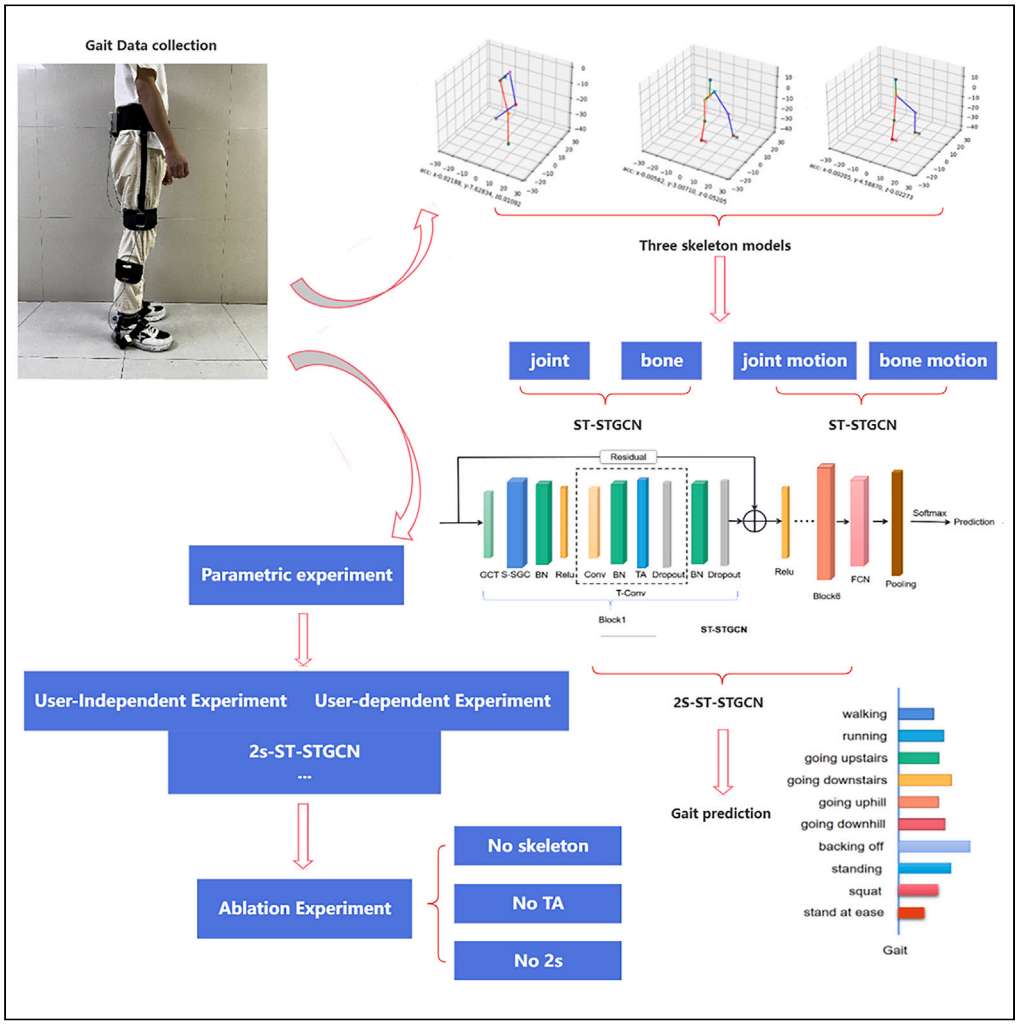
iScience

CellPress
OPEN ACCESS

Article

Spatial and temporal attention embedded spatial temporal graph convolutional networks for skeleton based gait recognition with multiple IMUs



Jianjun Yan, Weixiang Xiong, Li Jin, Jinlin Jiang, Zhihao Yang, Shuai Hu, Qinghong Zhang

jjyan@ecust.edu.cn (J.Y.)
2905430944@qq.com (W.X.)

Highlights

A skeleton-based gait recognition approach with IMUs using 2s-ST-STGCN is proposed

Different ways of constructing the human skeleton based on multiple IMUs are explored

The two-stream architecture is used to capture different gait patterns

About 99% accuracy of gait recognition is achieved in user experiments

Yan et al., iScience 27, 110646
September 20, 2024
https://doi.org/10.1016/j.isci.2024.110646

iScience

CellPress
OPEN ACCESS

Article

Spatial and temporal attention embedded spatial temporal graph convolutional networks for skeleton based gait recognition with multiple IMUs

Jianjun Yan,[1,3,*] Weixiang Xiong,[1,*] Li Jin,[2] Jinlin Jiang,[2] Zhihao Yang,[1] Shuai Hu,[1] and Qinghong Zhang[1]

SUMMARY

Gait recognition is one of the key technologies for exoskeleton robot control, while the current IMU-based gait recognition methods only use inertial data and do not fully consider the interconnections of human spatial structure and human joints. In this regard, a skeleton-based gait recognition approach with inertial measurement units using spatial temporal graph convolutional networks with spatial and temporal attention is proposed. A human forward kinematics solver module was used for constructing different human skeleton models and a temporal attention module was added for capturing the more important time frames in the gait cycle. Moreover, the two-stream structure was used to construct spatial temporal graph convolutional networks with spatial and temporal attention for gait recognition, and an average accuracy of about 99% was obtained in user experiments, which is the best performance compared to other algorithms, provides certain reference for gait recognition and real-time control of exoskeleton robots.

INTRODUCTION

Exoskeleton robots are intelligent wearable devices that are tightly integrated with the human body through the binding structure and carry out human-computer interaction and collaborative work to realize the functions of strength enhancement, body protection, and assisted movement for the wearer, and assist the wearer in accomplishing specific tasks. Exoskeleton robots can be categorized into three types according to their applications: power-assisted exoskeleton robots,[1] weight-bearing exoskeleton robots[2] and rehabilitation exoskeleton robots.[3] Lower limb-assisted exoskeletons are used to help people with lower limb fatigue to move, to protect normal people's joints from injury or to enhance their walking ability, which requires the exoskeleton to be able to judge the human body's intention of movement in advance, so as to cooperate with and assist the human body's movement.

Accurately judging the wearer's motion intention is an important prerequisite for realizing the precise control of exoskeleton robots, and only with a motion sensing system that can accurately judge the user's motion intention can a more reasonable and efficient control strategy be adopted.[4] Nowadays, the commonly used motion sensing systems mainly include electromyography signal sensing system,[5] electroencephalogram (EEG) signal sensing system,[6] and inertial measurement unit (IMU) sensing system.[7] And it is becoming more and more common to use IMU to form motion sensing systems for exoskeletons. The Israeli team, ReWalk Robotic, introduced ReWalk Personal 6.0,[8] which is mainly controlled by back posture sensors, and when it detects a change in the user's center of gravity, it immediately actuates the motors in the hip and knee joints to complete a specific movement, which is used in the rehabilitation of patients with spinal cord injuries. Indego[9] is a robot developed by Parker Hannifin, which can assist stroke patients in rehabilitation training. This exoskeleton can control separate modules through gyroscopes and sensors to realize the movement requirements in different modes. IMU motion sensing system is not easy to be interfered by the outside world, and the application is more mature, so this paper also adopts IMU to build the motion sensing system of the exoskeleton. The IMU-based exoskeleton sensing system provides hardware foundation for subsequent gait recognition research.

RESULTS

Related work

Gait recognition algorithm is one of the key technologies of human motion sensing system, and previous research mainly focuses on the field of machine learning. Nguyen et al.[10] used distributed plantar pressure sensors to obtain human motion information, and used K nearest neighbors (KNN) classification method to realize five kinds of motion pattern recognition such as walking on a flat surface, walking up the stairs, descending the stairs, going up the slope, and going down the slope. Liu et al.[11] used inertial sensors to collect motion data in real time, calculate group class correlation coefficients between motion data and template data, and apply hidden Markov model to identify

[1]Shanghai Key Laboratory of Intelligent Sensing and Detection Technology, East China University of Science and Technology, Shanghai 200237, China
[2]Shanghai Aerospace Control Technology Research Institute, Shanghai 201108, China
[3]Lead contact
*Correspondence: jjyan@ecust.edu.cn (J.Y.), 2905430944@qq.com (W.X.)
https://doi.org/10.1016/j.isci.2024.110646

the final motion states. Piris et al.[12] incorporated in-ear accelerometer sensors for gait classification between walking and running and compared them using support vector machines and KNN classifiers. The accuracy of both final classifiers exceeded 99%, outperforming most of the previous studies.

Deep learning is a new research direction in the field of machine learning, including a variety of network structures such as convolutional neural networks (CNNs),[13] long short-term memory (LSTM),[14] and recurrent neural network (RNN),[15] which have better learning ability and adaptability, and have been widely used in the field of gait recognition. Zou et al.[16] used a cell phone to collect gait datasets of walking in a natural state. Unlike the traditional methods that usually require a person to walk along a specified path or at a normal walking speed, the proposed method collects inertial gait data in an unconstrained manner and does not need to know when, where, and how the user walks. In their experiments, the spatial feature information of the data image extracted by CNN combined with the temporal information extracted by LSTM network, and then merges the features for classification, in the experiment, the experimental accuracy reached 93.7% for 118 experimenters. Dehzangi et al.[17] proposed a new method using the time-frequency (TF) expansion of human gait cycle to recognize human gait. A deep convolutional neural network (DCNN) was designed, and the raw motion data from five inertial sensors placed on the chest, lower back, right wrist, right knee, and right ankle of each human subject were collected synchronously. Two early (input level) and late (decision scoring level) multi-sensor fusion methods were proposed for a 10-class recognition task using the optimal individual IMU and 2DTF-DCNN achieved 91% subject recognition accuracy and improved the gait recognition accuracy of the system to 93.36% and 97.06%, respectively, by the multi-sensor fusion approach. Fang et al.[18] proposed a gait neural network (GNN) based on temporal convolutional networks for gait recognition and prediction, which consists of an intermediate network, a target network, and a recognition and prediction model that can fully utilize historical information from sensors, and has shown excellent performance on publicly available HuGaDB datasets as well as data collected by inertial-based wearable motion capture devices. However, those gait recognition methods based on inertial data do not take into account the spatial connection and graph structure of the human skeleton, and cannot reflect the relationship between joints during motion, and cannot represent human gait characteristics well.

Spatial temporal graph convolutional networks (ST-GCN)[19,20] are widely used in the field of video-based action recognition. This method utilizes video methods such as OpenPose to detect human skeleton, captures the relationship between human joints during action, and constructs a ST-GCN for action recognition based on human skeleton model. Sheng et al.[21] proposed an attention augmented temporal graph convolutional network for gait-based recognition and motion prediction. With spatial and temporal attention enhancement, the proposed model can capture discriminative features in spatial dependence and temporal dynamics. Liu et al.[22] used a multi-scale strategy to extract gait features, constructed a complex spatial temporal adjacency matrix, and the proposed G3D module utilizes dense inter-temporal edges as jump connections to directly propagate information in the spatial temporal graph, and the MS-G3D Net constructed by combining these features showed excellent performance on three publicly available datasets. Yin et al.[23] proposed an end-to-end, spatial-temporal, joint attention graph convolutional network (STJA-GCN) for recognizing anomalous gaits. Accuracy of 93.17% and 92.08% were obtained on two common anomalous gait datasets, which were improved by 9.22% and 20.65%, respectively, when compared with the original ST-GCN. Chen et al.[24] proposed a novel spatial temporal adaptive graph convolutional network (STA-GCN) where two main modules are introduced, respectively. Spatial feature learning module infers context-sensitive joint implicit connection and adaptively aggregates spatial feature mined from implicit and explicit connection. Temporal feature learning module extracts and adaptively aggregates multi-scale temporal feature of joint motion. Experimental results demonstrate STA-GCN outperforms state-of-the-art methods in two tasks. These methods mentioned previously have made improvements in space, time, and graph structure learning, but they do not take into account the construction of the human skeleton under wearable sensors, and there is still a lack of focus on spatial temporal aspects. Meanwhile, the existing IMU-based gait recognition methods do not fully consider the connection between the joints of the human lower limbs. However, the spatial location of the IMU can be utilized to construct a skeleton model, thus reflecting the physical connection relationships of the human body's joints, and at the same time, the construction of the skeleton does not have to be confined to the physically fixed position of the sensor, and there are a variety of different construction possibilities, so a human forward kinematics solver module is added in the ST-GCN in this paper, which converts the IMU motion data into human node coordinates to conduct different human skeletons, and then uses the spatial temporal graph network to mine the connections between the joints of the human lower limbs, capture time-domain relationships and spatial-domain relationships in gait, and extract the patterns of the different gaits for the prediction of each gait. And the addition of the temporal attention mechanism also allows the network to capture more important time frames within the gait cycle, thus improving the performance of the network. Therefore, in order to improve the robustness and accuracy of gait recognition based on inertial data, spatial and temporal attention embedded spatial temporal graph convolutional networks with two-stream (2s-ST-STGCN) is proposed for skeleton-based gait recognition with multiple IMUs.

We improved the existing acquisition device[25] through adding IMUs for gaining the motion data of ankles, and re-collected the motion data of ten gaits of human body, and carried out the research of gait recognition of lower limb exoskeleton robot based on ST-GCN. A human forward kinematics solver module was added to the network, and different human skeleton models were constructed for experiments, At the same time, the two-stream structure was introduced to simultaneously consider node data, skeleton data, and motion data, and a temporal attention module was added to focus on more important time frames in the gait cycle. Experiments were conducted to discuss the effects of different skeleton models on the network, and the effects of window size and shift size on the performance of the human gait recognition model were explored, and user-independent and user-dependent experiments were conducted, compared with other algorithms, and the ablation experiment was conducted to validate the enhancement of the model brought by the added modules.

Figure 1. Lower limb assisted exoskeleton robot

Methods and data

Lower limb assisted exoskeleton robots

The structure schematic of the lower limb assisted exoskeleton[26] developed in our laboratory is shown in Figure 1, which adopts a rigid mechanical support structure, installing inertial measurement units in the waist, thigh, calf, and ankle of the exoskeleton robot to collect real-time gait data of the human body; arranges a motor-driven structure in the knee joint to provide the human body with the assisting torque; places a battery at the waist to supply power for the controller and actuator; installs the controller and other components at the back of the exoskeleton to carry out the gait recognition of the human body and adopts the corresponding control strategy to realize the real-time control of the exoskeleton robot.

