Robotica (2024), **42**, pp. 367–388
doi:10.1017/S0263574723001492

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Interpretable long-term gait trajectory prediction based on Interpretable-Concatenation former

Jie Yin[1], Meng Chen[2], Chongfeng Zhang[3], Tao Xue[1], Ming Zhang[4] and Tao Zhang[1]

[1]Department of Automation, Tsinghua University, Beijing, P.R. China, [2]Aerospace System Engineering Shanghai, ASES, Shanghai, P.R. China, [3]Shanghai Academy of Spaceflight Technology, SAST, Shanghai, P.R. China, and [4]College of Engineering and Physical Sciences, Aston University, Birmingham, UK

Corresponding author: Tao Zhang; Email: taozhang@mail.tsinghua.edu.cn

Received: 5 June 2023; **Revised:** 8 October 2023; **Accepted:** 12 October 2023; **First published online:** 20 November 2023

Keywords: Human biomechanics; exoskeletons; gait trajectory prediction; long sequence time-series forecasting; interpretability

Abstract

Human gait trajectory prediction is a long-standing research topic in human–machine interaction. However, there are two shortcomings in the current gait trajectory prediction technology. The first shortcoming is that the neural network model of gait prediction only predicts dozens of future time frames of gait trajectory. The second shortcoming is that the gait prediction neural network model is uninterpretable. We propose the Interpretable-Concatenation former (IC-former) model, which can predict long-term gait trajectories and explain the prediction results by quantifying the importance of data at different positions in the input sequence. Experiments prove that the IC-former model we proposed not only makes a breakthrough in prediction accuracy but also successfully explains the data basis of the prediction.

1. Introduction

Gait trajectory prediction is a long-term research topic in human–machine interaction. Formally, gait trajectory prediction predicts the lower limb movement data based on certain types of sensor information collected currently (the movement data can be spatial position, angle, speed, and acceleration). It is an actual application of time-series forecasting techniques. Good gait prediction results can provide a necessary basis for the fields of lower limb exoskeleton control and human behavior prediction.

At time t, $X^t = \left\{x_1^t, \ldots, x_{L_x}^t | x_i^t \in R^{d_x}\right\}$ represents the input sensor data, $Y^t = \left\{y_1^t, \ldots, y_{L_y}^t | y_i^t \in R^{d_y}\right\}$ represents the output forecast data, where L_y and L_x are the lengths of the predicted sequence and the input sequence, respectively, and d_y and d_x are the dimensions of the predicted sequence and the input sequence, respectively.

At present, the deep neural network is the primary method used to predict the gait trajectory, but there are three primary deficiencies in the current research:

1. At present, most of the neural network models used for gait prediction only predict the gait trajectory of dozens or even only a few time frames in the future. A too-short prediction window will result in a significant time interval of the predicted trajectory or a short prediction time, which limits the application of the prediction results. For example, short-term gait prediction cannot effectively assist the control of the lower extremity exoskeleton because of motor output delay.
2. Currently, almost no neural network models with high accuracy used for gait prediction are interpretable. Although these models can somewhat predict the future gait trajectory, they are black

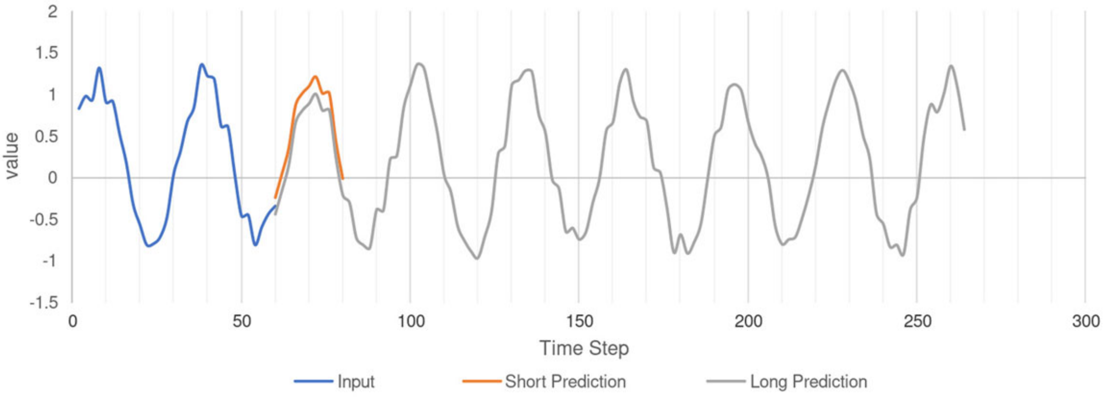


Figure 1. *Long-term series forecasting.*

boxes. The non-interpretable neural network model used for gait prediction cannot provide a further reference for gait research.

Long-term series forecasting refers to time-series forecasting with an enormous value of L_y. There is no clear threshold to define long-term series forecasting, but the value of L_y generally needs to be in the hundreds (Fig. 1).

Based on the above insights, we propose the Interpretable-Concatenation former (IC-former) model, which can interpretably predict long-term gait trajectories by quantifying the importance of data from different segments of the input sequence. The model is built according to the encoder–decoder structure using the dot-product attention mechanism. The IC-former model generatively outputs long-term forecast sequences, which avoids errors caused by cyclic calculations like the RNN model and speeds up calculations. The interpretive network can help people understand the model's logic, formulate control strategies, and assist in medical diagnosis. The framework structure of the IC-former model is shown in Fig. 2. The main contributions of this paper are as follows:

1. The proposed IC-former model can accurately predict long-term gait trajectories. It can generate long-term prediction sequences based on one-time input, reducing the cumulative calculation error. The model provides a good reference for formulating lower extremity exoskeleton control strategies and predicting human behavior.
2. Unlike previous black boxes, the proposed IC-former is an interpretive network relying on the dot-product attention mechanism, which can locate the critical parts of the gait trajectory.
3. The prediction accuracy of the IC-former is higher than that of a series of time-series prediction models widely used in gait trajectory prediction. In the case of the same length of the input sequence, the scale of the IC-former is smaller than that of the Informer model, which is more suitable for application on mobile computing platforms.

2. Related work

2.1. Gait trajectory prediction

Traditional gait prediction methods represented by controlled oscillators can deal with some simple fixed-period gait prediction problems. Adaptive oscillators can predict cyclic and quasi-cyclic gait sequences., but they cannot predict rapidly changing gaits in time. The traditional gait prediction method has a noticeable delay in predicting rapidly changing gait. This shortcoming limits the application of the traditional method [1, 2].

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

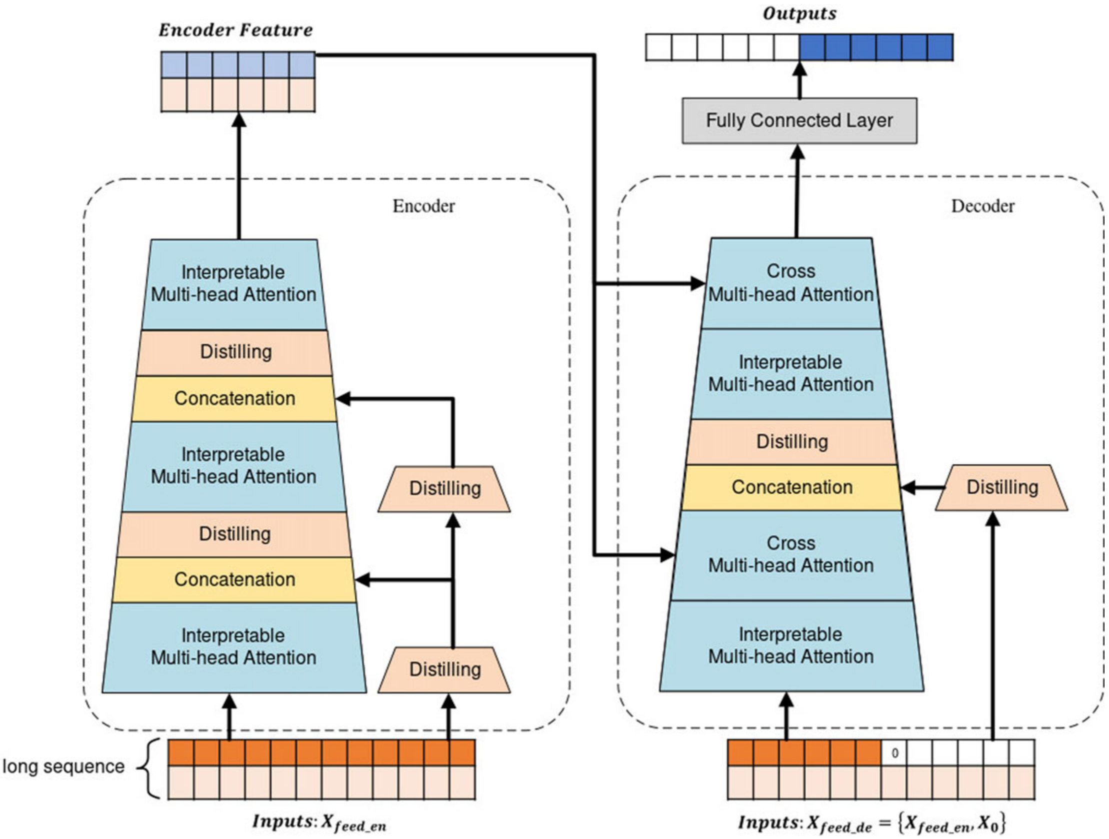

Encoder Feature
Outputs
Fully Connected Layer
Encoder
Decoder
Interpretable Multi-head Attention
Distilling
Concatenation
Interpretable Multi-head Attention
Distilling
Concatenation
Interpretable Multi-head Attention
Distilling
Distilling
Cross Multi-head Attention
Interpretable Multi-head Attention
Distilling
Concatenation
Cross Multi-head Attention
Interpretable Multi-head Attention
Distilling
long sequence
0
Inputs: X_{feed_en}
Inputs: $X_{feed\_de} = \{X_{feed\_en}, X_0\}$

Figure 2. *Interpretable Multi-head Attention is an interpretable dot-product attention layer proposed by us, whose attention value can measure the importance of the input. The distilling layers are 1D convolutional layers, halving the data length and efficiently extracting context-related features. The Concatenation operation merges the data features of parallel channels to add features with explicit meaning to the main channel. The encoder receives a long sequence and then feeds it into two parallel channels. The main channel contains Interpretable Multi-head Attention, through which the IC-former model extracts the main feature. The other channel is the auxiliary channel, which has distilling layers that preserve the mathematical meaning of the input. There are two differences between the decoder and the encoder. The first difference is that the decoder receives a long sequence of inputs, and pads target elements with zeros. Another difference is that the decoder contains Cross Multi-head Attention to combine the extracted features from the encoder and decoder. Finally, the decoder immediately predicts output elements in a generative style.*

The data-based deep learning method can directly output the trajectory in the future based on the sensor data without an iterative update of parameters, which solves the delay problem. Tanghe et al. and Kang et al. predicted the gait phase [3, 4] based on the deep neural network, but their models had high requirements for the type and quantity of sensors, and the workload of processing data was heavy. High requirements limit the real-world application of their method. It is significant to reduce the type and quantity of sensors as much as possible to reduce the cost of gait prediction.

