ORIGINAL SCIENTIFIC PAPER

A New Table Tennis Match Stroke Forecasting Method Using Transformer-Based Deep Neural Networks

Hsu, Ming-Hwa[1], Kondrič, Miran[2], Fan-Chiang, I-Yun[3], Wu, Jiunn-Lin[3]*

Affiliations: [1]Graduate Institute of Sports and Health Management, National Chung Hsing University, [2]Racket Sports Department, Faculty of Sport, University of Ljubljana, [3]Department of Computer Science and Engineering, National Chung Hsing University

Correspondence: Jiunn-Lin WU, National Chung Hsing University, Department of Computer Science and Engineering, 145 Xingda Rd., Taichung City, TAIWAN, E-mail: jlwu@cs.nchu.edu.tw

Abstract

This paper proposes a novel approach for forecasting stroke outcomes in table tennis matches using a transformer-based deep neural network architecture. Table tennis rallies' highly dynamic and fast-paced nature makes trajectory and stroke prediction a particularly challenging. Our model employs dual encoder-decoder structures to extract contextual features from rally sequences and individual players separately, addressing this issue. The model uses attention mechanisms to evaluate the relative importance of stroke techniques and landing positions. Key game-specific attributes, including the ball speed and spin, are incorporated to enhance strategic insight and prediction accuracy. We further introduce a stroke-level event stream representation to convert raw match records into a structured and consistent format, significantly improving interpretability and enabling more efficient analysis. A feature fusion network is employed to integrate rally dynamics and player-specific traits, allowing the model to accurately forecast the type and landing zone of the next stroke.
The "Intellectual Tactical System in Competitive Table Tennis" system database provided the table tennis match data collected in this study. This database collects the data of Lin Yun-Ju's matches against male opponents (23 matches, 121 games, 2,225 rallies, totaling 10,517 hits, an average of 4.7 hits per rally). Experimental results show that the proposed architecture significantly improves prediction performance. On the dataset, it achieved top-1 accuracies of 57.2% for stroke type and 42.8% for landing zone (spot), with top-5 accuracies of 98.2% and 91.8%, respectively. Furthermore, we visualize prediction outcomes alongside known stroke data, providing a novel perspective for tactical analysis. This visualization facilitates intuitive understanding for coaches and players, offering a valuable tool for performance evaluation and strategic development in professional table tennis.

Keywords: *Table Tennis, Deep learning, Attention Mechanism, Stroke forecasting, Technical and tactical analysis*

@MJSSMontenegro
TRANSFORMER-BASED DEEP NEURAL NETWORKS IN TABLE TENNIS
http://mjssm.me/?sekcija=article&artid=311

Cite this article: Hsu, M.-H., Kondrič, M., Fan-Chiang, I.-Y., & Wu, J.-L. (2025). A new table tennis match stroke forecasting method using transformer-based deep neural networks. *Montenegrin Journal of Sports Science and Medicine, 14*(1), Ahead of Print. doi: 10.26773/mjssm.260306

Received: 01 November 2025 | Accepted after revision: 01 December 2025 | Early access publication date: 01 June 2025 | Final publication date: 01 July 2026

Conflict of interest: None declared.

Introduction

With the rapid development of innovative sports technologies, data collection has become an integral aspect of ball games. Through comprehensive analysis of match data, game strategies are no longer solely reliant on intuitive experience; instead, they are grounded in concrete data evidence, enhancing the accuracy and effectiveness of tactical decision-making. In the highly competitive sport of table tennis, analyzing an opponent's match data enables players to understand their opponent's behavioral patterns better, allowing them to target weaknesses during matches. Players can gain a deeper understanding of their opponents' stroke patterns, tactical approaches, and technical characteristics by systematically collecting and analyzing various types of match data. This facilitates a more comprehensive understanding of the game situation, improving the precision and flexibility of tactical decision-making.