Human gait information acquisition equipment

In this paper, the existing human gait information acquisition device was improved by adding inertial sensors at the ankle, and its structure is shown in Figure 2, which consists of IMU at the waist, thighs, calves and ankles, a coprocessor, and a flexible binding structure, and the inertial sensors communicate data with the coprocessor via a can bus. Among them, the IMU is ICM20948, and the main control chip model of the coprocessor is STM32F767IGT6, and the physical object is shown in Figure 2. The data sampling frequency is 100 HZ. So the three-axis acceleration, three-axis angular velocity, and three-axis angle of the human waist, thigh, calf, and ankle are transmitted to the coprocessor through the inertial sensors, and the data can be transmitted to the upper computer terminal through the Bluetooth on the coprocessor, which can be used to prepare the data for the study of gait recognition.[25]

Acquisition of human gait data

Fewer inertial datasets are currently available for gait recognition and some gaits are not included, so the data are reorganized in this paper. In this paper, the gait data acquisition is done by the human gait information acquisition equipment, including the nine-axis motion data of ten gaits of human body, which are *walking*, *running*, *going upstairs*, *going downstairs*, *going uphill*, *going downhill*, *backing off*, *standing*, *squat*, and *stand at ease*.

In this paper, seven volunteers were invited to collect gait data with a sampling frequency of 100 Hz, and nine sets of data were collected, which were named as dataset 1, dataset 2, dataset 3, dataset 4, dataset 5, dataset 6, dataset 7, dataset 8, and dataset 9 of which datasets 3, 4, 5, 6, and 7 were the gait data of five different volunteers, dataset 1 and dataset 8 were the gait data of the sixth volunteer (re-dressed), and dataset 2 and dataset 9 were the gait data from the seventh volunteer (re-dressed), and the specifics of the data are shown in Table 1. The

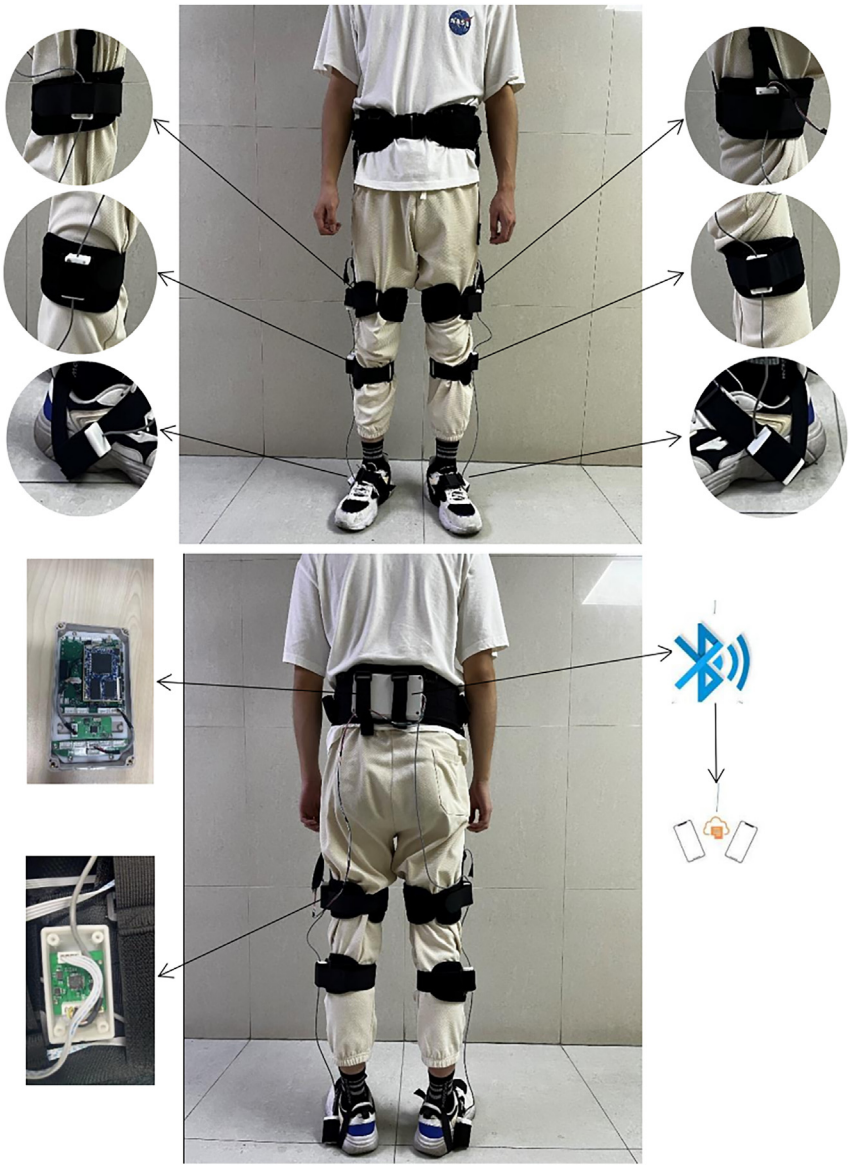


Figure 2. Human gait information acquisition device

schematic diagram of data acquisition is shown in Figure 3, and each dataset includes three-axis motion angles, three-axis acceleration, and three-axis angular velocity of waist, thigh, calf, and ankle, in which the sensor and the human body coordinate system are specified as shown in Figure 4, and both of them are based on the side of the human body, with x axis for the direction of motion parallel to the sagittal plane of the human body, y axis for the direction of gravitational acceleration, and z axis for the direction perpendicular to the sagittal plane of the human body.[25]

ST-STGCN based gait recognition

In this paper, a spatial temporal graph convolutional gait recognition method based on spatial temporal attention and IMU is proposed. The flow of the algorithm is as follows: firstly, gait data are obtained through IMUs; secondly, gait data are converted to joint coordinates through human forward kinematics solving module; thirdly, human lower limb skeleton model is established, and spatial temporal graph convolutional networks with spatial and temporal attention is constructed for gait recognition based on the two-stream structure; finally, the result of gait recognition is output. Therefore, the experimental process of the algorithm in this paper can be shown in Figure 5 further.

Table 1. Status of data

Activity	Time sec (min)	Percent (%)
Walking	1000(17)	10
Running	1000(17)	10
Going upstairs	1000(17)	10
Going downstairs	1000(17)	10
Going uphill	1000(17)	10
Going downhill	1000(17)	10
Backing off	1000(17)	10
Standing	1000(17)	10
Squat	1000(17)	10
Stand at ease	1000(17)	10
Total	10000(170)	100

Positive kinematics solution for the human body. The human forward kinematics solution obtains the position of each joint of the human body through the human joint lengths, relative joint rotations, and root node coordinates to get the posture of the whole skeleton for gait recognition.

In this part of the paper, the 3D coordinates of the lower limb joints will be obtained using the motion angles of the hip, knee, and ankle joints in the aforementioned dataset for subsequent experiments. Since the centers of the hip and waist are relatively fixed in human body during motion, there are two skeleton construction methods for the hip joint: one is to construct it in its natural state, separating the hip and waist and considering its small rotation in the horizontal plane; the other is to merge the hip and waist together without considering its rotation in the horizontal plane. Meanwhile, although this paper mainly focuses on the motion characteristics of the human lower limbs, the tilt of the human torso in motion will also reflect the motion of the gait to a certain extent, such as when going uphill, the human body will be slightly leaning forward, therefore, in order to express the motion angle of the end joints in the skeleton, it is necessary to lead out virtual nodes at the end joints, i.e., the chest node and the foot node. In summary, in order to explore a better representation of the skeleton model, this paper will consider three representations of the skeleton model: (1) separating the waist node from the hip node, but not to lead out the virtual chest node and foot node, and focus only on the hip node, knee node, and ankle node, which have the most significant motion of the lower limb, so as to establish a 7-node skeleton model; (2) combining the hip node with the waist node on the basis of 7-node, without considering the rotation of the hip joint in the horizontal plane, but eliciting the virtual chest node and the foot node, which are used to represent the tilting of the human body's torso and the movement of the foot in the movement, respectively, to establish an 8-node skeleton model; (3) on the basis of eliciting the virtual chest node and foot node, the waist node is separated from the hip node, and the rotation of the hip joint in the horizontal plane is taken into account to establish a 10-node skeleton model; all the aforementioned nodes are converted from the joint angle and joint length.

Figure 3. Schematic of data collection (from left to right: walking, going upstairs, going downstairs)

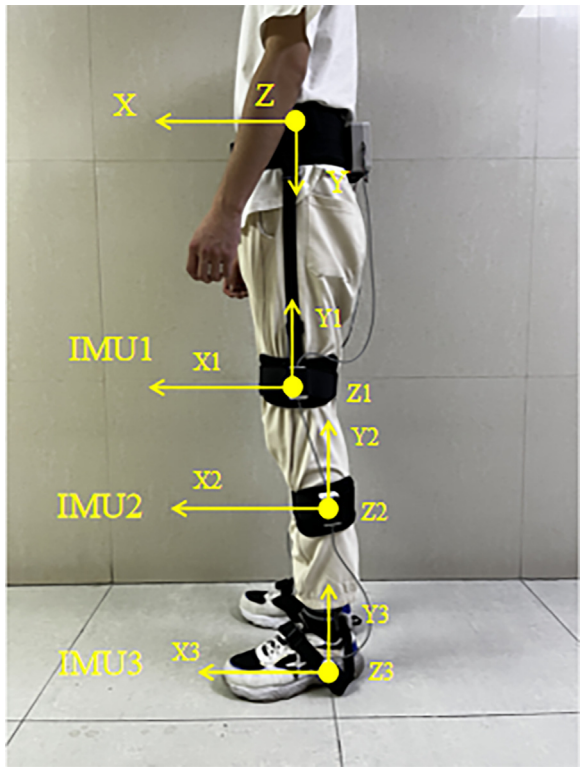


Figure 4. Sensor and human body coordinate system

The skeleton model is constructed in the way shown in Figure 6 taking the left leg of 10-node skeleton as an example (7-node skeleton lacks one chest node and two foot nodes, and 8-node skeleton merge the hip node with the waist node), defining the length from waist to chest as L1, the length from waist to hip as L2, the length of the thigh as L3, the length of the calf as L4, and the length of the foot as L5, and the angle of human body's tilting, the rotation angle of the thigh and the rotation angle of the calf, and the rotation angle of the foot in the sagittal plane

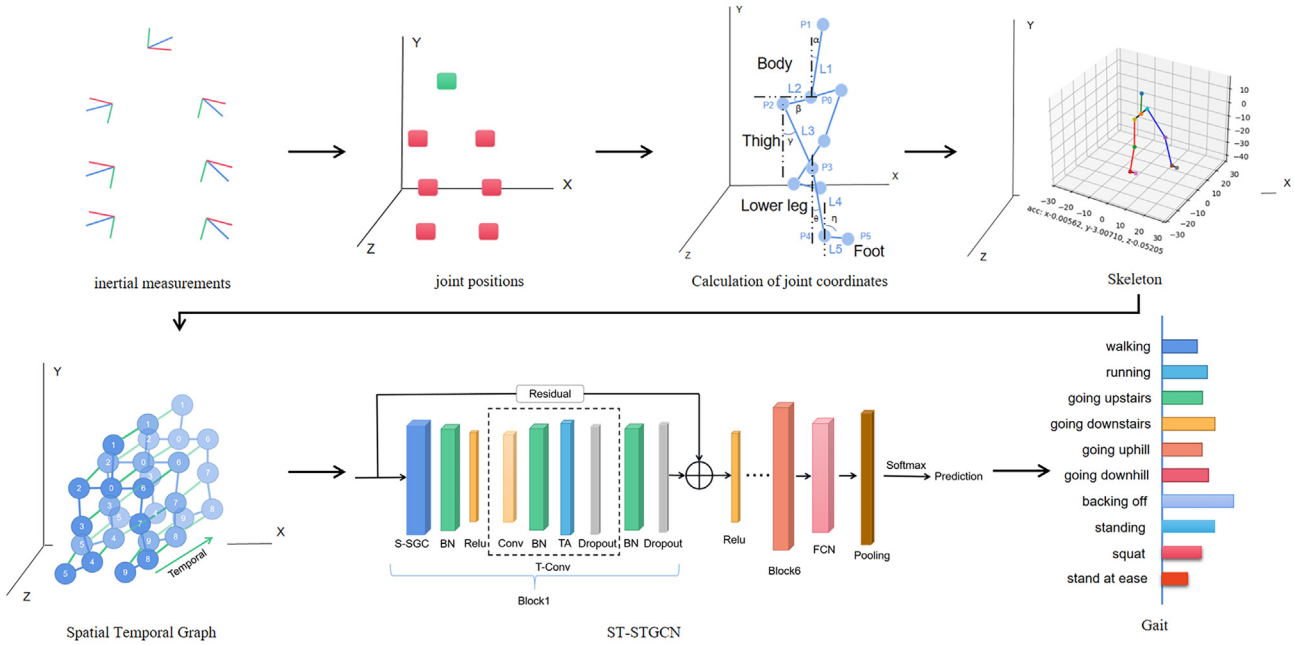


Figure 5. Flow of gait recognition

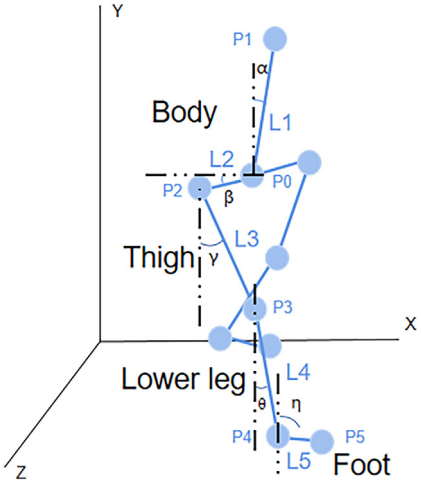


Figure 6. Node information conversion

are α, γ, θ, η, respectively, and the rotation angle of the hip with respect to the waist in the horizontal plane is β. Since the human body mainly moves in the sagittal plane, the motion in the horizontal plane is dominated by the hip joint, and other joints are not taken into account. From this, the waist coordinate P0 is defined as (0, 0, 0) as the base, the chest coordinate P1 can be calculated by L1 and angle α, the hip coordinate P2 can be obtained by L2 and β, the knee coordinate P3 can be calculated by L2 and γ as well as P2, the ankle coordinate P4 can be calculated by P3, L4, and θ, and the foot coordinate P5 can be calculated by P4, L5, and η, and the right leg adopts similar case processing, the human body node information is obtained as shown in Figure 7 (the joint length is averaged from the normal human body), and the calculation process is as follows:

$$P0 = (0,0,0)$$

$$P1 = (L1 * \sin(\alpha), 0, L1 * \cos(\alpha))$$

$$P2 = (\sin(\beta) * L2, 0, \cos(\beta) * L2)$$

$$P3 = (P2.x + \sin(\gamma) \times L3, P2.y + \cos(\gamma) * L3, P2.z)$$

$$P4 = (P3.x + \sin(\theta) \times L4, P3.y + \cos(\theta) * L4, P2.z)$$

$$P5 = (P4.x + \sin(\eta) \times L5, P4.y + \cos(\eta) * L5, P2.z)$$

Modeling the human skeleton. Previous studies have shown that fusing joint information with skeletal information facilitates gait recognition. The human skeleton can be expressed as the difference in coordinates between two connected joints. Take 3D joint coordinates as an

