM(+THP+stacking) is ranked lower than other methods, while it seems to have equivalent performances with Lie Group [10] and HDM-BG [32] statistically. The distance between STSM(+THP+stacking) and the other two methods (RA-GCN [46] and STA-LSTM [47]) is greater than the CD value. We can declare that our results perform significantly better compared with the results of RA-GCN [46] and STA-LSTM [47].

5 Conclusion

We present a stacking model to capture the temporal information of adjacent skeleton sequences and the spatial information of the whole skeleton joints, which combines multiple base classifiers to synthesize the performance of different classifiers. The main contributions of this paper are as follows. First, we use a structure to obtain the temporal relationship of skeleton sequences, named THP. Second, we compare the effects of six commonly used SPD matrices and find that the Laplace kernel function matrix is the best. Finally, we use the stacking model to learn a two-level classifier on the output of base classifiers and use the prediction results of the two-level classifier to recognize skeleton actions. Our method has been validated on widely used 3D action recognition datasets. Experimental results show that we have achieved state-of-the-art performance.

In the future, we will further use the ensemble learning method to improve the generalization performance of models. For example, in the first-level classifier, the matrices generated by different kernel functions are regarded as different attributes of the skeleton sequences, so that the stacking model can focus on the properties of different aspects of skeleton sequences. Another experiment that is worth a try is to replace a manual selection of kernel function with more advanced methods such as using multi-kernel metric learning method to learn hybrid kernels and so on.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (Grant No. 61977061, 51934007, and 61876186) and the National Innovation and Entrepreneurship Training Program for Undergraduate(No.201910290055Z).

References

1. Ofli F, Chaudhry R, Kurillo G, Vidal R, Bajcsy R (2014) Sequence of the most informative joints (smij) A new representation for human skeletal action recognition. J Vis Commun Image Represent 25(1):24–38
2. Ding W, Liu K, Cheng F, Shi H, Zhang B (2015) Skeleton-based human action recognition with profile hidden markov models. In: CCF Chinese conference on computer vision. Springer, pp 12–21
3. Halim AA, Dartigues-Pallez C, Precioso F, Riveill M, Benslimane A, Ghoneim S (2016) Human action recognition based on 3d skeleton part-based pose estimation and temporal multi-resolution analysis. In: 2016 IEEE international conference on image processing (ICIP). IEEE, pp 3041–3045
4. Wang P, Yuan C, Hu W, Li B, Zhang Y (2016) Graph based skeleton motion representation and similarity measurement for action recognition. In: European conference on computer vision. Springer, pp 370–385
5. Xia L, Chen C-C, Aggarwal JK (2012) View invariant human action recognition using histograms of 3d joints. In: 2012 IEEE computer society conference on computer vision and pattern recognition workshops. IEEE, pp 20–27
6. Wang J, Liu Z, Wu Y, Yuan J (2012) Mining actionlet ensemble for action recognition with depth cameras. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp 1290–1297
7. Zhu Y, Chen W, Guo G (2013) Fusing spatiotemporal features and joints for 3d action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 486–491
8. Hussein ME, Torki M, Gowayyed MA, El-Saban M (2013) Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In: Twenty-third international joint conference on artificial intelligence