As an important method for processing sequence information, the long short-term memory (LSTM) model is widely used to predict gait trajectory [5, 6, 7, 8, 9]. These models can only predict gait trajectories within a limited time frame. When predicting long-term gait trajectories, the LSTM model they adopted is slow and has a large cumulative error caused by the loop structure of the LSTM model itself.

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

Karakish et al. and Kang et al. used a CNN-based deep neural network to predict gait [10, 11], which is faster than the LSTM model. However, the model proposed by Karakish et al. can only predict the gait trajectory in a short time, and the model proposed by Kang et al. relies on user-specific basis data.

The gait trajectory prediction models mentioned above are difficult to predict long-term gait trajectory, and none of them consider the issue of model interpretability. They cannot judge the input's importance or explain the model's data basis. The above models only predict gait as a black box and cannot support in-depth gait research.

2.2. Long-term series forecasting

Currently, in the time-series forecasting field, one challenging work is improving the performance of long sequence time-series forecasting (LSTF). Predicting long-period gait is more difficult than predicting short-period gait. It can explore the deep-level characteristics of gait data and help people design more optimal control strategies.

Before the Transformer structure was developed, recurrent neural network (RNN) models dominated time-series forecasting [12, 13, 14]. Hochreiter et al. propose a LSTM model for solving the gradient vanishing issues [15]. Although the LSTM model is excellent, the low computational efficiency brought about by its cyclic structure and the difficulty of feature extraction when dealing with long sequences limit its application.

Since the Transformer model was proposed in the field of natural language processing [16], its ability to process sequence data has been applied to predict time series [17, 18]. Although the Transformer model has brought many breakthroughs, it is expensive to use the self-attention mechanism directly in LSTF because of its L-quadratic memory and computation consumption on L-length inputs/outputs. Many improvements are proposed to solve this problem [19, 20]. Li et al. present LogSparse Transformer which the memory cost of the self-attention mechanism is only $O\left(L * (log(L))^2\right)$ [21]. Beltagy et al. introduce an attention mechanism called Longformer that scales linearly with sequence length, reducing its complexity to $O\left(L * log(L)\right)$ [22]. A representative example of all relevant improvements is the Informer model proposed by Zhou et al., which predicts the output more quickly and accurately with less memory and computation consumption than previous methods [23]. The Informer model has one essential breakthrough: the ProbSparse self-attention mechanism.

The above time-series forecasting models significantly reduced the model size and improved forecasting accuracy. However, none is interpretable. Prediction accuracy also needs to be improved.

2.3. Model interpretability

The current methods of interpreting neural networks are mainly divided into gradient-based methods and methods based on adjusting input values [24]. The attention mechanism of a neural network multiplies the input with nonzero weights that sum to 1. The attention weight of a variable perceptually represents the importance of the variable. Research on the interpretation of neural networks proves that the value of the attention of the Transformer structure is indeed positively correlated with the importance of variables [25, 26, 27]. Many interpretable neural network models for time-series forecasting have been proposed. Oreshkin et al. proposed an interpretable time-series forecasting model named N-BEATS, but it can only predict the time series of a dozen time frames in the future, which cannot meet the needs of long-term series forecasting [28]. Lim et al. proposed Temporal Fusion Transformers, a time-series prediction model that combines LSTM and Transformer structures. The LSTM framework it uses limits its application to predicting long-term series [29]. In summary, studying interpretable neural network models for long-term series forecasting is meaningful.

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

3. Interpretable-Concatenation former

The Transformer model based on the dot-product attention mechanism can effectively extract the data features of sequence information. However, to improve efficiency and accuracy, the Transformer model breaks the mathematical meaning of the input variables. The breaking mathematical meaning makes the Transformer model an uninterpretable black box. A critical issue is how to preserve the mathematical meaning of the input variables with high accuracy and efficiency. Studying this problem is of great significance to studying the interpretability of neural network models, especially Transformer models.

Driven by the above ideas, we propose Interpretable Multi-head Attention, which adjusts the traditional Transformer structure to retain the clear mathematical meaning of the input and ensures that only the weight-adjusted input determines the output. Therefore, the weight calculated by Interpretable Multi-head Attention can measure the importance of the mathematical meaning represented by different input fragments. Further, we propose the IC-former model based on Interpretable Multi-head Attention. IC-former is a deep neural network model of the encoder–decoder structure. IC-former can measure the importance of inputs of different lengths at different locations on both local and global scales with high predicting accuracy. The IC-former model generatively outputs long-term forecast sequences, which makes the model calculation fast and eliminates cumulative errors. The IC-former model is shown in Fig. 3.

3.1. Attention mechanisms

Ashish et al. propose the Transformer based solely on attention mechanisms. The Transformer uses stacked self-attention and point-wise, fully connected layers for both the encoder and decoder to form an encoder–decoder structure.

The core of the attention mechanism is the scaled dot-product attention. Calculating the scaled dot-product attention requires three parameters: queries (Q), keys (K), and values (V), where $Q \in R^{L_Q * d}$, $K \in R^{L_K * d}$, $V \in R^{L_V * d}$, and d is the input dimension. The formula is Eq. (1):

$$Attention\ (Q, K, V) = softmax\left(\frac{Q * K^T}{\sqrt{d}}\right) * V \tag{1}$$

The Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers and outperforms them for translation tasks. The attention mechanism effectively splits the input space, emphasizing only elements relevant to the task. Transformer structure is widely used in text-based and chart-based tasks because of its robust feature extraction ability to sequences.

3.2. Informer

Dot-product attention mechanisms require calculating each dot-product pair, which makes the computational cost of the attention mechanism increase squarely with the linear increase of the sequence length $\left(L_Q, L_K\right)$. This property makes long-term series forecasting expensive in computation. To solve this problem, Haoyi et al. propose ProbSparse Self-attention and apply it to the Informer model. They propose the Query Sparsity Measurement to find the most critical vectors of queries (Q). The formula is Eq. (2), where L_K represents the length of K, d represents the model dimension, and q_i, k_i, and v_i are the ith row in Q, K, and V, respectively:

$$M\ (q_i, K) = ln\sum_{j=1}^{L_K} e^{\frac{q_i * k_j{}^T}{\sqrt{d}}} - \frac{1}{L_K}\sum_{j=1}^{L_K} \frac{q_i * k_j{}^T}{\sqrt{d}} \tag{2}$$

Query Sparsity Measurement is Kullback–Leibler divergence between q_i and K. The larger the Query Sparsity Measurement is, the more different q_i and K are, and the more critical q_i is. Based on the Query Sparsity Measurement, the ProbSparse Self-attention allows each K only to attend to the u-dominant q_i. The formula is Eq. (3):