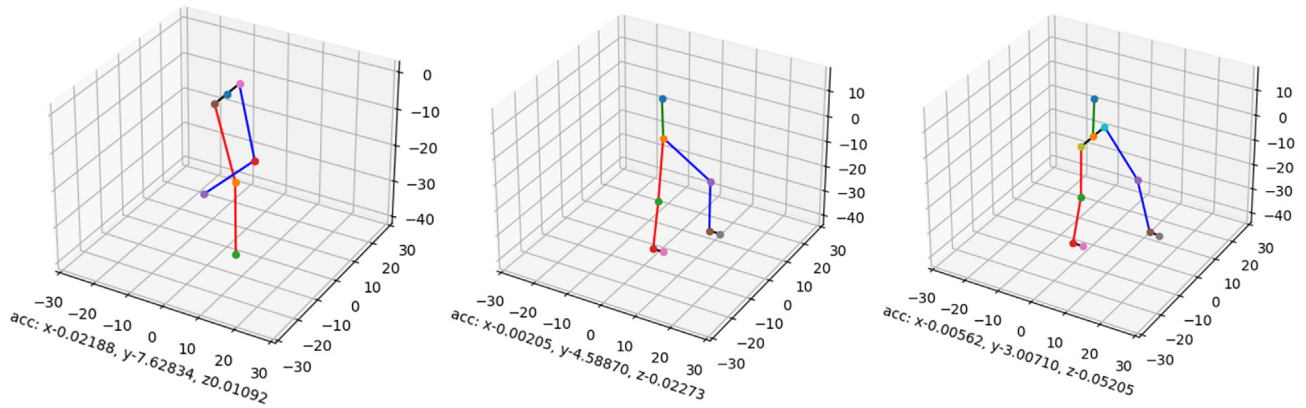


Figure 7. Different skeleton models for walking (from left to right: 7 nodes, 8 nodes, 10 nodes)

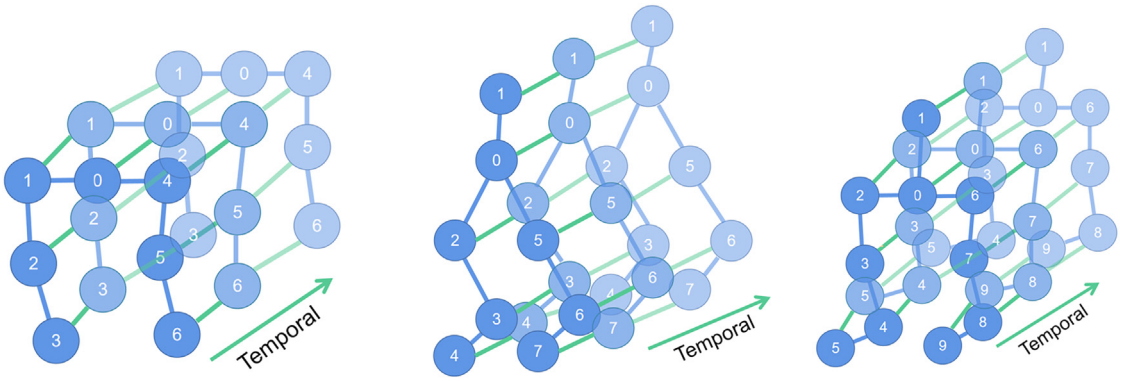


Figure 8. Spatial temporal graph (from left to right: 7 nodes, 8 nodes, 10 nodes)

example: the original joint data contains coordinates in both x and y directions. Given two nodes P1 = (x1, y1, z1) and P2 = (x2, y2, z2), then the skeleton formed by connecting two joints P1 and P2 can be expressed as the vector difference between the two joints, i.e., $B_{p1, p2}$ = (x1-x2, y1-y2, z1-z2), expanding to each node. But for the motion gait, we have to consider both the joint position and the joint rotation. After the joint position has been estimated, combined with the corresponding inertial measurement output to get the joint rotation. Therefore, taking walking as an example, the three skeleton construction methods to get the skeleton model are shown in Figure 7 below. As shown in the figure, the 7-node skeleton model separates the hip joint node from the waist node, taking into account the motion angle of the hip joint in the horizontal plane, but not the foot movement of the human body. On the contrary, the 8-node skeleton model takes into account the foot movement of the human body, but does not consider the rotation of the hip joint in the horizontal plane. Therefore, the 10-node skeleton model is further introduced. This model simultaneously considers the rotation of the hip joint in the horizontal plane and the

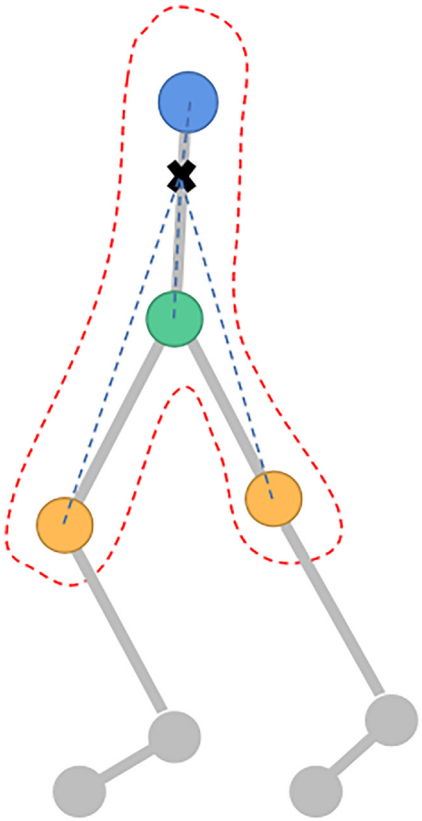


Figure 9. Spatial configuration partitioning (nodes are labeled according to their distance to the skeleton center of gravity (black fork) and the root node (green)

The distance to the centripetal node is shorter (blue) and the distance to the centrifugal node is farther (yellow).

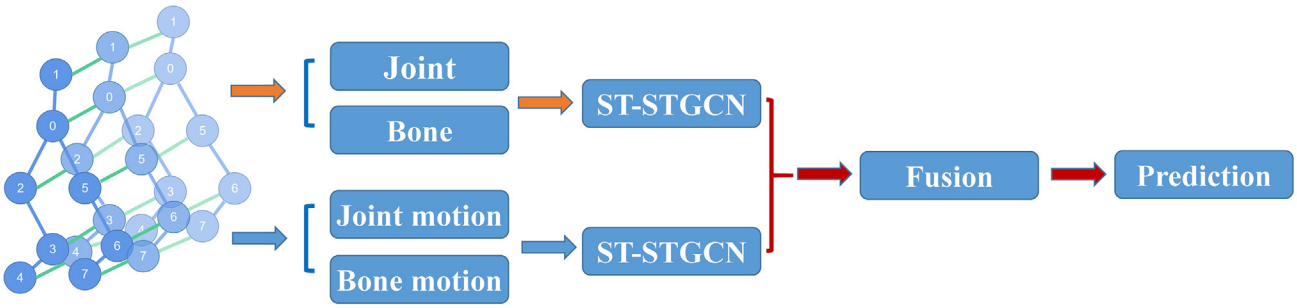


Figure 10. Two-stream structure, two ST-STGCN taking joint information, bone information, and joint motion information, and bone motion information as inputs, respectively, and the output tensor is fused to predict gait labels

movement of the human foot. In the future, these three skeleton models will be used for experiments to explore better skeleton construction methods.

Graph structure and graph convolution. A graph is a data structure consisting of a series of nodes and edges. As non-Euclidean data, graph data have no regular spatial structure and its complexity poses a great challenge to existing machine learning algorithms.[27] Recently, GNNs have been used in a variety of difficult tasks for previous machine learning algorithms because of their ability to model data generated from non-Euclidean domains and capture the internal correlation of the data and have been widely successful.[28,29] Encouraged by the success of CNN in computer vision, graph convolutional neural networks (GCN)[30] have attracted more people. These methods can be divided into spectral methods[31] and spatial methods.[32] Spectral methods define graph convolution by using the graph Fourier transform. Spatial methods use the topology of the graph directly and apply convolution filters based on the neighborhood information of the graph.

The core of graph convolution is to do matrix multiplication, and the matrix it uses is the adjacency matrix. In terms of image processing, the convolution operation uses a number of convolution kernels (filter/kernel) of fixed size to scan the input image. Near the center pixel of each scan, a pixel matrix of the same size as the weight matrix is extracted, and the feature vectors on these pixels are stitched in spatial order and inner product with the parameter vectors of the convolution kernel to obtain the convolution output value at that location. Here, a nearby pixel can be defined as a neighborhood on a grid of pixels. When extending the convolution operation on an image to an arbitrary graph structure, we can also define the neighborhood and a series of weight matrices of any node. This is the basic idea of graph convolution networks.

However, unlike images, the number of nodes in the neighborhood of each node is not fixed if an adjacency matrix is used to define the neighborhood on ordinary graph structures (the pixels near the pixels on the image are always fixed when considering the complement 0). This makes it difficult for us to determine: (1) the dimensionality of the parameters of the convolution kernel to be used; (2) how to arrange the weight matrix with the nodes in the neighborhood to perform inner product operations. In the original GCN article, the authors proposed to turn the inner product operation into one that computes: the inner product using the same vector with all the feature vectors on the points in the neighborhood and averages the result. This allows: (1) the parameters of the convolution kernel to be determined as a vector of fixed length; (2) the order of the nodes in the neighborhood need not be considered. This design allows GCN to be used on graphs with arbitrary connectivity relations and has achieved good performance in some tasks such as network analysis and semi-supervised learning.

Spatial-temporal graph convolutional network. A ST-GCN was proposed for solving the human action recognition problem based on key points of the human skeleton.[33] In addition to the novelty, the approach achieved a large performance improvement on standard action recognition datasets. The basis of ST-GCN is the spatial temporal graph structure. The idea of constructing a spatial temporal graph from a sequence of skeleton key points is derived from existing skeleton action recognition methods and image recognition methods. Most of the existing skeleton-based action recognition methods introduce some spatial structure information to improve the recognition accuracy, including the connection relationship of adjacent key points or body parts such as hand-elbow-shoulder connection relationship. To model this spatial information, existing methods often use a sequential model such as RNN to traverse the connected key points. This requires the model designer to define a rule for traversal or to define some body parts manually. But in such design, it is difficult to get an optimal traversal rule or partitioning of components. The natural connection relationship between key points actually constitutes a natural graph structure. Therefore, ST-GCN actually proposed a spatial temporal graph construction method with the following rules: (1) inside each frame, the spatial graph was constructed according to the natural skeleton connection relationship of the human body; (2) the same key points in two adjacent frames were connected to form a temporal edge; (3) all key points in the input frames formed a node set, and all edges in steps 1 and 2 form an edge set, i.e., the required temporal graph. The spatial temporal graph constructed by the three skeleton models in this paper is shown in Figure 8.

In order to perform skeleton action recognition on the spatial temporal graph, it is also necessary to define the convolution operation on the spatial temporal graph, so the original paper proposed three spatial partitioning rules, which were uni-labeling, distance partitioning, and spatial configuration partitioning. The three types of division mainly lie in the different ways of dividing the node neighborhoods. The third

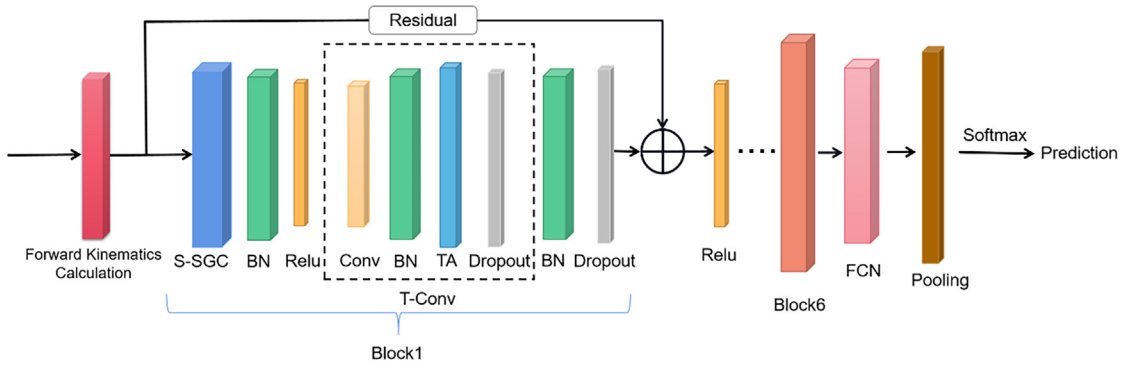


Figure 11. Illustration of the overall structure of the ST-STGCN

Residual connectivity is used to ensure the stability of training. S-SGC denotes the spatial graph convolutional layer with self-attention enhancement. T-Conv denotes the temporal convolutional layer with temporal attention.

partitioning method divides the 1 neighborhood of a node into 3 subsets, and the first subset is the node itself, and the second is the set of neighboring nodes that are closer to the center of gravity of the whole skeleton than this node in spatial location, and the third is the set of neighboring nodes that are further away from the center of gravity. The establishment of this division rule lies in the definition of centripetal and centrifugal motions according to the study of motion analysis. This division is also used in this paper, as shown in Figure 9 (taking 8-node skeleton as an example).