In table tennis, the variability of stroke patterns presents a significant challenge in predicting stroke trajectories. The factors influencing stroke trajectories in table tennis are highly complex, including variables such as stroke speed and spin angle, which can affect the ball's trajectory. Traditional methods for predicting stroke trajectories predominantly rely on observation and subjective judgment. Players and coaches often spend extensive time analyzing match footage frame by frame to anticipate the opponent's next move. These methods are not only time-consuming and labor-intensive, but they are also prone to errors and inconsistencies. However, implementing a stroke trajectory prediction system enables effective extraction and integration of key information from the complex features of players for analysis. This significantly enhances the speed of predictions, providing players with timely data support during the early stages of matches and actual gameplay. Such systems enable players to adjust their tactics quickly in response to their opponent's changes.

Recent research in table tennis has primarily focused on identifying stroke-related information from video footage or quantifying stroke performance through match records. In previous image analysis studies (Kulkarni & Shenoy, 2021), human pose estimation techniques have been utilized to recognize strokes and predict ball landing points by analyzing the positions of various human joint points in images. However, due to the high similarity in stroke actions among players, it is often difficult for computers to distinguish between them accurately. Furthermore, image recognition is susceptible to factors such as image quality and environmental conditions, resulting in poor recognition results, particularly when scene changes occur. Therefore, using human pose estimation methods for stroke prediction in table tennis videos may not be the ideal approach.

To enhance the accuracy of match predictions, many studies have employed the analysis of match records (Chiang & Denes, 2023; Liu et al., 2024,2025; Tsai et al., 2023), often utilizing machine learning or neural network techniques. However, current comprehensive table tennis datasets primarily focus on recording landing points, scores, and events. As a result, much of the existing research has concentrated on using statistical data to predict winning probabilities (Huang et al., 2021), overlooking the significance of other critical information, such as stroke types, in technical and tactical analysis. Thus, to accurately predict future stroke types and landing points, it is essential to consider the four fundamental elements of table tennis tactics: landing point, technique, speed, and spin (Liu et al., 2024). Given the high variability of the sport, variations in these elements - beyond the ball's landing point - must be considered. By effectively leveraging these features, we can gain a more objective understanding of different stroke characteristics, which, in turn, will aid in predicting stroke types and devising strategic approaches for competitive table tennis.

This study formulates stroke prediction as a sequence prediction task to address the challenges and limitations present in existing table tennis analysis applications. This constitutes the core research objective, wherein we employ sequence-to-sequence models to extract match features from input sequences and use an encoder to generate predictive outcomes. Since table tennis rallies consist of alternating strokes between two players, the prediction task is inherently turn-based rather than a conventional single-target sequence prediction problem. Furthermore, a player's overall playing style and rally dynamics may vary depending on the opponent, necessitating a model architecture that can account for such variability. To this end, the sequence-to-sequence model is divided into two distinct components, enabling the learning of player-specific and rally-specific features.

This study builds upon ShuttleNet (Wang et al., 2022a), a stroke prediction method designed to learn stroke techniques and landing points. ShuttleNet employs dual encoders and decoders to model rally progression and player style, effectively capturing stylistic differences between players. It then integrates these features via a fusion network to predict the technique and landing point of the next stroke. To improve the accuracy and comprehensiveness of technical and tactical predictions in table tennis, we convert match records into an event-stream representation focused on stroke sequences. This representation facilitates more efficient and precise analysis. Moreover, we incorporate critical attributes, including the spin and speed, into the rally and player extractors, allowing for more nuanced judgments of stroke techniques and ball landing points.

Related Works

Despite the growing interest in landing point prediction for technical and tactical analysis in table tennis, comprehensive literature remains on full-scale stroke prediction in the sport.

DOI 10.26773/mjssm.260306

This session explores prior studies on landing point prediction in table tennis, discusses the application of sequence prediction methods in sports trajectory forecasting, and further examines stroke forecasting within turn-based sports.