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

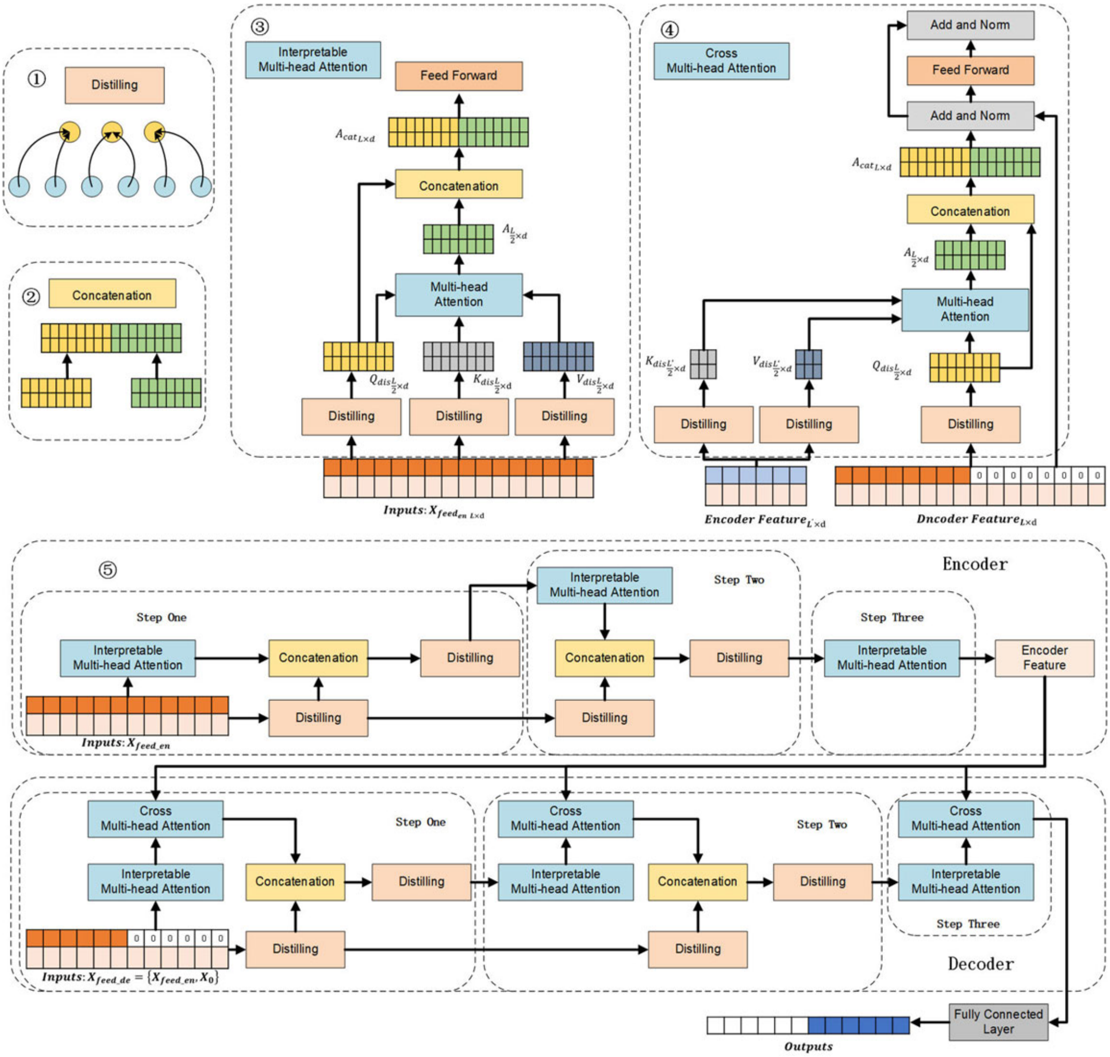

①
Distilling
②
Concatenation
③
Interpretable Multi-head Attention
Feed Forward
$A_{cat_{L\times d}}$
Concatenation
$A_{\frac{L}{2}\times d}$
Multi-head Attention
$Q_{dis_{\frac{L}{2}\times d}}$
$K_{dis_{\frac{L}{2}\times d}}$
$V_{dis_{\frac{L}{2}\times d}}$
Distilling
Distilling
Distilling
Inputs: $X_{feed_{en}\ L\times d}$
④
Cross Multi-head Attention
Add and Norm
Feed Forward
Add and Norm
$A_{cat_{L\times d}}$
Concatenation
$A_{\frac{L}{2}\times d}$
Multi-head Attention
$K_{dis_{\frac{L'}{2}\times d}}$
$V_{dis_{\frac{L'}{2}\times d}}$
$Q_{dis_{\frac{L}{2}\times d}}$
Distilling
Distilling
Distilling
Encoder $Feature_{L'\times d}$
Dncoder $Feature_{L\times d}$
⑤
Encoder
Step One
Interpretable Multi-head Attention
Concatenation
Distilling
Distilling
Inputs: X_{feed_en}
Step Two
Interpretable Multi-head Attention
Concatenation
Distilling
Distilling
Step Three
Interpretable Multi-head Attention
Encoder Feature
Decoder
Step One
Cross Multi-head Attention
Interpretable Multi-head Attention
Concatenation
Distilling
Distilling
Inputs: $X_{feed\_de} = \{X_{feed\_en}, X_0\}$
Step Two
Cross Multi-head Attention
Interpretable Multi-head Attention
Concatenation
Distilling
Distilling
Step Three
Cross Multi-head Attention
Interpretable Multi-head Attention
Fully Connected Layer
Outputs

Figure 3. *1: Distilling layer; 2: Concatenation operation; 3: Interpretable Multi-head Attention layer; 4: Cross Multi-head Attention layer; 5: Interpretable-Concatenation former.*

$$Attention\,(Q, K, V) = softmax\left(\frac{\overline{Q} * K^T}{\sqrt{d}}\right) * V \tag{3}$$

$\overline{Q}$ is a sparse matrix only containing the top-u vectors of Q measured by the Query Sparsity Measurement. $u = c \cdot ln\, L_Q$ where c is a constant sampling factor. Then, the ProbSparse Self-attention only calculates $O\left(ln\, L_Q\right)$ dot-product for each query-key pair, and the layer memory usage is reduced to $O\left(L_K\ ln\, L_Q\right)$. For self-attention $L_Q = L_K = L$, so the complexity and space complexity of the ProbSparse self-attention is $O(L\, ln\, L)$.

Informer adopts the self-attention distilling mechanism, adding a convolutional layer on the time dimension between the attention modules. A max-pooling layer with two strides is added to down-sample inputs into its half slice. The distilling mechanism reduces memory usage to $O\left((2 - \epsilon)L\, log\, L\right)$ while ensuring computational accuracy.

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

3.3. Distilling layer

The distilling layer is a 1D convolutional layer whose output sequence's length is half the length of the input sequence. The distilling layer reduces the sequence length, memory consumption, and computational complexity. It is also possible to further reduce the sequence length to a quarter or less, which needs to be determined according to the prediction accuracy and scale of the model. Finding the sequence segment with the most information is crucial for interpreting the gait trajectory prediction model. In this paper, we focus on methods to measure the importance of the temporal dimension. So we choose a one-dimensional convolutional network instead of a two-dimensional convolutional network:

$$C_{dis\,\frac{L}{2}\times d} = Conv1d_C\left(C_{L\times d}\right) \qquad C_{L\times d} \in \{Q, K, V\} \tag{4}$$

The distilling layer in the IC-former model has three important functions:

1. The distilling layers reduce the length of the input to half, significantly reducing the model size. Reducing model size is especially important when the input and predicted sequences are long.
2. The distilling layers effectively extract context-related features, which helps to improve the prediction accuracy of the model.
3. The distilling layers retain the mathematical meaning of the input. The convolutional network's local calculation characteristics make the model's calculation process easier for humans to understand.

3.4. Concatenation operation

The Concatenation operation is an operation that splices two pieces of data together in the time dimension. It is worth noting that there are Concatenation operations inside and outside the attention layer of the IC-former model. Concatenation operations play different roles inside and outside the attention layer:

$$A_{cat} = \mathbf{Concatenation}\left\{(Input_A, Input_B), \mathbf{dim} = \mathbf{time}\right\} \tag{5}$$

The function of the Concatenation operation inside the attention layer is to splice the process variable queries (Q) with the original attention. It enables queries (Q) to continue participating in subsequent calculations as primary features instead of being discarded directly, breaking the original barriers between data. Concatenation operation outside the attention layer splices the auxiliary and main channels in the time dimension to obtain the concatenated information. The attention calculated for the concatenated information represents the relative importance of each input segment.

3.5. Interpretable Multi-head Attention

Although the traditional Transformer structure has proved its powerful feature extraction ability, it destroys the mathematical meaning of the data. The Transformer model cannot measure the data's importance because of the following two reasons: 1: The queries (Q), keys (K), and values (V) in the traditional Transformer structure are obtained by the fully connected layer. But the fully connected layer directly destroys its mathematical meaning. 2: The traditional Transformer structure adds the output of dot-product attention with its original input at the element level, which keeps the original data features. However, it also causes the original data features to continue participating in the subsequent calculation process without being adjusted by the attention weight. The addition at the element level also confuses the mathematical meaning of the data.

In response to these two points, we made adjustments and proposed Interpretable Multi-head Attention. There are two main improvements of Interpretable Multi-head Attention:

1. Get queries (Q), keys (K), and values (V) through the distilling layer, which has three advantages. The characteristics of the local calculation of the convolutional network ensure that the

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

mathematical meaning of the obtained queries (Q), keys (K), and values (V) is clear and can be intuitively understood by humans. The convolutional network is suitable and efficient for processing sequence information. The data length of queries (Q), keys (K), and values (V) obtained by the distilling layer is half of the input, reducing the model's memory usage.

2. Interpretable Multi-head Attention no longer adds the original input to the output of the attention mechanism at the element level, which ensures that all the data features of the model are obtained based on the weight-adjusted data.

Each column of the attention weight matrix represents the importance of the data corresponding to each row of values (V). The weight matrix of the attention mechanism calculated by queries (Q) and keys (K) is represented by W, and the elements of each matrix are represented by lowercase letters. L is the length of the input sequence, and d is the model dimension:

$$W = Q * K = \begin{bmatrix} w_{1,1} & \cdots & w_{1,L} \\ \vdots & \ddots & \vdots \\ w_{L,1} & \cdots & w_{L,L} \end{bmatrix} \tag{6}$$

$$\begin{aligned} A = W * V &= \begin{bmatrix} w_{1,1} & \cdots & w_{1,L} \\ \vdots & \ddots & \vdots \\ w_{L,1} & \cdots & w_{L,L} \end{bmatrix} * \begin{bmatrix} v_{1,1} & \cdots & v_{1,d} \\ \vdots & \ddots & \vdots \\ v_{L,1} & \cdots & v_{L,d} \end{bmatrix} \\ &= \begin{bmatrix} w_{1,1} * v_{1,1} + \ldots + w_{1,L} * v_{L,1} & \cdots & w_{1,1} * v_{1,d} + \ldots + w_{1,L} * v_{L,d} \\ \vdots & \ddots & \vdots \\ w_{L,1} * v_{1,1} + \ldots + w_{L,L} * v_{L,1} & \cdots & w_{L,1} * v_{1,d} + \ldots + w_{L,L} * v_{L,d} \end{bmatrix} \end{aligned} \tag{7}$$

The multi-head attention mechanism divides the data's model dimension into multiple modules, which does not destroy the mathematical meaning of the 1D convolutional layer. The attention of the whole model dimension is the summing of all multi-head attention. Let H denote the total number of heads, and h represents the hth head:

$$W^h = Q^h * K^h = \begin{bmatrix} {w_{1,1}}^h & \cdots & {w_{1,L}}^h \\ \vdots & \ddots & \vdots \\ {w_{L,1}}^h & \cdots & {w_{L,L}}^h \end{bmatrix} \tag{8}$$

$$\begin{aligned} A^h = W^h * V^h &= \begin{bmatrix} {w_{1,1}}^h & \cdots & {w_{1,L}}^h \\ \vdots & \ddots & \vdots \\ {w_{L,1}}^h & \cdots & {w_{L,L}}^h \end{bmatrix} * \begin{bmatrix} {v_{1,1}}^h & \cdots & {v_{1,d/H}}^h \\ \vdots & \ddots & \vdots \\ {v_{L,1}}^h & \cdots & {v_{L,d/H}}^h \end{bmatrix} \\ &= \begin{bmatrix} {w_{1,1}}^h * {v_{1,1}}^h + \ldots + {w_{1,L}}^h * {v_{L,1}}^h & \cdots & {w_{1,1}}^h * {v_{1,d/H}}^h + \ldots + {w_{1,L}}^h * {v_{L,d/H}}^h \\ \vdots & \ddots & \vdots \\ {w_{L,1}}^h * {v_{1,1}}^h + \ldots + {w_{L,L}}^h * {v_{L,1}}^h & \cdots & {w_{L,1}}^h * {v_{1,d/H}}^h + \ldots + {w_{L,L}}^h * {v_{L,d/H}}^h \end{bmatrix} \end{aligned} \tag{9}$$

$$W = \sum_{h=1}^{h=H} W^h \tag{10}$$

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

To further reduce the computational complexity and memory consumption, we use the ProbSparse attention proposed in [23] to replace the traditional dot-product attention.

3.6. Cross Multi-head Attention

The basic structure of Cross Multi-head Attention is similar to that of Interpretable Multi-head Attention. They both use the distilling layer and Concatenation operations. There are only two differences between them. In the Cross Multi-Head Attention, keys (k) and values (v) are calculated by the encoder feature and queries (q) are calculated by the decoder feature. Cross Multi-head Attention retains the addition of the input and output of the attention mechanism at the element level. Cross Multi-head Attention effectively fuses the features of the encoder and decoder.

3.7. Encoder and decoder

The IC-former model is of an encoder–decoder structure. The encoder and decoder are obtained by repeatedly stacking the basic modules mentioned above according to the relationship shown in Fig. 3. The number of stacked modules can be flexibly adjusted based on data volume and model size.

We denote the length of the input sequence by L. The larger the value of L, the greater the consumption. This shortcoming is especially serious for dot-product attention because of L-quadratic memory and computation consumption on L-length inputs/outputs. To alleviate this problem, we use distilling layers in the attention layer and the encoder and decoder further to reduce the memory and computation consumption of the model.