Self-attention enhanced spatial graph convolution layer. The spatial-temporal graph convolution in this paper has been partially defined by ST-GCN previously, while in the message passing process it is necessary to consider not only the local neighborhood nodes predefined by the natural connectivity of the human body, but also other nodes with high information relevance. Therefore, the S-ST-GCN[20] added a self-attention enhanced spatial graph convolutional layer on top of the ST-GCN, which consisted of three parts: a graph convolutional part, a self-attention part, and a gating mechanism, where the self-attention mechanism calculated the weighted sum of the values of all nodes to aggregate the features of the whole graph and provided supplemental information to the spatial graph convolutional module.

Temporal convolution layer with temporal attention. However, in the human gait recognition task, it is necessary to focus not only on the spatial structure of the human body, but also to consider which time frames in the gait cycle are more important, so it is necessary to incorporate a temporal attention mechanism. A temporal attention mechanism[34] is incorporated in this paper to help improve the performance. The temporal attention part is added within the original temporal convolutional layer, so that a temporal convolutional part, a temporal attention part and a gating mechanism constitute the temporal convolutional layer with temporal attention. The basic mechanism of the temporal attention part can be represented by the following equation:

$$E = V_e \cdot \sigma\left(\left(\left(\chi_h^{(r-1)}\right)^T U_1\right) U_2\left(U_3 \chi_h^{(r-1)}\right) + b_e\right) \quad \text{(Equation 1)}$$

That is, the inputs of time steps T are transposed into vector multiplications in the time dimension and the correlation degree between different times is calculated. This allows considering time frames within the gait cycle that are more relevant to the gait.

Two-stream structure. In some previous work on action recognition, in addition to joint positions, second-order features, i.e., skeletal information representing the length and orientation of the human skeleton, have also proved to be useful for skeleton-based action recognition tasks. Therefore, in gait recognition tasks, skeletal information may also play an important role and needs to be considered as well. At the same time, for some gaits, motion information such as velocity and acceleration may become equally important for differentiation between the two, and thus we constructed a two-stream network to use joint information, bone information, and motion information of both simultaneously.

Skeletal vectors have been constructed previous, so the two-stream spatial temporal attention 2s-ST-STGCN is shown in Figure 10. The joint data representing the joint position and the bone data representing the length and orientation of the bones, as well as the motion data of both, are input into the two ST-STGCN, and the neighbor matrix A is learned separately, and the edge_importance (a matrix of weights of edges, which is used to give a larger weight to important edges in the neighbor matrix and suppresses the weights of non-important edges) is learned, i.e., each data stream is trained separately, and then the output tensor of the two data streams is fused to predict the gait labels.

The basic block of ST-STGCN consisted of the self-attention enhanced spatial graph convolutional layer, the temporal convolutional layer with temporal attention and several functional layers. The self-attention enhanced spatial graph convolutional layer was used to aggregate information of joints along the spatial dimension, and the temporal convolutional layer with was temporal attention used to aggregate information along the temporal dimension. The whole S-ST-GCN structure in the original paper consisted of three parts: an input layer of

Table 2. Experimental environment configuration

Name	Configuration Information
OS	Windows 10
Hardware	CPU: AMD R7-5800H
	Memory: 16GB
	Graphs card: RTX3060, 6GB
Python library	Python 3.8
	Pytorch 1.8.1
	Sklearn 0.24.2
	Numpy 1.20.0
	Pandas 1.2.5
	Matplotlib 3.4.2
	Keras 2.4.3

BatchNorm; nine basic graph convolution blocks; and a fully convolutional network (FCN). But this paper did not use all the basic graph convolution blocks, only six basic graph convolution blocks were used. In this paper, we use the overall structure of the ST-STGCN as shown in Figure 11. The coordinate conversion module converts the IMU data into human skeleton data and constructs the spatial-temporal graph in the subsequent network according to the previous rules. After the constructed spatial temporal graph are convolved by six graph convolution modules, respectively, the output tensor is input to the global average pooling layer to obtain the feature vector for each gait. Finally, the vectors are passed to the output layer by the softmax function to obtain the prediction of gait categories.

Experiment

Experimental setup

We converted the joint angle data of the previous ten gaits into joint coordinates by human forward kinematics solving, conducted experiments on the effect of window size and shift size, and applied MiniRocket,[35] time series transformer (TST),[36] temporal convolutional network (TCN),[37] LSTM,[14] RNN,[15] five purely data-driven time series algorithms for user-independent and user-dependent experiments, and compared with 2s-ST-STGCN to demonstrate the superiority of 2s-ST-STGCN proposed in this paper for gait recognition.

In order to verify the feasibility of the proposed human gait recognition method based on 2s-ST-STGCN, in this part of the paper, three types of experiments are conducted: (1) user-independent experiment: using the motion data of five different volunteers as the training set, using the motion data of another two volunteers as the test set, and comparing the algorithms; (2) user-dependent experiment: using the motion data of seven different volunteers as the training set, using the motion data of the two volunteers of the seven volunteers after re-wearing as the test set, to compare the algorithms and to compare with experiment (1); (3) ablation experiment: remove the human forward kinematics solving module, the time attention module and the two-stream structure for user-independent and user-dependent experiments, respectively, to validate the performance enhancement of the improvements made in this paper.

In this paper, training and testing platform for gait recognition models was built on Windows 10, mainly implemented using Python and Pytorch. Table 2 shows the specific hardware configuration information of the experimental environment.

Experiment of skeleton construction

In order to choose a better skeleton representation, the previously mentioned construction methods of three skeleton models were experimented separately, and the other parameters of 2s-ST-STGCN were kept consistent, and the results are shown in Table 3. It can be seen from the results that under the skeleton model representation of 8 nodes, the accuracy of the model is the highest, 98.307%, which is 0.518% and 0.419% higher than that of 7 nodes and 10 nodes, respectively, and the F1-score, which combines recall and precision, is 98.308%, which is 0.532% and 0.419% higher than that of 7 nodes and 10 nodes, respectively, which indicates that under the 8-node the network model performance is better. Therefore, the skeleton model expression of 8 nodes will be used in this paper.

Table 3. Results of experiment of number of nodes

Number of nodes	Accuracy (%)	Recall (%)	Precision (%)	F_1-score (%)
7 nodes	97.789	97.799	97.748	97.776
8 nodes	**98.307**	**98.311**	**98.321**	**98.308**
10 nodes	97.888	97.890	97.889	97.889

Table 4. Impact of shift size on algorithm performance

Shift size	Accuracy (%)	Recall (%)	Precision (%)	F_1-score (%)
10	**98.407**	**98.411**	**98.421**	**98.408**
20	98.116	98.119	98.146	98.116
30	97.876	97.876	97.939	97.873
40	98.037	98.040	98.090	98.041
50	97.796	97.799	97.869	97.772
60	98.017	98.028	98.073	98.009

Experiment of shift size

Since the data processing method used in this paper is data sliding window, it is necessary to explore the effect of both window size and window shift size on the model. Since the sampling frequency of the sensor used in this experiment is 100Hz, a normal person walks about 100~120 steps per minute, and a gait cycle is about 1s, so the window size should vary around 100. Therefore, in order to better investigate the effect of shift size on the performance of the algorithm, this paper sets the window size to 100, and changes only the shift size of the training data, and unifies the shift size of the predicted data to 1. So we change the shift size to 10, 20, 30, 40, 50, and 60, respectively, for the experiment, using datasets 1, 2, 3, 4, and 5 as the training set and dataset 6 and 7 as the test set, and 90% of the data were used for training and 10% for validation, with 500 training sessions each time, and the learning rate was set to 0.0001. The results are shown in Table 4.

The average accuracy of the shift size of 10 is 98.407%, which is 0.291%, 0.531%, 0.370%, 0.611%, and 0.390% higher than shift sizes of 20, 30, 40, 50, and 60, respectively. Moreover, its F1-sorce of 98.408% was also the highest when shift size was 10, so shift size was set to 10 in all subsequent experiments.

Experiment of window size

It has shown that the window size affects the experimental results when adding windows to the data. Therefore, when other parameters remain unchanged, experiments on the effect of window size on the performance of 2s-ST-STGCN algorithm were conducted to explore the optimal window size. It has been mentioned that the window size should be varied around 100, so in this part the window size is set to 60, 70, 80, 90, 100, 110, 120, and 130, respectively, and the shift size is set to 10 to keep it unchanged, using datasets 1, 2, 3, 4, and 5 as the training set and dataset 6 and 7 as the test set. The obtained results are shown in Table 5.

The average accuracy is 98.978% that of the window size is 80, which is 0.961%, 0. 72%, 0.27%, 1.102%, 0.121%, 0.041%, and 0.181% higher than the window sizes of 60, 70, 90, 100, 110, 120, and 130, respectively. And its F1-score is 98.979% when the window size is 80, which is also the highest and the best performance, so the window size is set to 80 in the subsequent experiments.

User-independent experiment

The optimal window size and shift size settings have been obtained from the previous experiments. In this section, an experimental comparison between the 2s-ST-STGCN and other algorithms will be conducted. In this section, the raw three-axis angular velocity and three-axis acceleration are used as inputs to MiniRocket, TST, TCN, LSTM, and RNN, respectively, and experimental comparisons are performed separately.

In this part, we used datasets 1, 2, 3, 4, and 5 as the training set and dataset 6 and 7 as the test set, and the window size and shift size were set to 80 and 10, respectively, the results shown in Table 6.

As shown in Table 6, among the algorithms, 2s-ST-STGCN has the highest average accuracy, which is 4.107%, 11.105%, 20.041%, 19.38%, and 17.441% higher than MiniRocket, TST, TCN, LSTM, and RNN, respectively, and its F1-score of 98.979%, which is also the highest.

Table 5. Impact of window size on algorithm performance

Window size	Accuracy (%)	Recall (%)	Precision (%)	F_1-score (%)
60	98.017	98.028	98.073	98.009
70	98.258	98.259	98.274	98.255
80	**98.978**	**98.980**	**98.983**	**98.979**
90	98.708	98.710	98.721	98.708
100	97.876	97.876	97.939	97.873
110	98.857	98.858	98.862	98.858
120	98.937	98.939	98.943	98.938
130	98.797	98.799	98.802	98.799

Table 6. User-independent experimental results

Algorithm	Accuracy (%)	Recall (%)	Precision (%)	F_1-score (%)
RNN	81.537	78.887	80.439	79.016
LSTM	79.598	77.489	83.241	76.842
TCN	78.937	77.787	79.584	76.880
TST	87.873	86.543	87.754	86.287
MiniRocket	94.871	93.761	93.890	93.881
2s-ST-STGCN	**98.978**	**98.980**	**98.983**	**98.979**

User-dependent experiment

This part is basically the same as the user-independent experiments in terms of experimental setup except for the use of datasets. This part uses datasets 1, 2, 3, 4, 5, 6, and 7 as training sets and dataset 8 and 9 as test set for user-dependent experiments, and the results are obtained as shown in Table 7.

As shown in Table 7, 2s-ST-STGCN has the highest average accuracy among the algorithms, which is 1.288%, 7.699%, 13.595%, 11.181%, and 15.403% higher than MiniRocket, TST, TCN, LSTM, and RNN, respectively, and also achieves the highest with its F1-score of 99.159%.

Ablation experiment

The ST-STGCN has been compared with other algorithms in the previous section, but it has not been verified that the modules added in this paper bring performance improvement to the model. Therefore, in this part, in order to verify the performance enhancement brought to the model by the human forward kinematics solving module as well as the temporal attention module, experiments are carried out on the ST-GCN containing only the human forward kinematics solving module (S-STGCN-Y) and the ST-GCN containing only the temporal attention module (ST-STGCN-N). In addition, to verify the performance of the two-stream structure proposed in this article, ST-STGCN (including the human forward kinematics solving module and temporal attention module) with one-stream was also tested, and the results are obtained as shown in Table 8, and the parameters of the experiments are the same as those described in the previous section.

As shown in Table 8, the average accuracy of 2s-ST-STGCN is highest, which is 0.102% higher than ST-STGCN in user-independent experiment, and in user-dependent experiment, the average accuracy of 2s-ST-STGCN is 0.049% higher than ST-STGCN, indicating that the two-stream structure brings positive gains. At the same time, the average accuracy of ST-STGCN is 0.29% and 0.279% higher than S-STGCN-Y and ST-STGCN-N in user-independent experiment, respectively. And in user-dependent experiment, ST-STGCN is 0.232% and 0.011% higher than S-STGCN-Y and ST-STGCN-N, respectively. The results of the experiment suggest that both the human forward kinematics solving module and the temporal attention module improve the model's performance.

DISCUSSION

Effect of skeleton construction method on algorithm performance

As can be seen from Table 3, the algorithm performs better using the 8-node skeleton model expression pattern than using the 7-node and 10-node skeleton model expression patterns. To further analyze it, the difference between the skeleton model of 10 nodes and the skeleton model of 8 nodes is that 10 nodes separate the hip nodes from the waist nodes and consider the rotation of the hip joints in the horizontal plane, but adding this information its results are not improved but the difference is not big, probably the similarity of the hip joint rotation angle between different gaits leads to this result; and the difference between it and the skeleton model of 7 nodes is that in 10 nodes, virtual chest node and foot node are added, and extra human torso information is added, such as the human body leans back when going downstairs and leans forward when going up. The virtual chest node is able to represent the inclination degree of the torso tilt, which is more helpful for the expression of human gait and facilitates the differentiation of different gaits, and thus the results are better than those at 7 nodes. To further analyze the results, the mean value of the torso tilt angle for each gait is calculated, as shown in Table 9.

Table 7. Results of user-dependent experiment

Algorithm	Accuracy (%)	Recall (%)	Precision (%)	F_1-score (%)
RNN	83.756	81.437	82.853	80.112
LSTM	87.978	87.487	86.511	86.332
TCN	85.564	84.787	85.223	85.778
TST	91.460	90.190	92.464	90.287
MiniRocket	97.871	96.598	97.687	97.881
2s-ST-STGCN	**99.159**	**99.160**	**99.163**	**99.158**

Table 8. Results of ablation experiment

Algorithm	Accuracy (%)	Recall (%)	Precision (%)	F_1-score (%)
Independent experiment				
S-STGCN-Y	98.586	98.589	98.593	98.588
ST-STGCN-N	98.597	98.602	98.609	98.600
ST-STGCN	98.876	98.880	98.883	98.879
2s-ST-STGCN	**98.978**	**98.980**	**98.983**	**98.979**
Dependent experiment				
S-STGCN-Y	98.878	98.881	98.887	98.879
ST-STGCN-N	99.099	99.107	99.106	99.099
ST-STGCN	99.110	99.092	99.123	99.108
2s-ST-STGCN	**99.159**	**99.160**	**99.163**	**99.158**

As can be seen from Table 9, the average tilt angle of the torso under different gaits is different, for example, for the three gaits of *running*, *going upstairs*, and *going uphill*, the average angle between the human torso and the positive direction of the x axis is less than 90°, which indicates that the human body is tilted forward at this time, but the average angle between the human torso and the positive direction of the x axis is more than 90° for the two gaits of *going downstairs* and *going uphill*, indicating that the human body is leaning back, which effectively helps the algorithm to differentiate between these similar gaits, so the addition of the virtual chest node helps to improve the performance of the algorithm.