9. Wang L, Zhang J, Zhou L, Tang C, Li W (2015) Beyond covariance: Feature representation with nonlinear kernel matrices. In: Proceedings of the IEEE international conference on computer vision, pp 4570–4578
10. Vemulapalli R, Arrate F, Chellappa R (2014) Human action recognition by representing 3d skeletons as points in a lie group. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 588–595
11. Hou Y, Li Z, Wang P, Li W (2018) Skeleton optical spectra-based action recognition using convolutional neural networks. IEEE Trans Circ Syst Video Technol 28(3):807–811
12. Li S, Li W, Cook C, Ce Z, Gao Y (2018) Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5457–5466
13. Li C, Cui Z, Zheng W, Xu C, Yang J (2018) Spatio-temporal graph convolution for skeleton based action recognition. In: 32nd AAAI conference on artificial intelligence, AAAI 2018, pp 3482–3489
14. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139
15. Lv F, Nevatia R (2006) Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In: European conference on computer vision. Springer, pp 359–372
16. Bloom V, Makris D, Argyriou V (2012) G3D: A gaming action dataset and real time action recognition evaluation framework. In: IEEE computer society conference on computer vision and pattern recognition workshops
17. Bloom V, Makris D, Argyriou V (2014) Clustered spatio-temporal manifolds for online action recognition. In: 2014 22nd international conference on pattern recognition. IEEE, pp 3963–3968
18. Bloom V, Argyriou V, Makris D (2013) Dynamic feature selection for online action recognition. In: International workshop on human behavior understanding. Springer, pp 64–76
19. Islam MS, Bakhat K, Khan R, Iqbal M, Ye Z (2021) Action recognition using interrelationships of 3d joints and frames based on angle sine relation and distance features using interrelationships. Appl Intell: 1–13
20. Oreifej O, Liu Z (2013) Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In: 2013 IEEE conference on computer vision and pattern recognition, pp 716–723
21. Devanne M, Wannous H, Berretti S, Pala P, Daoudi M, Del Bimbo A (2015) 3d human action recognition by shape analysis of motion trajectories on riemannian manifold. IEEE Trans Cybern 45(7):1340–1352
22. Nie S, Wang Z, Ji Q (2015) A generative restricted boltzmann machine based method for high-dimensional motion data modeling. Comput Vis Image Underst 136:14–22
23. Zhu W, Lan C, Xing J, Zeng W, Li Y, Shen L, Xie X (2016) Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In: Proceedings of the AAAI conference on artificial intelligence, vol 30
24. Shahroudy A, Liu J, Ng TT, Wang G (2016) Ntu rgb+d: A large scale dataset for 3d human activity analysis. pp 1010–1019
25. Ke Q, Bennamoun M, An S, Sohel F, Boussaid F (2017) A new representation of skeleton sequences for 3d action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3288–3297
26. Li C, Zhong Q, Xie D, Pu S (2018) Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. arXiv:1804.06055
27. Wang P, Li W, Li C, Hou Y (2018) Action recognition based on joint trajectory maps with convolutional neural networks. Knowl Based Syst
28. Huang Z, Van Gool L (2017) A riemannian network for spd matrix learning. In: Proceedings of the AAAI conference on artificial intelligence, vol 31
29. Huang Z, Wan C, Probst T, Van Gool L (2017) Deep learning on lie groups for skeleton-based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6099–6108
30. Yang Y, Deng C, Tao D, Zhang S, Liu W, Gao X (2016) Latent max-margin multitask learning with skelets for 3-d action recognition. IEEE Trans Cybernet 47(2):439–448
31. Yang Y, Deng C, Gao S, Liu W, Tao D, Gao X (2016) Discriminative multi-instance multitask learning for 3d action recognition. IEEE Trans Multimed 19(3):519–529
32. Zhao R, Xu W, Su H, Ji Q (2019) Bayesian hierarchical dynamic model for human action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7733–7742
33. Memmesheimer R, Theisen N, Paulus D (2020) Gimme' signals: Discriminative signal encoding for multimodal activity recognition. arXiv, pp 10394–10401
34. Liu M, Yuan J (2018) Recognizing human actions as the evolution of pose estimation maps. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 1159–1168
35. Tang Y, Tian Y, Lu J, Li P, Zhou J (2018) Deep progressive reinforcement learning for skeleton-based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5323–5332
36. Gao X, Hu W, Tang J, Liu J, Guo Z (2019) Optimized skeleton-based action recognition via sparsified graph regression. In: The 27th ACM international conference
37. Ding Y, Zhu Y, Wu Y, Jun F, Cheng Z (2019) Spatio-Temporal attention lstm model for flood forecasting. In: Proceedings - 2019 IEEE International Congress on Cybermatics: 12th IEEE International Conference on Internet of Things, 15th IEEE International Conference on Green Computing and Communications, 12th IEEE International Conference on Cyber, Physical and So, pp 458–465
38. Islam MM, Iqbal T (2020) HAMLET: A hierarchical multimodal attention-based human activity recognition algorithm
39. Ding C, Liu K, Cheng F, Belyaev E (2021) Spatio-temporal attention on manifold space for 3D human action recognition. Appl Intell 51(1):560–570
40. Ting KM, Witten IH (1999) Issues in stacked generalization. J Artif Intell Res 10:271–289
41. Li W, Zhang Z, Liu Z (2010) Action recognition based on a bag of 3d points. In: 2010 IEEE computer society conference on computer vision and pattern recognition-workshops. IEEE, pp 9–14
42. Chen C, Jafari R, Kehtarnavaz N (2015) Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In: 2015 IEEE International conference on image processing (ICIP). IEEE, pp 168–172
43. Müller M, Röder T, Clausen M, Eberhardt B, Krüger B, Weber A (2007) Documentation mocap database hdm05

44. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830
45. Yang Y, Deng C, Tao D, Zhang S, Liu W, Gao X (2017) Latent max-margin multitask learning with skelets for 3D action recognition. IEEE Trans Cybern 47(2):439–448
46. Song Y-F, Zhang Z, Wang L (2019) Richly activated graph convolutional network for action recognition with incomplete skeletons. In: 2019 IEEE international conference on image processing (ICIP). IEEE, pp 1–5
47. Ding Y, Zhu Y, Wu Y, Jun F, Cheng Z (2019) Spatio-temporal attention lstm model for flood forecasting. In: 2019 international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE Cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData). IEEE, pp 458–465
48. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
49. Nemenyi PB (1963) Distribution-free multiple comparisons. Princeton University, Princeton
50. Iman RL, Davenport JM (1980) Approximations of the critical region of the fbietkan statistic. Commun Stat-Theory Methods 9(6):571–595

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Yufeng Zhong received the B.S. degree in computer science and technology from China University of Mining and Technology, Jiangsu, China, in 2021. He is currently pursuing the master's degree with the University of Chinese Academy of Sciences, Beijing, China. His research interests include deep learning for action recognition, person re-identification, and image captioning.

Qiuyan Yan received the Ph.D. degree from the China University of Mining and Technology in 2010. She is currently an Associate Professor with the China University of Mining and Technology. Her current research interests include multi-modal action recognition, big data analytics for education, and series data mining.