The encoder and the decoder have two channels: main and auxiliary channels. The main channel measures the importance of the input sequence. The auxiliary channel provides the input with a precise mathematical meaning. Because the auxiliary channel only contains distilling layers, the data it processes retain a clear mathematical meaning. A Concatenation operation between the encoder and decoder concatenates the uninterpretable main channel and the interpretable auxiliary channel to obtain spliced features. The spliced features are input to the next module's Interpretable Multi-head Attention module, and each fragment's importance is measured. The attention of the interpretable auxiliary channel represents the relative richness of information in each sequence segment. The importance of the interpretable auxiliary channel features is also global, indicating the relative information abundance of each sequence segment in the model.

After the sequence passes through the stacked distilling layers in the auxiliary channel, the sequence length is shortened layer by layer. The granularity of the sequence continues to increase, which means that the length of the original input sequence represented by each vector increases accordingly. IC-former compares the relative importance of each fragment at different granularities. Measuring the local and global importance of sequence fragments reveals the data basis of the neural network model.

The structure of the decoder and the encoder is similar. The main difference is that the decoder has a Cross Multi-head Attention module, which fuses the features of the encoder and decoder. The input to the encoder is the entire input sequence. The input to the decoder is a combined sequence obtained by concatenating the input sequence and a zero sequence with the same length as the output sequence. The final decoder feature is input to a fully connected layer to obtain the final prediction output.

4. Experiment

4.1. Accuracy test

4.1.1. Datasets

We conduct experiments on datasets consistent with the Informer model to verify the prediction accuracy of our proposed IC-former model, in which the ETT dataset is collected by Zhou et al., and ECL and Weather are public benchmark datasets.

ETT (Electricity Transformer Temperature): ETT is a key indicator of the long-term deployment of electricity. The dataset records 2 years of data collected from two counties in China. To study the effect of granularity on the LSTF problem, the dataset contains 1-hour-level sub-datasets ETTh1 and ETTh2 and 15-minute-level sub-dataset ETTm1. Each data point has a target value, "oil temperature," and six power load characteristics. Training/validation/testing is 12/4/4 months.

ECL (Electricity Consuming Load): It collects electricity usage (Kwh) from 321 customers. Like Zhou et al., we transformed the dataset into hourly consumption for 2 years. Training/validation/testing is 15/3/4 months

Weather: The dataset contains local climate data for nearly 1,600 locations in the United States, with data points collected every hour for 4 years from 2010 to 2013. Each data point has a "wet bulb" target value and 11 climate features. Training/validation/testing is 28/10/10 months.

4.1.2. Experimental details

Baselines: Zhou et al. tested Informer [30], ARIMA [31], Prophet [32], LSTMa [33], LSTnet [34], and DeepAR [35] on the above-mentioned datasets. We cite Zhou et al.'s experimental results to compare with those obtained by our proposed IC-former on the above datasets.

Although our proposed model can effectively multivariate and univariate forecasting, we focus on univariate forecasting. We believe predicting gait trajectory should depend on as few sensors as possible. Fewer sensors mean fewer hardware requirements, which can significantly reduce the application threshold of algorithms. In the part of experiments, the IC-former model contains a three-layer encoder and a two-layer decoder. In the other part of the experiments, the IC-former model includes a two-layer encoder and a one-layer decoder. The IC-former model is optimized with Adam optimizer, and its learning rate starts from 0.0001, decaying two times smaller every two epochs, and the total epochs are 20. We set the batch size to 32.

We used two evaluation metrics, *MSE* and *MAE*, on each prediction window (averaging for multivariate prediction) and rolled the whole set with $stride = 1$:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y - \hat{y}\right)^2$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y - \hat{y}\right|$$

The prediction window sizes in ETTH, ECL, and Weather are $\{1d, 2d, 7d, 14d, 30d, 40d\}$. The prediction window size in ETTm is $\{6h, 12h, 24h, 72h, 168h\}$. Our computing platform (Nvidia 3090 24 GB GPU) is a little worse than the platform (Nvidia V100 32 GB GPU) used in the paper [30], but we still completed all prediction tasks, which shows that our model is adequate.

4.2. Gait trajectory prediction

4.2.1. Datasets

We fix the sensor on the outer thigh of the tester with a strap. The sensor integrates an inertial measurement unit to measure three-axis acceleration and angular velocity. After complementary filtering, the participant's leg posture is obtained, and the data are transferred to the computer. The locomotion mode recognition experiments are conducted with seven healthy subjects (one female, six males; 25.2 ± 2 years old; 172.0 ± 5 cm height; 70.10 ± 20 kg weight). All the participants gave their consent before taking part in the study. The experimental protocol is approved by the Institute Review Board of Tsinghua University (No. 20220222).

We collected two sets of data from seven participants named BaseData and GeneralizationData. BaseData is the thigh flexion angle sequences recorded by six participants who performed slow walking, fast walking, sitting down, and standing up. GeneralizationData is the thigh flexion angle sequence of the seventh experimenter who performed going upstairs and going downstairs. We collect BaseData

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

as the train data and GeneralizationData as the test data to test the generalization ability of our model. All data are min-max scaled to be between 0 and 1 for better results. Normalization makes the model converge faster.

To obtain a well-trained model, we performed data augmentation on the BaseData. Specifically, we perform cubic fitting on the trajectory curves in the BaseData to obtain trajectory functions. The obtained trajectory functions are sparsely and densely sampled to obtain variable frequency trajectory curves to supplement the original BaseData.

4.2.2. Hyperparameter tuning

In the experiment of gait trajectory prediction, the IC-former model contains a two-layer encoder and a one-layer decoder. The IC-former model is optimized with Adam optimizer, and its learning rate is 0.001. The batch size is 200. An aggressive initialization strategy is adopted for the model used for gait trajectory prediction. To test the accuracy of the IC-former model in predicting long-term gait trajectories, we use the gait trajectory of 512 time frames as the input to predict the gait trajectory of 512 time frames in the future. The rest of the model parameter settings are the same as the prediction accuracy experiments.

5. Results and analysis

5.1. Accuracy

We summarize the experimental results on the four datasets in Table I. The best results in each test are marked in bold. IC-former* represents models with classical attention mechanisms corresponding to IC-former. Except for the results of the IC-former model and the IC-former* model, the metrics are directly quoted from the paper [30].

In this experiment, each model predicts time series from univariate input data. From the experimental results, we conclude the following points: IC-former gets the best results by 36. IC-former gets better results from a global perspective; Compared with Informer, the total MSE error of IC-former is reduced by 34.7%, and the total MAE error is reduced by 23.9%. Compared with Informer, the single-experiment MSE error of IC-former is reduced by at most 64.2%, and the single-experiment MAE error is reduced by at most 44.5% (at TEEH1-720).

This experiment proves that the accuracy of our proposed IC-former model in predicting a single variable is better than that of the classical sequence prediction algorithms involved in the comparison. The high accuracy provides a basis for our research on explaining neural network models.

5.2. Attention mechanism

To explore the impact of the classic attention mechanism and ProbSparse self-attention on the IC-former model, we replaced the ProbSparse self-attention in IC-former with the classic attention mechanism. IC-former* represents models with classical attention mechanisms corresponding to IC-former. It can be seen from the experimental results that although the computational cost of the classical attention mechanism is high, the accuracy improvement brought by it is relatively limited, and even most of the results are not as good as ProbSparse self-attention.

The results reflect the high efficiency of ProbSparse self-attention and prove that our proposed IC-former can effectively cooperate with various attention mechanisms, especially ProbSparse self-attention, to achieve good results.

5.3. Gait trajectory prediction

Take the data-augmented BaseData as the training set to train our proposed IC-former model. Then use GeneralizationData as the test set to test the trained model. There are four completely different gait types

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

Table I. *Univariate long sequence time-series forecasting results on four datasets. Each column is the error for different prediction lengths for each dataset. The minimum error is marked in bold, and the rightmost column is the number of times each model performed best. Except for the results of IC-former and IC-former*, the error is directly quoted from the paper [30].*

		ETTH1					ETTM1					Weather					ECL					
Methods	**Metric**	**24**	**48**	**168**	**336**	**720**	**24**	**48**	**96**	**288**	**672**	**24**	**48**	**168**	**336**	**720**	**48**	**168**	**336**	**720**	**960**	**count**
IC-former	MSE	**0.059**	0.100	**0.100**	**0.115**	**0.096**	**0.014**	0.037	0.093	**0.196**	**0.360**	**0.087**	**0.142**	0.241	0.239	**0.229**	0.206	0.310	**0.336**	0.380	0.350	23
	MAE	**0.189**	0.248	**0.248**	**0.269**	**0.241**	**0.088**	0.144	0.242	**0.369**	**0.525**	**0.203**	**0.268**	0.344	**0.362**	**0.371**	0.323	0.399	**0.416**	0.449	0.440	
IC-former*	MSE	0.061	**0.086**	0.111	0.121	0.103	0.014	**0.025**	**0.076**	0.237	0.377	0.096	0.158	**0.216**	**0.233**	0.242	0.208	0.309	0.356	**0.332**	**0.334**	13
	MAE	0.196	**0.226**	0.265	0.273	0.250	0.091	**0.118**	**0.213**	0.414	0.536	0.218	0.282	**0.337**	0.364	0.376	0.327	0.401	0.432	**0.424**	**0.426**	
Informer	MSE	0.098	0.158	0.183	0.222	0.269	0.030	0.069	0.194	0.401	0.512	0.117	0.178	0.266	0.297	0.359	0.239	0.447	0.489	0.540	0.582	0
	MAE	0.247	0.319	0.346	0.387	0.435	0.137	0.203	0.372	0.554	0.644	0.251	0.318	0.398	0.416	0.466	0.359	0.503	0.528	0.571	0.608	
LogTrans	MSE	0.059	0.111	0.155	0.196	0.217	0.061	0.156	0.229	0.362	0.450	0.120	0.182	0.267	0.299	0.274	0.360	0.410	0.482	0.522	0.546	0
	MAE	0.191	0.263	0.309	0.370	0.379	0.192	0.322	0.397	0.512	0.582	0.247	0.312	0.387	0.416	0.387	0.455	0.481	0.521	0.551	0.563	
Reformer	MSE	0.172	0.228	1.460	1.728	1.948	0.055	0.229	0.854	0.962	1.605	0.197	0.268	0.590	1.692	1.887	0.917	1.635	3.448	4.745	6.841	0
	MAE	0.319	0.395	1.089	0.978	1.226	0.170	0.340	0.675	1.107	1.312	0.329	0.381	0.552	0.945	1.352	0.840	1.515	2.088	3.913	4.913	
LSTMa	MSE	0.094	0.175	0.210	0.556	0.635	0.099	0.289	0.255	0.480	0.988	0.107	0.166	0.305	0.404	0.784	0.475	0.703	1.186	1.473	1.493	0
	MAE	0.232	0.322	0.352	0.644	0.704	0.201	0.971	0.370	0.528	0.805	0.222	0.298	0.404	0.476	0.709	0.509	0.617	0.854	0.910	0.926	
DeepAR	MSE	0.089	0.126	0.213	0.403	0.614	0.075	0.197	0.336	0.908	2.371	0.108	0.177	0.259	0.535	0.407	**0.188**	**0.295**	0.388	0.471	0.583	4
	MAE	0.242	0.291	0.382	0.496	0.643	0.205	0.332	0.450	0.739	1.256	0.242	0.313	0.397	0.580	0.506	**0.317**	**0.398**	0.471	0.507	0.583	
ARIMA	MSE	0.086	0.133	0.364	0.428	0.613	0.074	0.157	0.242	0.424	0.565	0.199	0.247	0.471	0.678	0.996	0.861	1.014	1.102	1.213	1.322	0
	MAE	0.190	0.242	0.456	0.537	0.684	0.168	0.274	0.357	0.500	0.605	0.321	0.375	0.541	0.666	0.853	0.726	0.797	0.834	0.883	0.908	
Prophet	MSE	0.093	0.150	1.194	1.509	2.685	0.102	0.117	0.146	0.414	2.671	0.280	0.421	2.409	1.931	3.759	0.506	2.711	2.220	4.201	6.827	0
	MAE	0.241	0.300	0.721	1.766	3.155	0.256	0.273	0.304	0.482	1.112	0.403	0.492	1.092	2.406	1.030	0.557	1.239	3.029	1.363	4.184	