Effect of shift size on algorithm performance

Appropriately changing the shift size improves the performance of the 2s-ST-STGCN algorithm. From Table 4, it can be seen that the average accuracy of gait recognition is the lowest when the shift size is 50, which is 97.796%; when the shift size is 10, the average accuracy of gait recognition is the highest, which is 98.407%, which is 0.611% higher than the shift size of 50.

In order to further analyze the effect of shift size change on the performance of gait recognition, the recognition accuracy of each gait in different shift size was calculated, as shown in Figure 12.

As can be seen from Figure 12, in the process of changing the shift size, 2s-ST-STGCN recognizes most of the gaits stably, and the recognition accuracies are around 97%, and only *running* is relatively lower. From the confusion matrix (Figure 13), the relatively low recognition accuracy of *running* is because it is recognized as *going downstairs* and *going upstairs* indicating that there is still some error between similar gaits due to the similarity of data. However, the performance of 2s-ST-STGCN is good for different shift sizes, so the performance of the spatial temporal graph convolution algorithm with spatial temporal attention proposed in this paper is not greatly affected by the parameter of shift size, and the gait model has a strong robustness, which can reduce the computational amount of finding the appropriate shift size for different data in practical use.

Effect of window size on algorithm performance

Different window size affects the performance of 2s-ST-STGCN algorithm. As can be seen from Table 5, 2s-ST-STGCN performs the worst when the window size is 100, with an average accuracy of 97.876%, and 2s-ST-STGCN performs the best when the window size is 80, with

Table 9. Mean values of trunk tilt angle for each gait

Gait	Mean waist angle (°)
Walking	95.684
Running	84.823
Going upstairs	86.978
Going downstairs	97.691
Going uphill	86.676
Going downhill	108.640
Backing off	89.465
Standing	86.575
Squat	71.809
Stand at ease	93.047

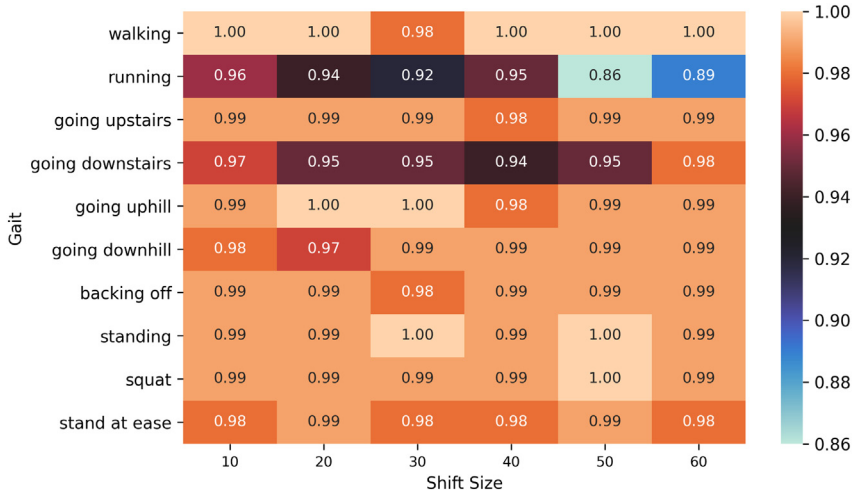


Figure 12. Recognition accuracy of each gait during shift size changing

an average accuracy of 98.978%, which is 1.102% higher than the window size of 100. To further analyze the effect of window size on the accuracy of gait recognition, the accuracy of each gait recognition under different window sizes was calculated, as shown in Figure 14.

As shown in Figure 14, similar to shift size, 2s-ST-STGCN also recognizes most of the gaits stably with recognition accuracies around 97% during the process of changing the window size, still only *running* is relatively low. Similarly, from the confusion matrix (shown in Figure 15), the relatively low recognition accuracy of *running* is because it is recognized as *going upstairs* and *going downstairs*. But overall, for different window sizes, the performance of 2s-ST-STGCN is good, so the performance of the spatial temporal graph convolution algorithm with spatial temporal attention proposed in this paper is not greatly affected by the parameter of window size, and the gait model has a strong robustness, which can reduce the computational amount of finding the appropriate window size for different data in practical use.

Analysis of results of user-independent experiment

2s-ST-STGCN performed the best in the user-independent experiment with an average recognition accuracy of 98.978%. To further analyze the performance of each algorithm, the recognition accuracy of each gait was calculated, as shown in Figure 16. 2s-ST-STGCN has higher and more stable recognition accuracy than the other five algorithms in most gait recognition, and the lowest accuracy is around 98%, while the other algorithms even have the lowest accuracy of 23%. The accuracy of 2s-ST-STGCN on the nine gaits of *walking*, *running*, *going upstairs*, *going downstairs*, *going uphill*, *backing off*, *standing*, *squat*, and *stand at ease* are 100%, 99%, 99%, 98%, 99%, 99%, 99%, 100%, and 98%, respectively, ranking first, and the accuracy of *going downhill* is 98%, ranking second, indicating that 2s-ST-STGCN outperforms other algorithms in each gait, and its high accuracy and stability are more suitable for gait recognition.

The experimental results of 2s-ST-STGCN are further analyzed, and its confusion matrix is shown in Figure 15. 2s-ST-STGCN has high recognition accuracy for all ten gaits, but there are still cases of misrecognition of gaits such as *going downhill*, such as the misrecognition rate of 1% for recognizing *going downhill* as *walking*, etc., and the other five algorithms have higher misrecognition rates for these gaits, which is the main reason why the average recognition accuracy of the other algorithms is relatively low.

The experimental results of each algorithm are further analyzed. In terms of data, the reason for the misrecognition of these gaits is that because the data used in this paper are the three-axis motion angle, three-axis acceleration and three-axis angular velocity of the human lower limb. In the actual movement process, the difference in the motion characteristics of these gaits in the leg is not very large, so there is a misrecognition of these gaits with each other, resulting in a lower average accuracy rate. However, there is also a difference in the average recognition accuracy of each algorithm when using the same data, indicating that the characteristics of the model itself are also an important factor affecting the performance of gait recognition.

For 2s-ST-STGCN, in order to highlight the role played by the model in feature extraction, the original data and the middle layer of the trained model were, respectively, downscaled to a two-dimensional plane using the T-SNE method and visualized, and the results are shown in Figure 17. As can be seen from the figure, the ten features of the original data had overlapping distributions, and the distinction between different categories was not obvious, while after the network extraction, the entropy values of the ten features were greatly reduced, and they gradually formed communities in their respective categories and were well separated, and especially the categories of *stand at ease* and *squat* were obviously separated from other categories, so the better experimental results were obtained. From the model, it was the same as ST-GCN with two important points: the first one was the evolution from understanding the skeleton sequence as a frame of the skeleton to understanding the whole dataset as a whole spatial-temporal graph, which made it possible to analyze the action with a unified model, and this idea was also used in this paper to construct the inertial data as a spatial-temporal graph of the human lower limbs skeleton, and it can better express the motion characteristics of each gait, so it can unify the recognition of each gait. And the spatial-temporal graph network

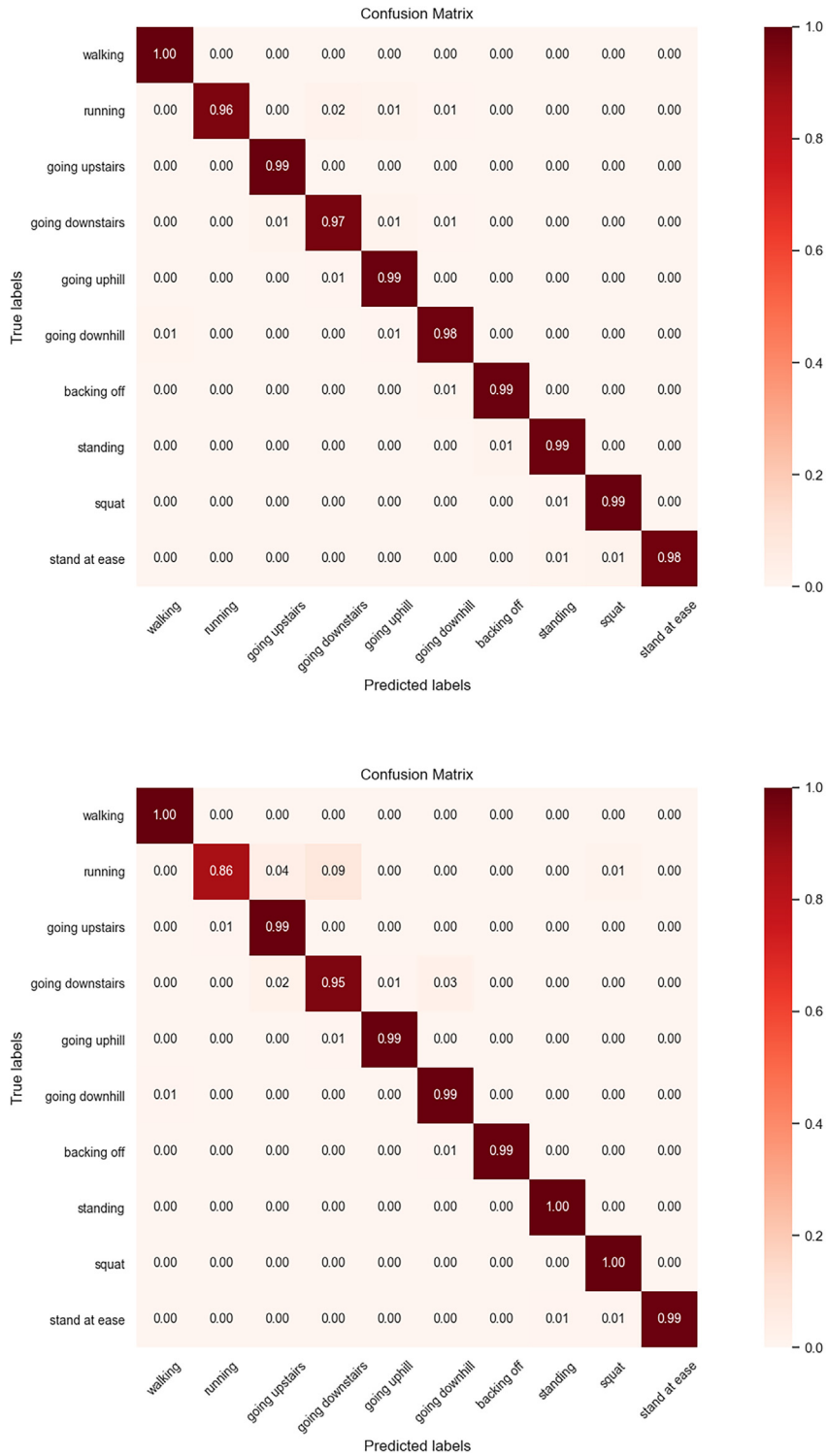


Figure 13. Confusion matrix for different shift size (top: 10, bottom: 50)

itself possesses the ability to capture this spatial location relationship and temporal relationship, easily finding the movement patterns of different gaits with strong robustness; the second one was the evolution from the plain idea of the original GCN to the definition of convolution using division rule-based convolution. This idea made ST-GCN can surpass the original GCN and got huge performance improvement.

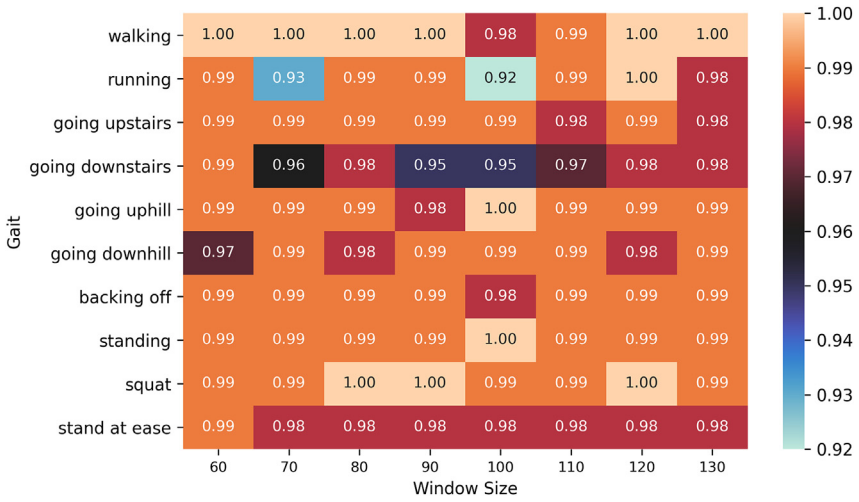


Figure 14. Recognition accuracy of each gait during window size changing

Based on the previous, the 2s-ST-STGCN proposed in this paper not only has a spatial attention mechanism that can take different weights for different neighborhoods of human skeleton nodes and consider other nodes with higher relevance, but also adds a temporal attention mechanism that enables it to consider time frames within the gait cycle that have a greater impact on motion, thus obtaining better performance, and in this experiment, the recognition accuracy of each gait is above 98%.

Meanwhile, in order to represent the learning for the adjacency matrix A in 2s-ST-STGCN, the unlearned, predefined adjacency matrix A and the learned, multiplied and stacked adjacency matrix *A′* with edge_importance (learned weight matrix used to give larger weights to important edges in the adjacency matrix and to suppress the weights of non-important edges) are extracted, respectively. And due to the adoption of the spatial configuration partitioning, max_dis_connect is defined as 1 and the number of nodes is 8. Therefore, three 8*8 matrices representing the weights between the root node and the root node, the root node and the centripetal node, and the root node and the centrifugal node, respectively, are generated, which expresses the stationary motion characteristics of the whole gait, centripetal motion characteristics and centrifugal motion characteristics, which are combined together to form an adjacency matrix graph, as shown in Figure 18.