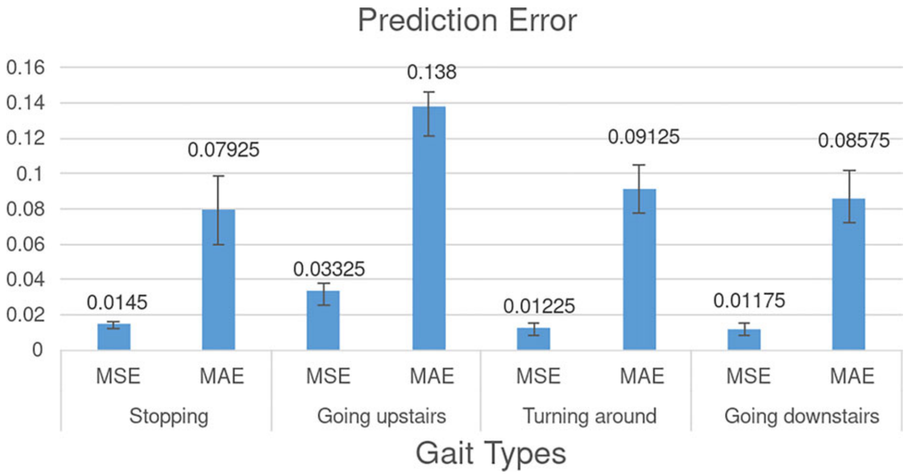


Figure 4. *Gait trajectory prediction error. The value of the y-axis in the picture represents the value of the normalized gait angle.*

in GeneralizationData: stopping, going upstairs, turning around, and going downstairs. We performed tests for each of the four gaits separately.

The experimental results are shown in Fig. 4. It is worth mentioning that the four predicted gaits are not involved in the training set at all, and all of them are real data. For different gaits, the prediction accuracy of the IC-former model is satisfactory. We plotted the prediction curves for all four gait types, as shown in Fig. 5. The left curve is the input data, and the right is the predicted and real data. The prediction of the IC-former model is not only satisfactory in terms of error but also accurately predicts critical quantities as peaks and valleys of the gait trajectory for a long period.

We trained the Informer model under the same parameters initialization method, learning rate, and other conditions as the IC-former model. The Informer model failed to predict gait trajectories in 13 randomly initialized experiments. Some prediction results of the Informer model are shown in Fig. 6. The IC-former model was trained successfully in 4 out of 13 randomly initialized experiments. The IC-former model is more accurate than the Informer model and is easier to train with stronger generalization ability.

5.4. Interpretability

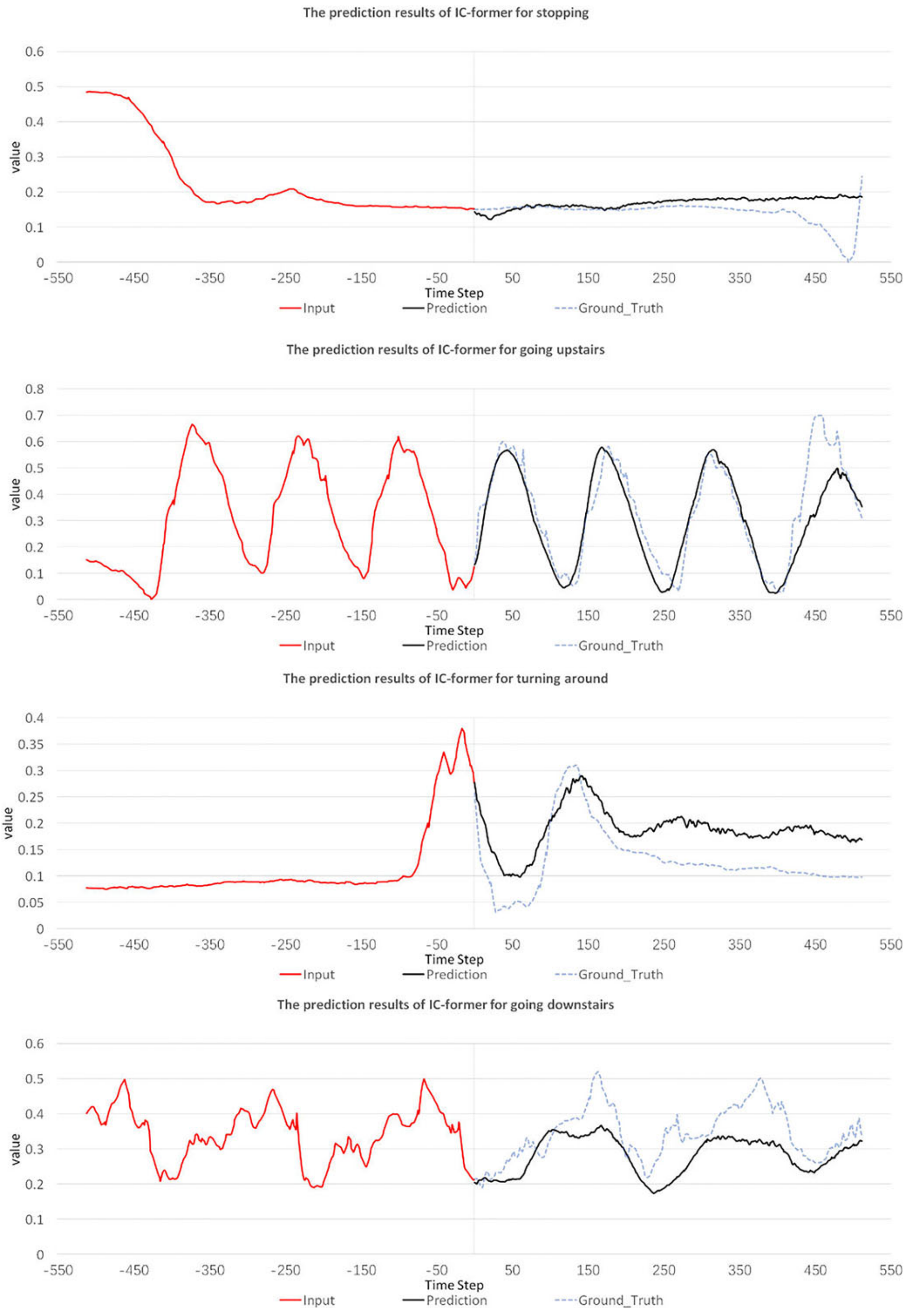
In the experiment of gait trajectory prediction, the IC-former model contains a two-layer encoder and a one-layer decoder. There are three Interpretable Multi-head Attention layers. Each Interpretable Multi-head Attention layer has eight head attentions. We analyze the attention of the Interpretable Multi-head Attention layer of the IC-former model when predicting the turning gait trajectory to interpret the model.

The first is the Interpretable Multi-head Attention layer located in the first layer of the encoder. In this experiment, the encoder input of the IC-former model is a 512-length gait trajectory sequence. After a distilling layer, the data length is 256. Therefore, the size of the attention matrix of the Interpretable Multi-head Attention layer located in the first layer of the encoder is 256*256. Each number represents a segment of the original sequence of length 2.

We show the eight-head attention in Fig. 7. The distributions of the eight-head attention are quite different because different head attention extracts different features. The multi-head attention mechanism ensures the diversification of attention and feature, enhancing the robustness.

As shown in Fig. 8, we sum all head attention to getting total attention. When predicting the turn gait trajectory, the attention of the first layer of the encoder is focused on the end of the sequence, which is consistent with the characteristics of the turn gait trajectory.