It can be seen from Figure 18 that by multiplying and stacking with edge_importance, the weights of the three types of edges represented by the adjacency matrix *A′* have been changed, and the weights of the edges closer to the end of the limb (farther away from the center of gravity) have been increased, which is more useful for expressing different gaits, and conforms to the laws of human gait movement, thus achieving better results.

For the other algorithms, RNN still has the gradient vanishing and gradient explosion problems of recurrent networks, and performs poorly when dealing with larger data volumes and longer period data. LSTM improves on RNN, and is able to avoid the problems of gradient vanishing and gradient explosion, but it still faces challenges with parallel data processing and capturing the dependencies of long sequences. TCN allows parallel computation of data and can avoid above issues, but it still needs complete sequences and has limatations in performance due to restricted receptive field and difficulty in migration. TST supports parallel computation and possesses stronger long-term dependence modeling ability. However, it has large computational space complexity, is insensitive to local information, and is susceptible to anomalous data. As a result, the anomalous data generated during motion may result in degradation of its performance. MiniRocket does not pay attention to the temporal order of the convolutional output, and thus still produces some confusion on some gait data with similar features, which reduces its average recognition accuracy. Therefore, 2s-ST-STGCN performs best among six algorithms.

Analysis of results of user-dependent experiment

The generalized model for gait recognition was developed through user-independent experiments, and good results were obtained. However, this experiment did not use the user data, and although the generic model could reduce the time required in the training process, the generic model might not be applicable to a specific individual.[38,39] Therefore, user-dependent experiments were conducted to add user data to the model for training to further improve the accuracy of gait recognition.

The average accuracy of 2s-ST-STGCN in the user-dependent experiment is the highest, reaching 99.159%, which is 0.181% higher than the user-independent experiment, and the accuracies of these six algorithms have been improved in user-dependent experiment, and the highest one is the LSTM, which has been improved by 8.38%, which indicates that the user-dependent experiment is conducive to the improvement of the average recognition accuracies of each algorithm. The recognition accuracy of each algorithm for different gaits is shown in Figure 19. 2s-ST-STGCN achieves the highest recognition accuracy for all gaits, with a minimum of 97%, while the other algorithms have a minimum of even 46%. This indicates that 2s-ST-STGCN performs well on gait recognition and outperforms several other algorithms.

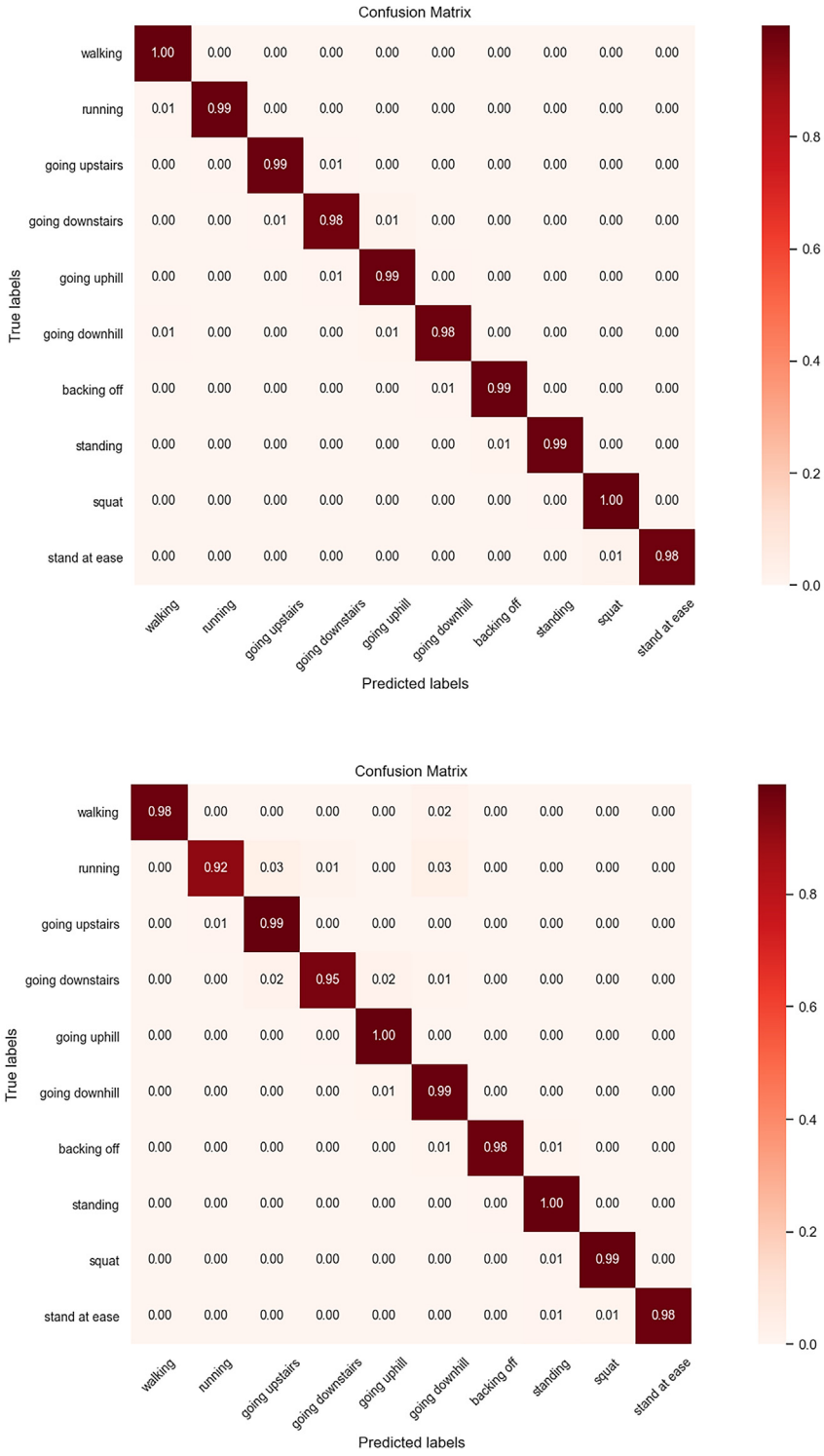


Figure 15. Confusion matrix for different window size (top: 80, bottom: 100)

The confusion matrix of 2s-ST-STGCN is shown in Figure 20. The accuracy of 2s-ST-STGCN in user-dependent experiments is highest for all ten gaits, with a minimum of 97%, which is better than the user-independent experiments, and is more stable and robust, and although there are still cases of misrecognition, they are greatly improved compared with the user-independent experiments.

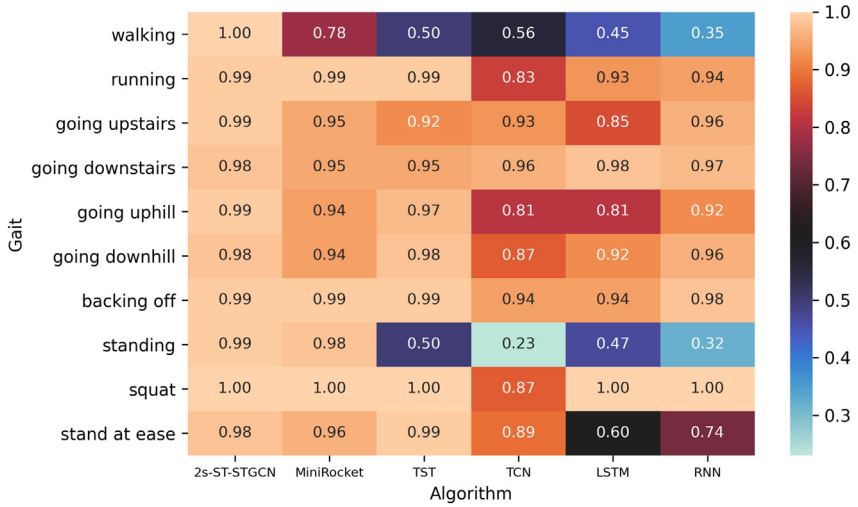


Figure 16. Gait recognition accuracy of each algorithm in user-independent experiment

For further analysis of the model, the middle layer of the trained model is also taken out, which is downscaled to a two-dimensional plane using the T-SNE method and visualized, and compared with the original data, and the results are shown in Figure 21. As can be seen from the figures, similar to the user-independent experiment, the distribution of the ten features of the original data overlapped, and the distinction between different categories was not obvious, while the entropy value of the ten features extracted by the network was greatly reduced, and they gradually formed clusters in their respective categories and were well separated, so that better experimental results were obtained. Overall, most of the gait recognition accuracies of the algorithms increased in the user-dependent experiments, among which 2s-ST-STGCN performed the best, with the highest total accuracy of 99.159%, which is an improvement of 0.181% over the user-independent experiments. Its recognition accuracy for each gait is highest, and the misrecognition rate is generally no more than 2%, with more stable and robust. This indicates that the user-dependent experiment is conducive to improving the accuracy of the algorithm for each gait, which can reduce the misrecognition rate of gait and improve the gait recognition results, and in practice, it can be considered to add user data to participate in the model training to improve the accuracy of gait recognition.

Analysis of results of ablation experiment

As can be seen from Table 8, the average accuracy of 2s-ST-STGCN is the highest in the ablation experiment, which is 0.102% higher than ST-STGCN in the user-independent experiment, and 0.049% higher than ST-STGCN in the user-dependent experiment. This is because ST-STGCN only uses joint data, but 2s-ST-STGCN considers joint data, bone data, and their motion data at the same time, and thus 2s-ST-STGCN achieves higher accuracy, indicating that the two-stream structure synthesizes more information and is more beneficial for the representation of different gaits. At the same time, the average accuracy of ST-STGCN is 0.29% and 0.279% than only the human forward kinematics solving module added (S-STGCN-Y) and only the time attention module included (ST-STGCN-N) in user-dependent experiment, and 0.232% and 0.011% higher in the user-dependent experiment, respectively. This proves that both the human forward kinematics solving module and the time attention module have improved the performance of the model. The results indicates converting inertial data to human node coordinates facilitates a better representation of the connections between human joints, enabling the network to better capture the posture of the entire human skeleton, which improves the accuracy of human gait recognition; the temporal attention module helps the model to focus on the more important temporal frames in the sequence, and assigns a greater weight to moments of each gait pattern with a higher degree of specificity, which further improves the model's performance.

In summary, this paper establishes a recognition model for ten human gaits based on 2s-ST-STGCN. The necessity of the existence of virtual nodes is verified, and the optimal window size and shift size are determined to be 80 and 10, respectively; three types of experiments are carried out: in user-independent experiment, 2s-ST-STGCN performs the best among the six algorithms, with an average gait recognition accuracy of 98.978%; in user-dependent experiment, 2s-ST-STGCN also performs the best, with an average recognition accuracy of 99.159%, which is improved by 0.181% compared with user-independent experiments, and the recognition result of each gait is improved by 0.1% compared with the user-independent experiment, and the recognition results of each gait are more stable; in the ablation experiment, the two-stream structure improved the performance of the model, and the ST-STGCN with the addition of both the human body forward kinematics solver module and the temporal attention module performs better than that when only one of the modules is added.

Conclusion and limitation

In this paper, skeleton-based gait recognition approach with inertial measurement units using ST-GCN with spatial and temporal attention is proposed to carry out the gait recognition research of human lower limb. Firstly, the human gait data acquisition equipment of the

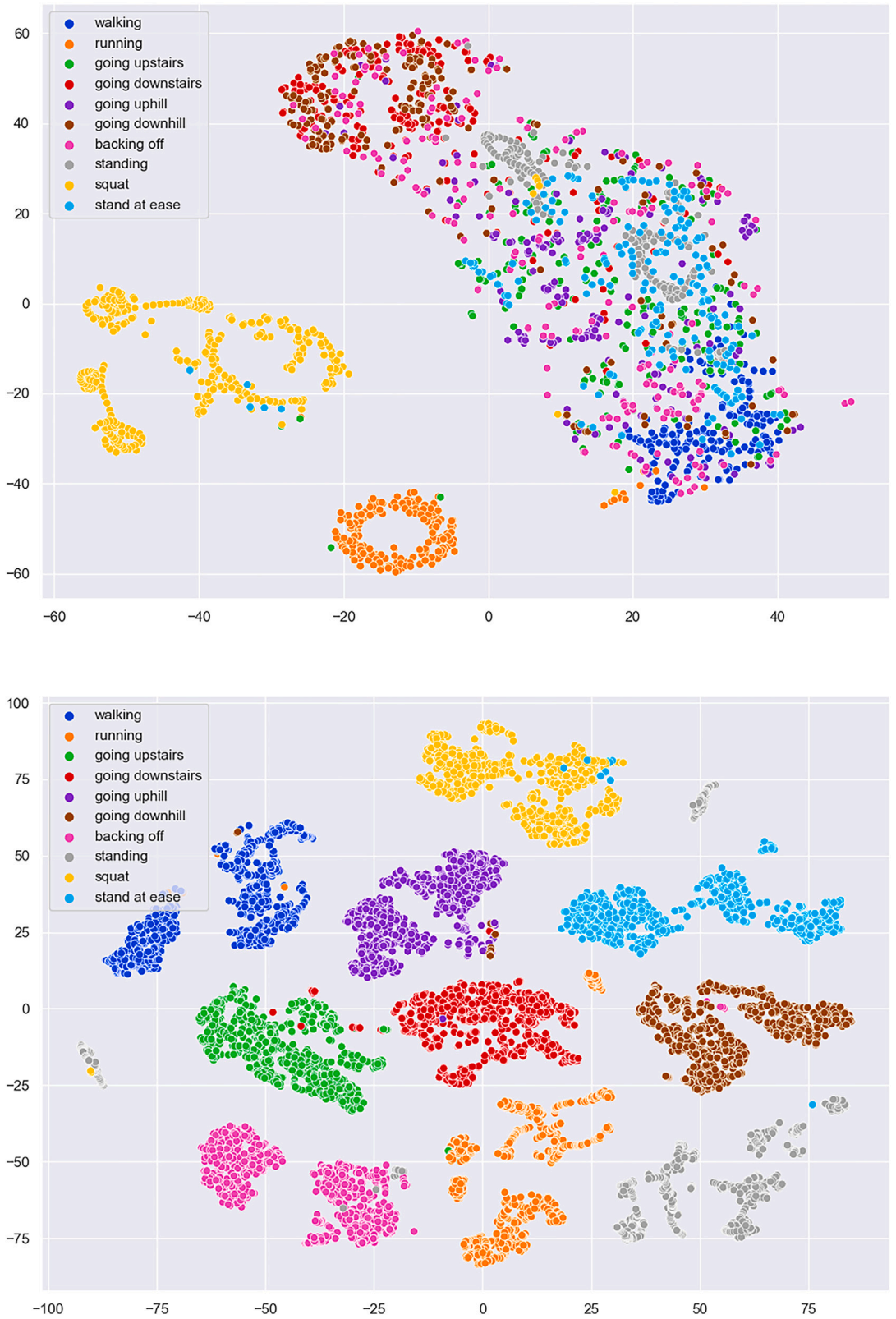


Figure 17. T-SNE feature display in user-independent experiment (top: raw data, bottom: model middle layer)