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press



The prediction results of IC-former for stopping

The prediction results of IC-former for going upstairs

The prediction results of IC-former for turning around

The prediction results of IC-former for going downstairs

Figure 5. *Predicted trajectories of the IC-former model for different gait types. The value of the y-axis in the picture represents the value of the normalized gait angle.*

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

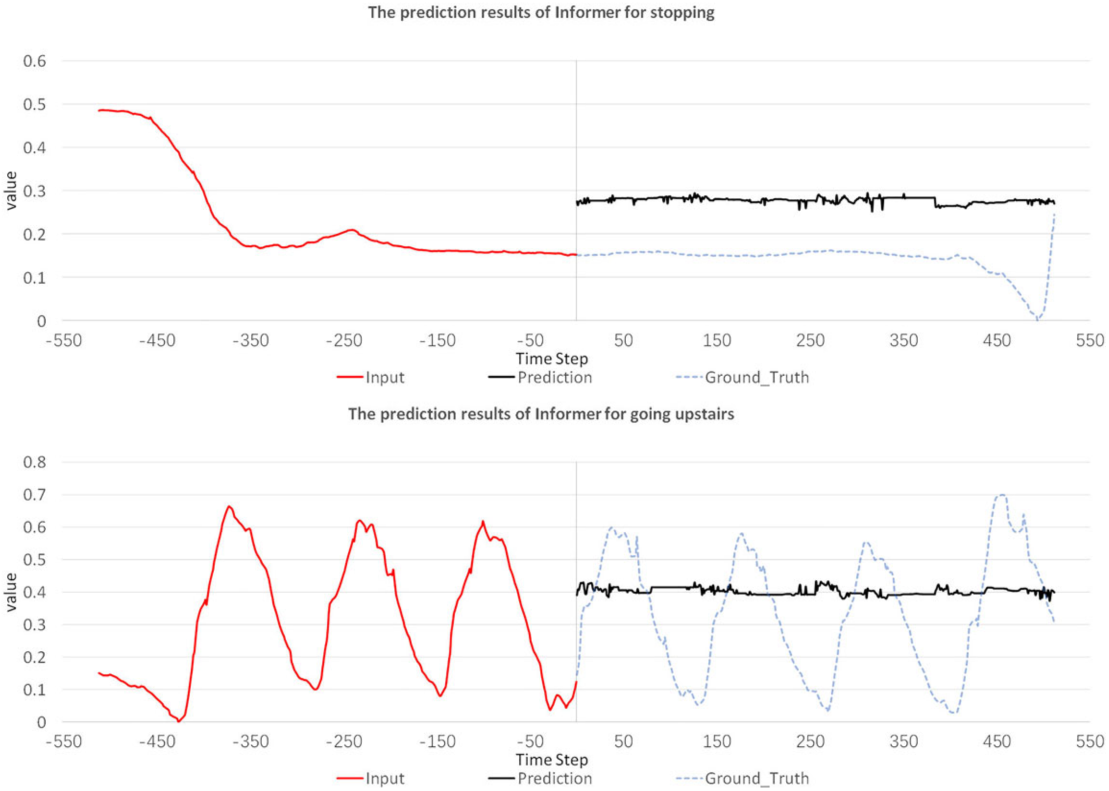


Figure 6. *Predicted trajectories of the Informer model for different gait types. The value of the y-axis in the picture represents the value of the normalized gait angle.*

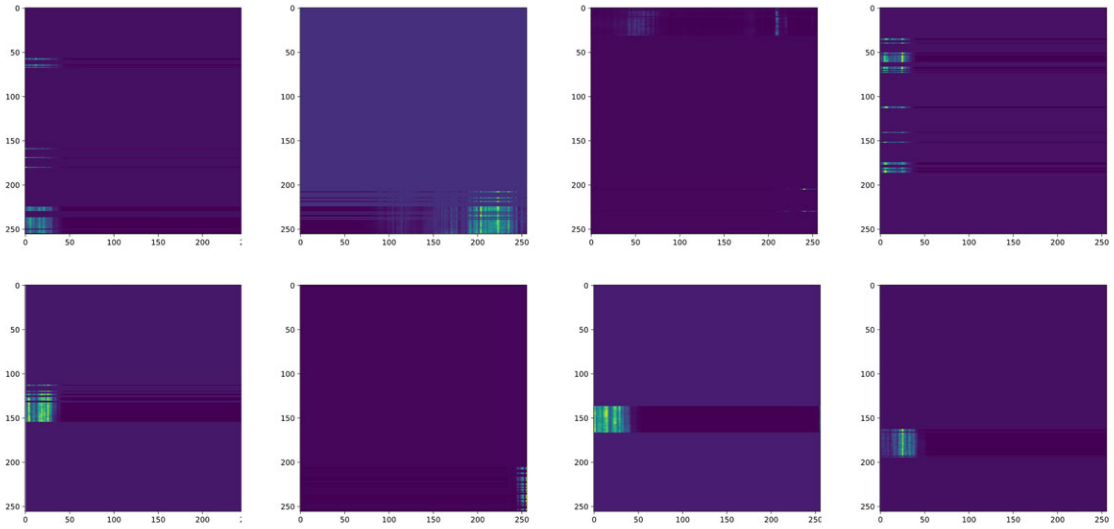


Figure 7. *Head attention of the first layer of the encoder.*

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

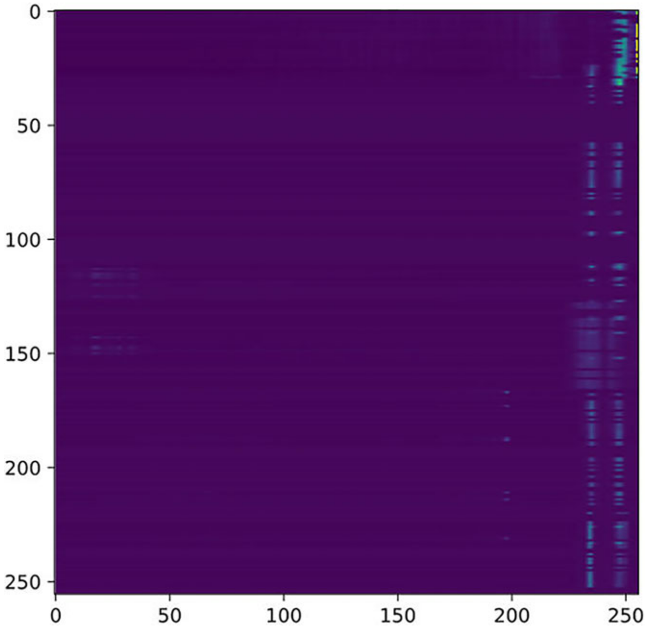


Figure 8. *The total attention of the first layer of the encoder.*

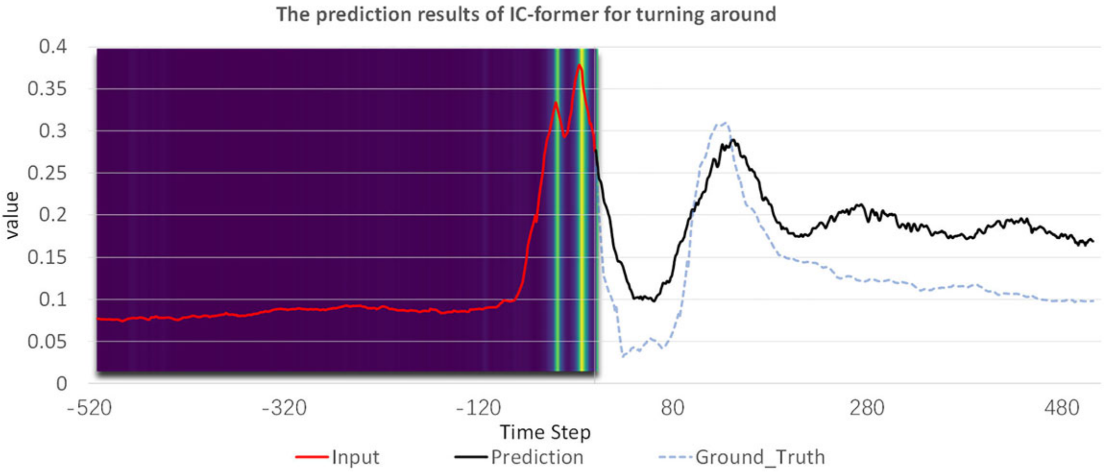


Figure 9. *Predicted trajectories of the IC-former model for turning around with the attention of the first layer of the encoder. The left curve with the heat map in the background is the input data, and the right is the output data. The higher the brightness of the heat map, the greater the weight of the corresponding input data.*

We add each line of attention to get a vector of size 1*256, and the value of this vector represents the importance of the corresponding sequence segment. We draw the input–output trajectory image with the highlighted background representing the attention. From Fig. 9, we can intuitively see that attention is focused on the two peaks at the end of the trace. This shows that the two peak data contain the most abundant information for predicting trajectories.

The attention distribution of the second layer of the encoder is relatively uniform, and attention is concentrated on the first half of the input sequence (Fig. 10). The concentration indicates that the model

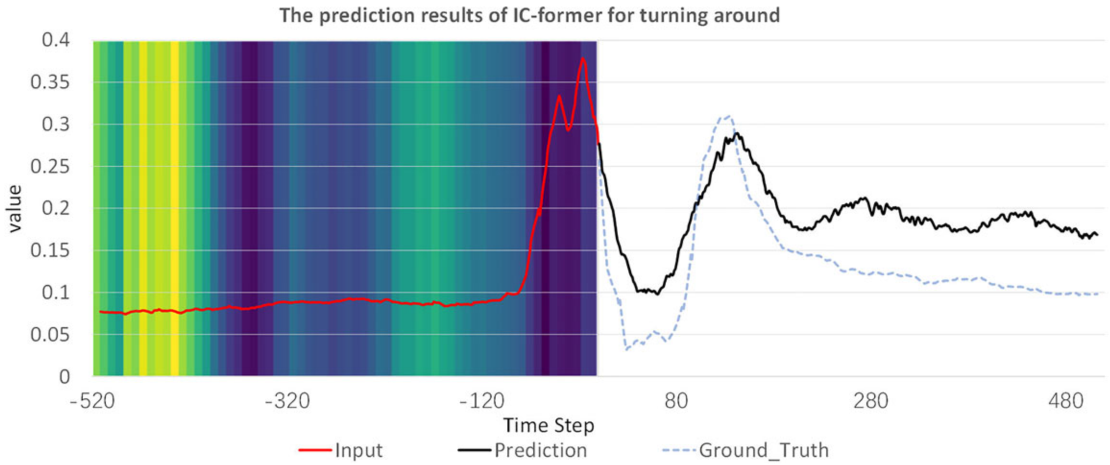


Figure 10. Predicted trajectories of the IC-former model for turning around with the attention of the second layer of the encoder. The left curve with the heat map in the background is the input data, and the right is the output data. The higher the brightness of the heat map, the greater the weight of the corresponding input data.

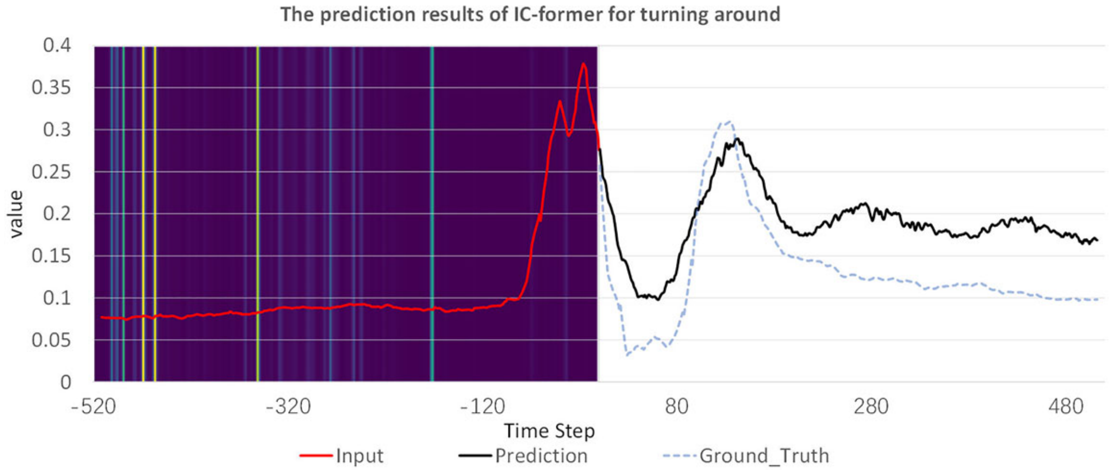


Figure 11. Predicted trajectories of the IC-former model for turning around with attention of the first layer of the decoder. The left curve with the heat map in the background is the input data, and the right is the output data. The higher the brightness of the heat map, the greater the weight of the corresponding input data.

extracts features from the first half of the input trajectory to fuse with features previously extracted from the second half.

The input to the first layer of the decoder is composed of two parts: the first part is the input gait 512-length trajectory sequence and the second part is the 512-length all-zero blank sequence. The length of the input becomes 512 through a distillation layer. The size of the attention matrix of the first layer of the decoder is 512*512. The attention distribution of the first layer of the decoder is relatively scattered, and the attention is automatically all focused on the first half with practical meaning, which is consistent with the actual nature of the input (Fig. 11).

The above are local attention images of each layer. We splice the attention of all layers into one matrix as the global attention matrix. We plot the input–output trajectory images under global attention,

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

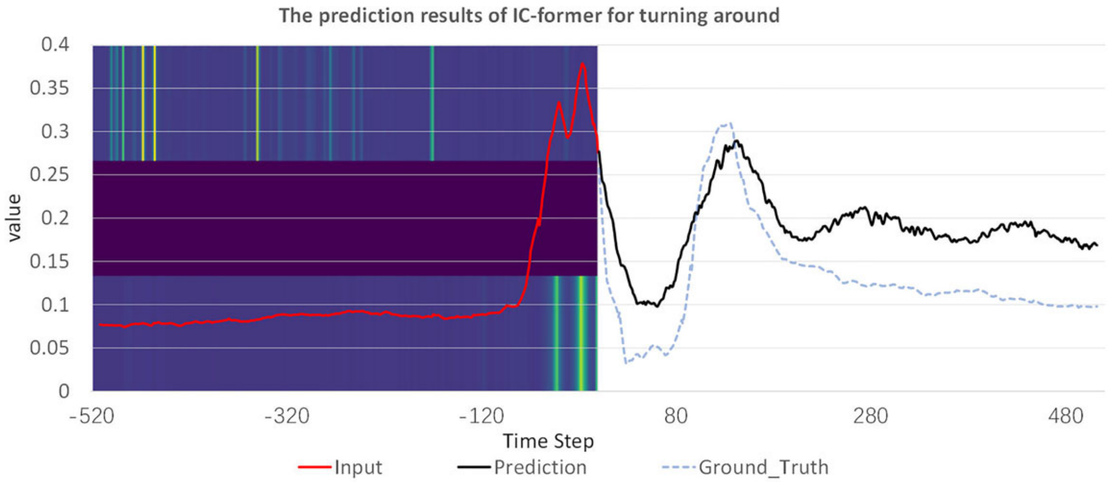


Figure 12. *Predicted trajectories of the IC-former model for turning around with global attention. The highlighted background representing attention is divided into three layers: from bottom to top the first layer of the encoder, the second layer of the encoder, and the first layer of the decoder.*

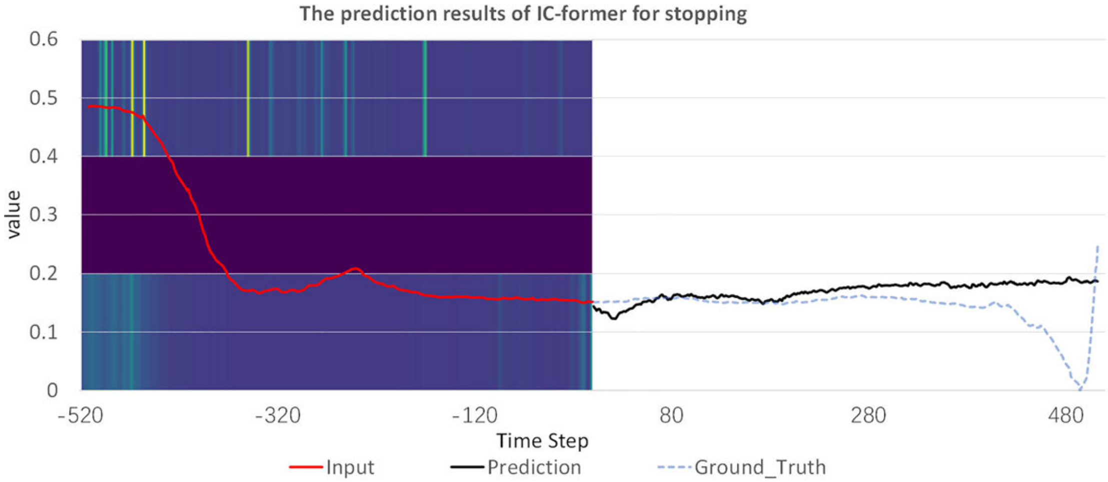


Figure 13. *Predicted trajectories of the IC-former model for stopping with all attention.*

as shown in Fig. 12. The attention of the second layer of the encoder is smaller than that of the other two layers, meaning the second layer obtains little information from the original input. But this does not mean that the auxiliary channel is unnecessary because, according to our experiments, the existence of the auxiliary channel does improve the model's prediction accuracy.

We plot the global attention images of the same IC-former model when predicting the remaining three gaits (Figs. 13, 14, 15). Attention is mainly distributed at the input sequence's back or front end for aperiodic gaits. For cyclic gaits, the attention is mainly distributed in the last cycle, and the peak part of the last cycle is the most important.

The attention distribution of the IC-former model has two things in common when predicting different gait types.

1. The attention of the input sequence of the second layer of the encoder is very low, which means that the input does not provide enough features for the model.

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

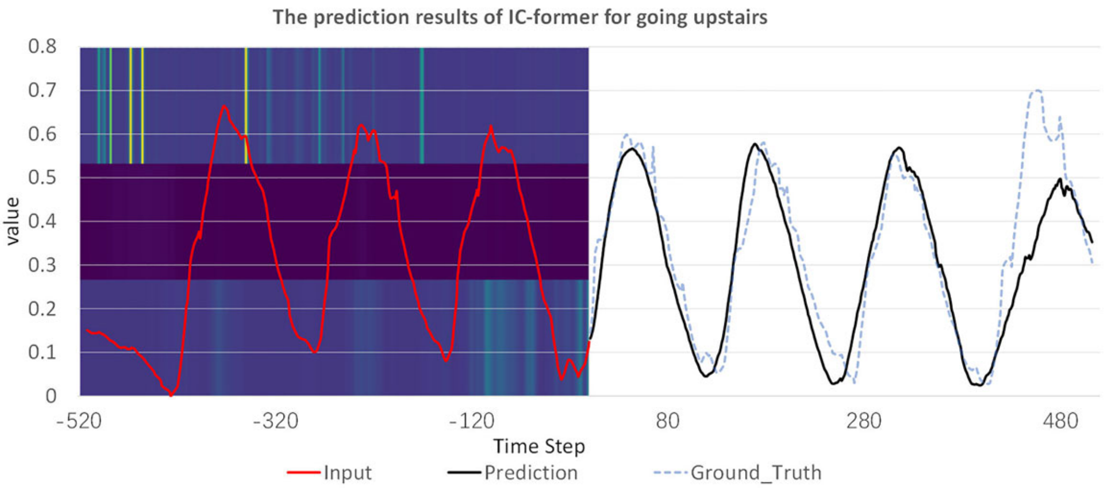


Figure 14. *Predicted trajectories of the IC-former model for going upstairs with all attention.*

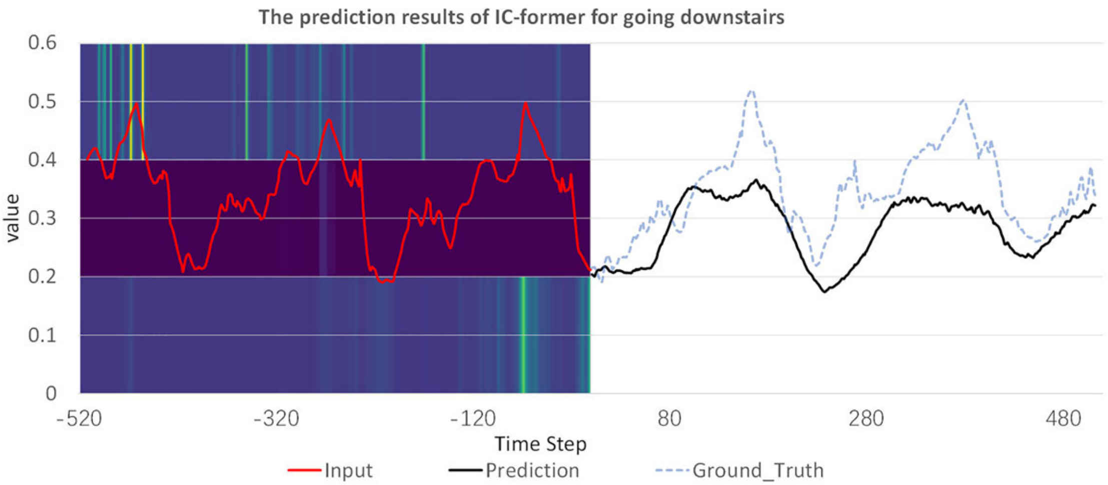


Figure 15. *Predicted trajectories of the IC-former model for going downstairs with all attention.*

2. The attention distribution of the decoder is very similar when predicting different gait types. We argue that this attention distribution is determined by the ensemble of gait data, which reflects the most informative trajectory segment overall. This attention distribution provides a good point for us to study gait data.

To further analyze and explain the IC-former for predicting gait trajectories, we plot the prediction results of another trained model for four gaits, as shown in Fig. 16.

The attention distributions of the two models are very similar. Both set high weights for the peaks of the gait trajectory. The distribution of attention is obtained based on inherent properties, such as data and model structure, rather than randomly. The IC-former effectively reveals the data basis of the neural network model. The difference between the two attention distributions is that the former is more concentrated and plays the role of the attention mechanism better.

For public evaluation, we analyze the data interpretability of a prediction result within the ECL dataset, as shown in Fig. 17. The same as when predicting the gait trajectory, the model accurately selects the key peak data to predict the future trend of the curve.