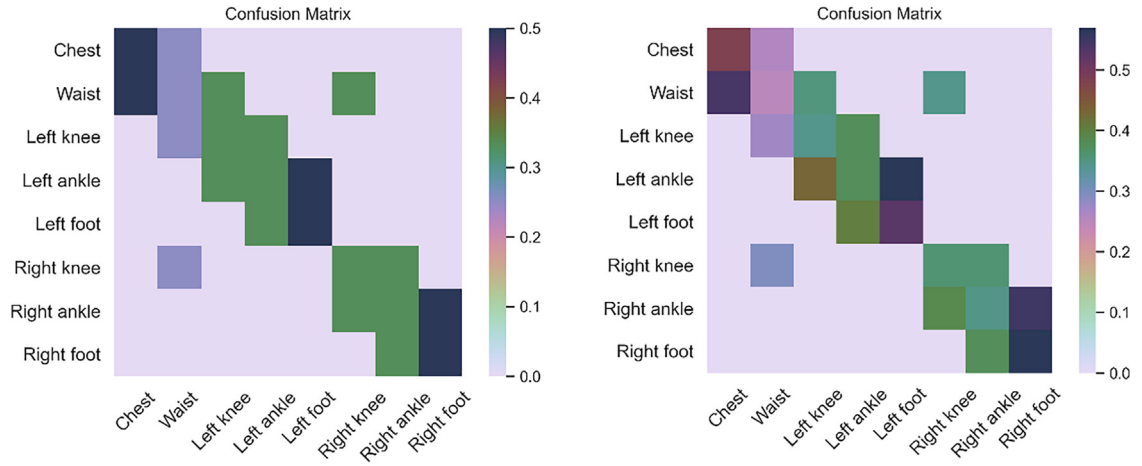


Figure 18. Diagram of the adjacency matrix (left: initial matrix, right: learned matrix)

exoskeleton robot is improved, and the motion data of ten human gaits are acquired, then different human lower limb skeleton models are established by the forward kinematics solution of the human body and the natural connection of the human body, and the spatial temporal graph of each gait are constructed by certain rules, and a two-stream structure is used to input the joint data, the skeleton data, and the motion data of the two, respectively, into the two ST-STGCN, and then the gait features are extracted using the spatial convolutional layer with self-attention enhancement and the temporal convolutional layer with temporal attention, and at last the information of the two streams are fused together to build a human gait recognition model. The experimental part discusses the necessity of the existence of virtual nodes; the effects of two types of parameters window size and shift size on the performance of 2s-ST-STGCN are investigated, and the optimal values of window size and shift size are determined; user-independent experiments and user-dependent experiments are carried out, respectively, in comparison with several algorithms, including MiniRocket, TST, TCN, LSTM, and RNN; ablation experiments are conducted to verify the effectiveness of the two-stream structure and the significance of the existence of the human forward motion solving module and the temporal attention mechanism. The experimental results prove that the average gait recognition accuracy of 2s-ST-STGCN reaches 98.978% and 99.159% in user-independent and user-dependent experiments, respectively, which is higher than that of other algorithms, and the ablation experiments show that the two-stream structure brings improvement and the human forward motion solving module and the temporal attention mechanism also bring about performance enhancement, which achieves a more satisfactory result in the gait recognition task in this paper, and can meet the requirements of the motion perception requirements of exoskeleton robots.

In this paper, we improved the temporal graph convolutional network by adding the human forward kinematics solving part and the temporal attention mechanism, but it is not comprehensive enough to learn the structure of the human skeleton graph, and the learning method of graph structure can be improved to enhance the performance of the gait recognition in the future work.

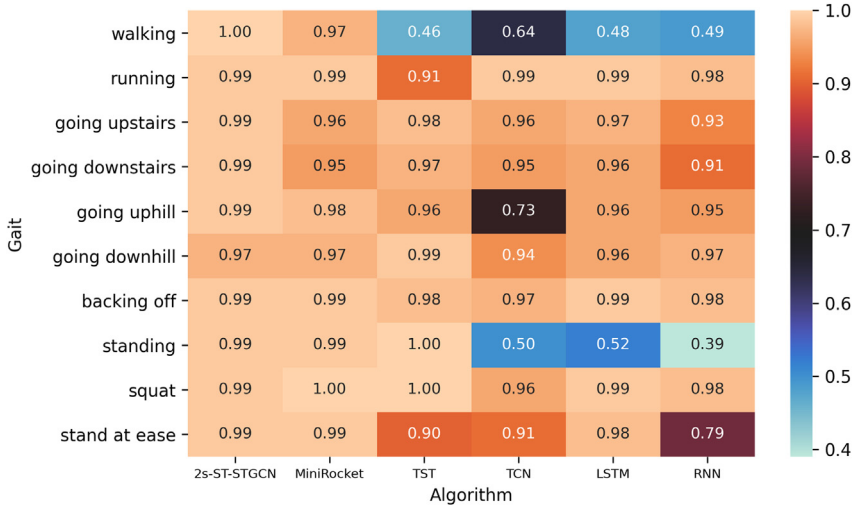


Figure 19. Gait recognition accuracy of each algorithm in user-dependent experiment

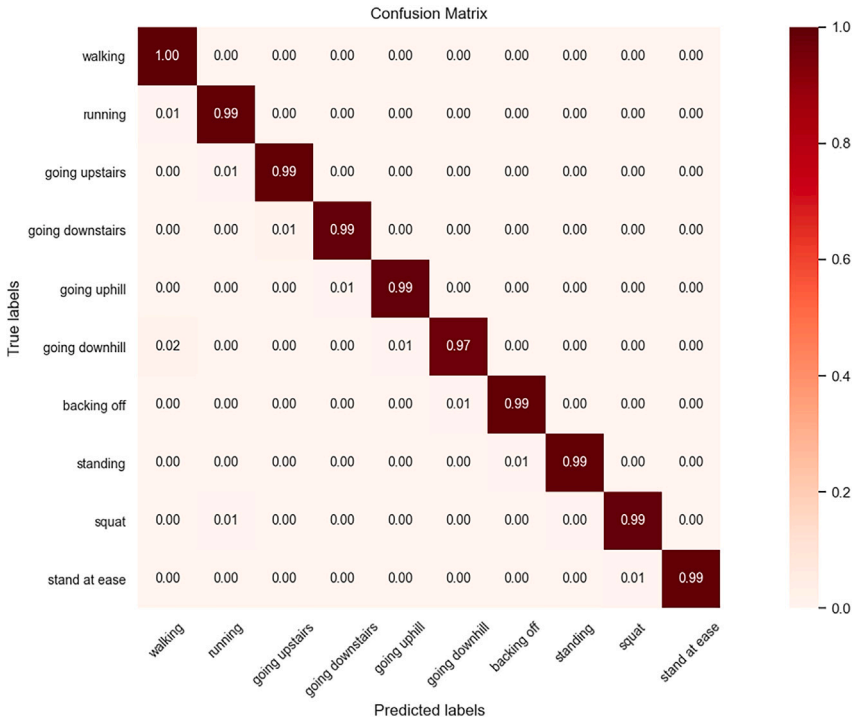


Figure 20. Confusion matrix of 2s-ST-STGCN in user-dependent experiment

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS
  - Experiment of skeleton construction
  - Experiment of shift size
  - Experiment of window size
  - User-independent experiment
  - User-dependent experiment
  - Ablation experiment
- METHOD DETAILS
  - Positive kinematics solution for the human body
  - Temporal convolution layer with temporal attention
  - Two-stream structure
- QUANTIFICATION AND STATISTICAL ANALYSIS

ACKNOWLEDGMENTS

This work was supported by the Major Research Plan of the National Natural Science Foundation of China (no. 91748110).

AUTHOR CONTRIBUTIONS

J.Y. and W.X. designed this study. L.J. and J.J. provide hardware assistance. Z.Y., J.S., and S.H., helped with the data collection and performed the data analysis. W.X. wrote the manuscript text, which was revised by J.Y. and W.X.. All authors read and approved the manuscript.

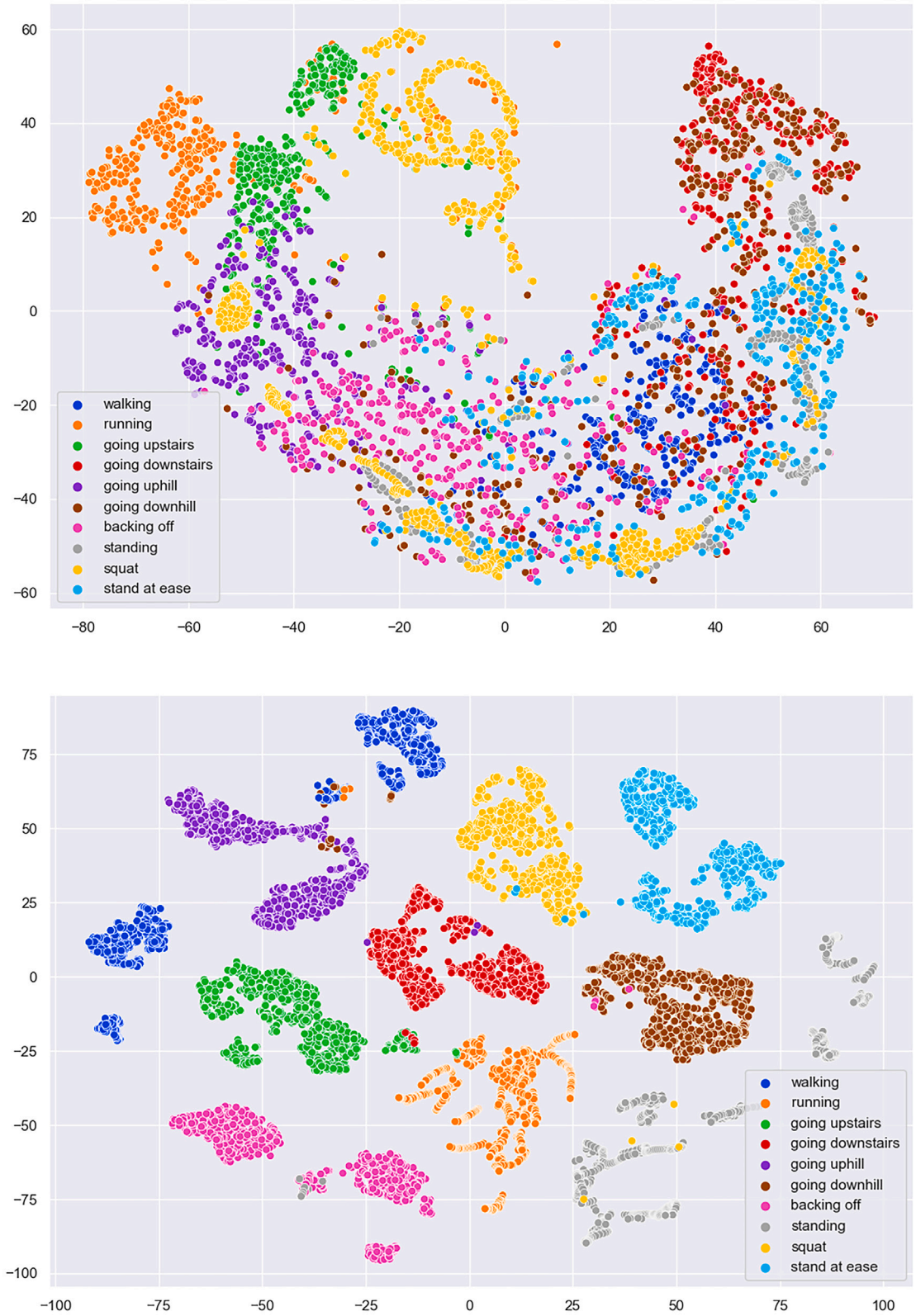


Figure 21. T-SNE feature display in user-dependent experiment (top: raw data, bottom: model middle layer)

DECLARATION OF INTERESTS

All authors declare no competing interests.