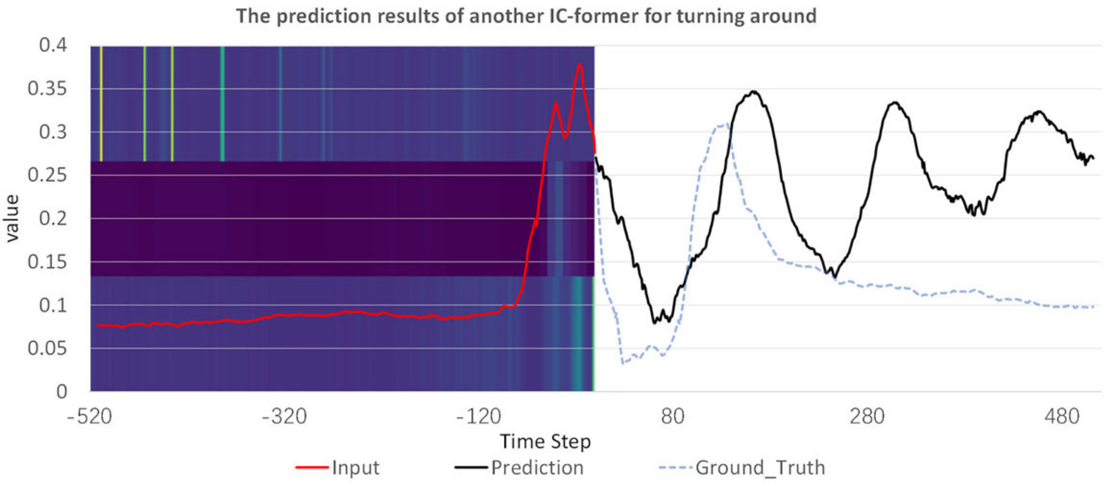


Figure 16. *Predicted trajectories of another IC-former model for going downstairs with all attention.*

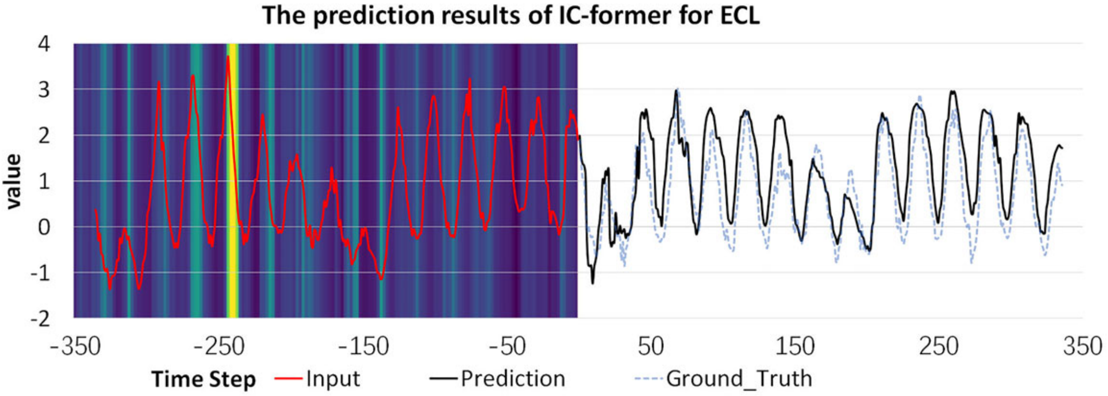


Figure 17. *Predicted trajectories of the IC-former model for ECL with attention of the first layer of the encoder.*

6. Conclusion

This paper proposes the Interpretable Multi-head Attention layer and the IC-former model. The Interpretable Multi-head Attention layer preserves the input data's mathematical meaning while enhancing the ability to extract features. The IC-former model can predict long-term sequence data with high precision and measure the input data's importance to explain the data basis of the prediction results. We tested the IC- former model on multiple datasets. The results show that the former model exceeds the prediction accuracy of many existing sequence prediction models and successfully marks important data fragments.

Based on the IC-former model, we predict long-term gait trajectories and explain the prediction results by quantifying the importance of data at different positions in the input sequence. Important input sequence segments for predicting gait trajectories contain the richest gait information, and finding them accurately is of great significance for intent perception and gait diagnosis.

Competing interests. The authors declare no competing interests exist.

Financial support. This work was supported by the National Natural Science Foundation of China [grant numbers U21B6002].

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

Ethical approval. The experimental protocol is approved by the Institute Review Board of Tsinghua University (No. 20220222). All the participants gave their consent before taking part in the study.

Author contributions. Jie Yin, Ming Zhang, and Tao Zhang conceived and designed the study. Meng Chen, Chongfeng Zhang, and Tao Xue conducted data gathering. Jie Yin and Meng Chen performed statistical analyses. Jie Yin, Ming Zhang, and Tao Zhang wrote the article.

References

[1] T. Xue, Z. Wang, T. Zhang and M. Zhang, Adaptive oscillator-based robust control for flexible hip assistive exoskeleton," *IEEE Robot. Automat. Lett.* **4**, 3318–3323 (2019).

[2] K. Seo, K. Kim, Y. J. Park, J.-K. Cho, J. Lee, B. Choi, B. Lim, Y. Lee and Y. Shim. Adaptive Oscillator-based Control for Active Lower-Limb Exoskeleton and Its Metabolic Impact. **In:** *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (2018) pp. 6752–6758.

[3] K. Tanghe, F. De Groote, D. Lefeber, J. De Schutter and E. Aertbeliën, Gait trajectory and event prediction from state estimation for exoskeletons during gait," *IEEE Trans. Neural Syst. Rehabil. Eng.* **28**, 211–220 (2019).

[4] I. Kang, P. Kunapuli and A. J. Young, Real-time neural network-based gait phase estimation using a robotic hip exoskeleton," *IEEE Trans. Med. Robot. Bion.* **99**, 1–1 (2019).

[5] Y. Wang, Z. Li, X. Wang, H. Yu, W. Liao and D. Arifoglu, Human gait data augmentation and trajectory prediction for lower-limb rehabilitation robot control using gans and attention mechanism," *Machines* **9**, 367 (2021).

[6] A. Zaroug, A. Garofolini, D. T. Lai, K. Mudie and R. Begg, Prediction of gait trajectories based on the long short term memory neural networks," *PLoS One* **16**, e0255597(2021).

[7] A. Zaroug, D. T. Lai, K. Mudie and R. Begg, Lower limb kinematics trajectory prediction using long short-term memory neural networks," *Front. Bioeng. Biotechnol.* **8**, 362 (2020).

[8] B. Su and E. M. Gutierrez-Farewik, Gait trajectory and gait phase prediction based on an lstm network," *Sensors (Basel)* **20**, 7127 (2020).

[9] D.-X. Liu, X. Wu, C. Wang and C. Chen. Gait Trajectory Prediction for Lower-Limb Exoskeleton based on Deep Spatial-Temporal Model (DSTM). **In:** *2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*, IEEE (2017) pp. 564–569.

[10] M. Karakish, M. A. Fouz and A. ELsawaf, "Gait trajectory prediction on an embedded microcontroller using deep learning," *Sensors (Basel)* **22**, 8441 (2022).

[11] I. Kang, D. D. Molinaro, S. Duggal, Y. Chen, P. Kunapuli and A. J. Young, Real-time gait phase estimation for robotic hip exoskeleton control during multimodal locomotion, *IEEE Robot. Autom. Lett.* **6**, 3491–3497 (2021).

[12] J. Cao, Z. Li and J. Li, Financial time series forecasting model based on CEEMDAN and LSTM," *Phys. A: Stat. Mech. Appl.* **519**, 127–139 (2019).

[13] A. Sagheer and M. Kotb, Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems," *Sci. Rep.* **9**, 19038 (2019b).

[14] A. Sagheer and M. Kotb, Time series forecasting of petroleum production using deep lstm recurrent networks," *Neurocomputing* **323**, 203–213 (2019a).

[15] S. Hochreiter and J. Schmidhuber, Long short-term memory," *Neural Comput.* **9**, 1735–1780 (1997).

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need, *Comput. Lang.* (2017).

[17] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty and C. Eickhoff. A Transformer-based Framework for Multivariate Time Series Representation Learning. **In:** *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, 2114–2124.

[18] M. Xu, W. Dai, C. Liu, X. Gao, W. Lin, G.-J. Qi and H. Xiong, Spatial-temporal transformer networks for traffic flow forecasting," *Signal Process.* (2020).

[19] N. Kitaev, Ł. Kaiser and A. Levskaya, Reformer: The efficient transformer," *Mach. Learn.* (2020). arXiv:2001.04451.

[20] S. Wang, B. Z. Li, M. Khabsa, H. Fang and H. Ma, Linformer: Self-attention with linear complexity," *Mach. Learn.* (2020).

[21] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang and X. Yan, Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Mach. Learn.* (2019).

[22] I. Beltagy, M. E. Peters and A. Cohan, Longformer: The long-document transformer," *Comput. Lang.* (2020).

[23] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong and W. Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-series Forecasting. **In:** *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**, (2021a), 11106–11115.

[24] J. Chen, L. Song, M. Wainwright and M. Jordan. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. **In:** *International Conference on Machine Learning*, PMLR (2018) 883–892.

[25] S. Jain and B. C. Wallace, Attention is not explanation," *Comput Lang.* (2019). arXiv:1902.10186.

[26] J. Vig and Y. Belinkov, Analyzing the structure of attention in a transformer language model," *Comput. Lang.* (2019).

[27] S. Vashishth, S. Upadhyay, G. S. Tomar and M. Faruqui, Attention interpretability across NLP tasks," *Comput. Lang.* (2019).

[28] B. N. Oreshkin, D. Carpov, N. Chapados and Y. Bengio, N-beats: Neural basis expansion analysis for interpretable time series forecasting," *Mach. Learn.* (2019).

https://doi.org/10.1017/S0263574723001492 Published online by Cambridge University Press

[29] B. Lim, S.Ö. Arık, N. Loeff and T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Mach. Learn.* (2021).
[30] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong and W. Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-series Forecasting. **In:** *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**, (2021b), 11106–11115.
[31] A. A. Ariyo, A. O. Adewumi and C. K. Ayo. Stock Price Prediction Using the Arima Model. **In:** *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, IEEE (2014) pp. 106–112.
[32] S. J. Taylor and B. Letham, Forecasting at scale," *PeerJ* (2018).
[33] D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Comput. Lang.* (2014). arXiv preprint arXiv:1409.0473.
[34] G. Lai, W.-C. Chang, Y. Yang and H. Liu. Modeling Long-and Short-term Temporal Patterns with Deep Neural Networks. **In:** *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018) pp. 95–104.
[35] D. Salinas, V. Flunkert, J. Gasthaus and T. Januschowski, DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecast.* **36**, 1181–1191 (2020).

Cite this article: J. Yin, M. Chen, C. Zhang, T. Xue, M. Zhang and T. Zhang (2024). "Interpretable long-term gait trajectory prediction based on Interpretable-Concatenation former", Robotica **42**, 367–388. https://doi.org/10.1017/S0263574723001492