Received: January 14, 2024
Revised: May 16, 2024
Accepted: July 30, 2024
Published: August 2, 2024

REFERENCES

1. Walsh, C., Paluska, D., Pasch, K., Grand, W., Valiente, A., and Herr, H. (2006). Development of a lightweight, underactuated exoskeleton for load-carrying augmentation. In IEEE International Conference on Robotics and Automation (IEEE), pp. 3485–3491. https://doi.org/10.1109/ROBOT.2006.1642234.
2. (2018). LG to usher in new era of AI robotics with wearable robot [EB/OL]. http://www.lgnewsroom.com/2018/08/lg-to-usher-in-new-era-of-ai-robotics-with-wearable-robot/.
3. Strausser, K., and Kazerooni, H. (2011). The development and testing of a human machine interface for a mobile medical exoskeleton. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE), pp. 4911–4916. https://doi.org/10.1109/IROS.2011.6095025.
4. Li, J. (2013). Research and implementation of the exoskeleton gait detection system. In Chengdu: University of Electronic Science and Technology of China (School of Automation Engineering), pp. 1–3. https://doi.org/10.7666/d.D762886.
5. Hayashi, T., Kawamoto, H., and Sankai, Y. (2005). Control method of robot suit HAL working as operator's muscle using biological and dynamical information. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2–6. https://doi.org/10.1109/IROS.2005.1545505.
6. Contrerasvidal, J.L., and Grossman, R.G. (2013). NeuroRex: a clinical neural interface roadmap for EEG-based brain machine interfaces to a lower body robotic exoskeleton. In International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1579–1582. https://doi.org/10.1109/EMBC.2013.6609816.
7. Nuckols, R.W., Lee, S., Swaminathan, K., Orzel, D., Howe, R.D., and Walsh, C.J. (2021). Individualization of exosuit assistance based on measured muscle dynamics during versatile walking. Sci. Robot. *6*, eabj1362. https://doi.org/10.1126/scirobotics.abj1362.
8. Ramanujam, A., Momeni, K., Ravi, M., Augustine, J., Garbarini, E., Barrance, P., Spungen, A., Asselin, P., Knezevic, S., and Forrest, G.F. (2019). Center of mass adaptations and its interaction between the trunk and lower extremity during exoskeleton walking. In 2019 Wearable Robotics Association Conference (WearRAcon), pp. 57–62. https://doi.org/10.1109/WEARRACON.2019.8719398.
9. Laubscher, C.A., Goo, A., Farris, R.J., and Sawicki, J.T. (2022). Hybrid Impedance-Sliding Mode Switching Control of the Indego Explorer Lower-Limb Exoskeleton in Able-Bodied Walking. J. Intell. Robot. Syst. *104*, 76–80. https://doi.org/10.1007/s10846-022-01583-7.
10. Nguyen, N.D., Bui, D.T., Truong, P.H., and Jeong, G.M. (2018). Classification of Five Ambulatory Activities Regarding Stair and Incline Walking Using Smart Shoes. IEEE Sensor. J. *18*, 5422–5428. https://doi.org/10.1109/JSEN.2018.2837674.
11. Liu, Z., Lin, W., Geng, Y., and Yang, P. (2017). Intent pattern recognition of lower-limb motion based on mechanical sensors. IEEE/CAA J. Autom. Sinica *4*, 651–660. https://doi.org/10.1109/JAS.2017.7510619.
12. Burgos, C.P., Gärtner, L., Ballester, M.A.G., Noailly, J., Stöcker, F., Schönfelder, M., Adams, T., and Tassani, S. (2020). In-Ear Accelerometer-Based Sensor for Gait Classification. IEEE Sensor. J. *32588*, 1–19. https://doi.org/10.1109/JSEN.2020.3002589.
13. Lecun, Y. (1989). Generalization and network design strategies. Connectionism Perspect. *19*, 143–155. https://doi.org/10.48550/arXiv.2209.01610.
14. Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. Neural Comput. *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
15. Elman, J.L. (1990). Finding structure in time. Cognit. Sci. *14*, 179–211. https://doi.org/10.1207/s15516709cog1402_1.
16. Zou, Q., Wang, Y., Wang, Q., Zhao, Y., and Li, Q. (2020). Deep Learning-Based Gait Recognition Using Smartphones in the Wild. IEEE Trans. Inf. Forensics Secur. *15*, 3197–3212. https://doi.org/10.1109/TIFS.2020.2985628.
17. Dehzangi, O., Taherisadr, M., and ChangalVala, R. (2017). IMU-Based Gait Recognition Using Convolutional Neural Networks and Multi-Sensor Fusion. Sensors (Basel) *17*, 2735–2749. https://doi.org/10.3390/s17122735.
18. Fang, B., Zhou, Q., Sun, F., Shan, J., Wang, M., Xiang, C., and Zhang, Q. (2020). Gait Neural Network for Human-Exoskeleton Interaction. Front. Neurorobot. *14*, 58. https://doi.org/10.3389/fnbot.2020.00058.
19. Yan, S., Xiong, Y., and Lin, D. (2018). Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. Preprint at arXiv. https://doi.org/10.38550/arXiv.1801.07455.
20. Shi, J., Liu, C., Ishi, C.T., and Ishiguro, H. (2020). Skeleton-Based Emotion Recognition Based on Two-Stream Self-Attention Enhanced Spatial-Temporal Graph Convolutional Network. Sensors (Basel) *21*, 205. https://doi.org/10.3390/s21010205.
21. Sheng, W., and Li, X. (2021). Multi-task learning for gait-based identity recognition and emotion recognition using attention enhanced temporal graph convolutional network. Pattern Recogn. *114*, 107868. https://doi.org/10.1016/j.patcog.2021.107868.
22. Liu, Z., Zhang, H., Chen, Z., Wang, Z., and Ouyang, W. (2020). Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition (IEEE). https://doi.org/10.1109/CVPR42600.2020.00022.
23. Yin, Z., Jiang, Y., Zheng, J., and Yu, H. (2023). STJA-GCN: A Multi-Branch Spatial–Temporal Joint Attention Graph Convolutional Network for Abnormal Gait Recognition. Appl. Sci. *13*, 4205. https://doi.org/10.3390/app13074205.
24. Chen, C., and Sun, X. (2023). STA-GCN: Spatial Temporal Adaptive Graph Convolutional Network for Gait Emotion Recognition. In 2023 IEEE International Conference on Multimedia and Expo (ICME), pp. 1385–1390. https://doi.org/10.1109/ICME55011.2023.00240.
25. Yan, J., Xiong, W., Jin, L., Jiang, J., Yang, Z., Shi, J., and Hu, S. (2023). Gait Recognition Based on Minirocket with Inertial Measurement Units. Int. J. Human. Robot. *20*, 1–28. https://doi.org/10.1142/S0219843623500093.
26. Wang, Y. (2021). Gait modeling and experimental study of the human-machine system of the lower extrimity assisted exoskeleton (East China University of Science and Technology).
27. Yang, M., Zhou, M., Li, Z., Liu, J., Pan, L., Xiong, H., and King, I. (2022). Hyperbolic Graph Neural Networks: A Review of Methods and Applications. Preprint at arXiv. https://doi.org/10.48550/arXiv.2202.13852.
28. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P.S. (2021). A Comprehensive Survey on Graph Neural Networks. IEEE Trans. Neural Netw. Learn. Syst. *32*, 4–24. https://doi.org/10.1109/TNNLS.2020.2978386.
29. Bastings, J., Titov, I., Aziz, W., Marcheggiani, D., and Sima'an, K. (2017). Graph convolutional encoders for syntax-aware neural machine translation. Preprint at arXiv. https://doi.org/10.48550/arXiv.1704.04675.
30. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., and Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. Preprint at arXiv. https://doi.org/10.48550/arXiv.1806.01973.
31. Kipf, T.N., and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. Preprint at arXiv. https://doi.org/10.48550/arXiv.1609.02907.
32. Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. Preprint at arXiv. https://doi.org/10.48550/arXiv.1312.6203.
33. Hu, F., Zhu, Y., Wu, S., Wang, L., and Tan, T. (2019). Hierarchical graph convolutional networks for semi-supervised node

classification. Preprint at arXiv. https://doi.org/10.48550/arXiv.1902.06667.

34. Yun, S., Cong, J., Wendong, F., Zelin, D., and Xinke, B. (2022). TARGCN: Temporal Attention Recurrent Graph Convolutional Neural Network for Traffic Prediction. https://github.com/csust-sonie/TARGCN.
35. Dempster, A., Schmidt, D.F., and Webb, G.I. (2021). MiniRocket: A Very Fast (Almost) Deterministic Transform for Time Series Classification. In KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (ACM). https://doi.org/10.1145/3447548.3467231.
36. Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.X., and Yan, X. (2019). Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. Preprint at arXiv. https://doi.org/10.48550/arXiv.1907.00235.
37. Bai, S., Kolter, J.Z., and Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. Preprint at arXiv. https://doi.org/10.48550/arXiv.1803.01271.
38. Ai, Q., Ding, B., Liu, Q., and Meng, W. (2016). A subject-specific EMG-driven musculoskeletal model for applications in lower-limb rehabilitation robotics. Int. J. Human. Robot. *13*, 1650005. https://doi.org/10.1142/S0219843616500043.
39. Torricelli, D., Cortés, C., Lete, N., Bertelsen, Á., Gonzalez-Vargas, J.E., Del-Ama, A.J., Dimbwadyo, I., Moreno, J.C., Florez, J., and Pons, J.L. (2018). A subject-specific kinematic model to predict human motion in exoskeleton-assisted gait. Front. Neurorobot. *12*, 18. https://doi.org/10.3389/fnbot.2018.00018.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Gait Data	This paper	N/A
Software and algorithms		
Python 3.8; RRID: SCR_008394	This paper	https://www.python.org/getit/
Pytorch 1.8.1; RRID: SCR_018536	This paper	N/A
Sklearn 0.24.2; RRID: SCR_019053	This paper	N/A
Numpy 1.20.0; RRID: SCR_008633	This paper	N/A
Pandas 1.2.5; RRID: SCR_018214	This paper	N/A
Matplotlib 3.4.2; RRID: SCR_008624	This paper	N/A
Keras 2.4.3	This paper	N/A
S-ST-GCN	Shi J et al.[20]	https://doi.org/10.3390/s21010205
MiniRcket	Dempster A et al.[35]	https://doi.org/10.1145/3447548.3467231
TST	Li S et al.[36]	https://doi.org/10.48550/arXiv.1907.00235
TCN	Shaojie B et al.	https://doi.org/10.48550/arXiv.1907.00235
LSTM	Hochreiter, S et al.[14]	https://doi.org/10.1162/neco.1997.9.8.1735
RNN	Elman J L et al.[15]	https://doi.org/10.1207/s15516709cog1402_1
2s-ST-ST-GCN/ST-STGCN	This paper	N/A
Other		
Windows 10	N/A	N/A
CPU: AMD R7-5800H	N/A	N/A
Memory: 16GB	N/A	N/A
Graphs card: RTX3060, 6GB	N/A	N/A
ICM20948	N/A	N/A
STM32F103C8T6	N/A	N/A
STM32F767IGT6	N/A	N/A

RESOURCE AVAILABILITY

Lead contact

The Lead Contact of our manuscript is **Jianjun Yan** with his email address jjyan@ecust.edu.cn.

Materials availability

In our experiment, we used hardware resources mainly, such as the inertial measurement unit ICM20948, microcontroller STM32F103C8T6 and STM32F767IGT6, and a computer (detailed environment configuration in Table 2).

Data and code availability

- The data that support the findings of this study are available from the corresponding author upon reasonable request. The remaining hardware and software resources are listed in the table of key resources table.
- All original code has been listed in the key resources table with DOIs and is publicly available from the lead contact upon reasonable request.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon reasonable request.

EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS

Experiment of skeleton construction

This experiment was mainly used to select and compare the effects of different skeleton construction methods on model performance. In this experiment, we change the number of nodes and the way of connecting the nodes through the human forward kinematics solver module to obtain different skeleton models, and conduct the experiment when other algorithmic parameters remain unchanged, and the result proves that the 8-node skeleton model is more beneficial for the expression of gait.

Experiment of shift size

This experiment was used to compare the effect of different shift size on model performance when adding windows to process data. Applying the same external conditions, the data are processed by varying the shift size and then used separately as inputs to the model for experimental comparisons, and the result proves that the model performs best when the shift size is 10.

Experiment of window size

This experiment was used to compare the effect of different window size on model performance when adding windows to process data. Applying the same external conditions, the data are processed by varying the window size and then used separately as inputs to the model for experimental comparisons, and the result proves that the model performs best when the window size is 10.

User-independent experiment

This experiment uses the algorithm proposed in this paper to experimentally compare with other algorithms to demonstrate the superiority of 2s-ST-STGCN. In the experiment, other external conditions are kept the same (e.g., shift size of 10, window size of 80), the established human skeleton (including joints, bones and motion data) is used as the input of 2s-ST-STGCN, and the original three-axis acceleration and three-axis angular velocity of the human body are used as the inputs of the other algorithms, and both of them are trained for 500 rounds with the learning rate of 0.0001. The experimental results prove that 2s-ST-STGCN achieves the best results.

User-dependent experiment

This experiment uses the algorithm proposed in this paper to experimentally compare with other algorithms to demonstrate the superiority of 2s-ST-STGCN. Unlike the user-independent experiments, the dataset of this experiment does not guarantee a one-to-one correspondence between the dataset and the person, but rather includes the data of someone in both the training and validation sets. In the experiment, other external conditions are kept the same (e.g., shift size of 10, window size of 80), the established human skeleton (including joints, bones and motion data) is used as the input of 2s-ST-STGCN, and the original three-axis acceleration and three-axis angular velocity of the human body are used as the inputs of the other algorithms, and both of them are trained for 500 rounds with the learning rate of 0.0001. The experimental results prove that 2s-ST-STGCN achieves the best results.

Ablation experiment

This experiment is to verify that the improvements to the algorithm in this paper are effective. The algorithm with the removal of the temporal attention module (S-STGCN-Y), the algorithm with the removal of the human positivism module (ST-STGCN-N), the algorithm with the removal of the two-stream structure (ST-STGCN), and the algorithm with all modules included (2s-ST-STGCN) were experimented with the same external conditions, respectively. The experimental results prove that all the modules added have positive gain on the model.

METHOD DETAILS

The algorithm proposed in this paper consists of the following main parts.

Positive kinematics solution for the human body

This part is mainly through the human body movement angle data and human joint length calculation to get the human skeleton model, the specific formula and calculation method can be found in section positive kinematics solution for the human body.

Temporal convolution layer with temporal attention

The temporal attention part is added within the original temporal convolutional layer, so that a temporal convolutional part, a temporal attention part and a gating mechanism constitute the temporal convolutional layer with temporal attention. The basic mechanism of the temporal attention part can be represented by the following equation:

$$E = V_e \cdot \sigma\left(\left(\left(\chi_h^{(r-1)}\right)^T U_1\right) U_2 \left(U_3 \chi_h^{(r-1)}\right) + b_e\right)$$

That is, the inputs of T time steps are transposed into vector multiplications in the time dimension and the correlation degree between different times is calculated. This allows considering time frames within the gait cycle that are more relevant to the gait.

Two-stream structure

The joint data representing the joint position and the bone data representing the length and orientation of the bones, as well as the motion data of both, are input into the two ST-STGCN, and the neighbor matrix A is learned separately, and the edge_importance (a matrix of weights of edges, which is used to give a larger weight to important edges in the neighbor matrix and suppresses the weights of non-important edges) is learned, i.e., each data stream is trained separately, and then the output tensor of the two data streams is fused to predict the gait labels.

So the whole ST-STGCN structure consisted of three parts: an input layer of BatchNorm; six basic graph convolution blocks; and a Fully Convolutional Network (FCN). The basic graph convolution block mainly contains two kinds of modules, S-SGC and T-Conv. S-SGC is a spatial graph convolution module, which contains spatial attention mechanism; T-Conv is a temporal graph convolution module, which contains temporal attention mechanism. The two-stream structure consists of two ST-STGCNs, which form a 2s-ST-STGCN by processing different data separately and fusing the outputs at the end.

QUANTIFICATION AND STATISTICAL ANALYSIS

We mainly use the confusion matrix provided by python to compute the metrics so as to make a comparison between the algorithms.