Zhang et al. (2010) proposed a high-speed stereo vision system for table tennis. This system utilizes two smart cameras with a distributed parallel processing architecture based on local networks to track the ball. It computes the ball's trajectory and landing point by analyzing the coordinates obtained from the cameras using mathematical methods. The study primarily employs traditional image processing techniques, such as adjacent frame difference, to detect moving objects and contour analysis to identify the ball's position. The center of the ball is then accurately calculated using the growth of sampled points method. The system derives the ball's 3D position through stereo vision and integrates these equations to predict the ball's flight path, hitting point, and landing point.

While this method effectively predicts landing and hitting points, providing a comprehensive system for practical application, its image processing technique requires a stable environment. Variations in lighting, background, or player movements can reduce accuracy. The system relies on pre-designed mathematical models and requires extensive manual parameter adjustments to adapt to different scenarios, making it time-consuming and less adaptable to environmental changes. The system's complex hardware and software architecture also require high equipment standards, which increases implementation difficulty and cost.

The researchers (Wu & Koike, 2020) proposed a landing point prediction system for table tennis based on deep learning techniques. This system employs a single camera to capture the player's serving motion, and the video data is processed using the lightweight residual convolutional neural network ResNet50, which performs well in real-time human pose estimation. The network accurately estimates the player's 2D joint positions, extracting motion features before the stroke. These pose data are then input into a long short-term memory (LSTM) network, effectively capturing dependencies in the sequence of stroke actions, leading to accurate landing point predictions. Finally, the results are projected onto the table to provide real-time feedback during training.

Although this system enables real-time prediction of landing points without requiring model parameter adjustments, it mainly relies on the player's current hitting posture for prediction, which overlooks essential factors such as hitting techniques and spin. Consequently, the prediction is less effective in complex situations involving spin strokes. Additionally, like the previously mentioned image processing methods, the system is susceptible to environmental factors that affect image recognition accuracy. Precise camera calibration is necessary to ensure correct image angles, which is a cumbersome process susceptible to interference, presenting operational challenges and increasing maintenance costs.

In recent years, the application of transformers in deep learning has become increasingly widespread, particularly excelling in natural language processing and other sequence modeling tasks. Giuliari et al. (2021) introduced a transformer network based on an attention mechanism to predict future pedestrian trajectories. The model's input consists of an individual's current and past positions; its output is the predicted future positions. The input positions are initially combined with time encoding to assign a unique marker to each time point. The encoder then captures feature relationships within the observed sequences, primarily utilizing the attention mechanism to determine which parts of the sequence to focus on to learn features more effectively. The decoder then autoregressively predicts future positions.

Unlike traditional LSTM networks, transformers can process input data in parallel, resulting in faster training speeds and improved performance, particularly in capturing global relationships for long-term predictions. Their flexibility and scalability enable them to be applied to tasks beyond natural language processing, including image processing and time-series data analysis. This versatility makes transformers well-suited for sequence prediction and feature extraction tasks. However, conventional sequence prediction methods typically only consider the sequential relationship of data to generate subsequent predictions. In turn-based sports like table tennis, it is crucial to consider the sequence of strokes and the stylistic differences between players. Predicting based solely on stroke sequences may reduce accuracy. Thus, transformers require further refinement to effectively apply to table tennis stroke prediction, thereby achieving the desired results.

The researchers (Wang et al., 2022a) proposed a sequence prediction model, ShuttleNet, for predicting badminton strokes, marking the first research effort to address stroke prediction in this context. This model emphasizes rally progression and player style as key features, effectively integrating them to understand match dynamics and player characteristics comprehensively. The model utilizes a transformer-based rally extractor to capture overall rally features from sequential data and incorporates a multi-head attention mechanism (Vaswani et al., 2017) to capture global relationships better. Additionally, a transformer-based player extractor splits the sequential data into two subsequences, using encoder-decoder structures to extract feature information for each player. A position-aware fusion network integrates player and rally information from different transformers to predict future stroke techniques and landing points.

While this method is based on past match records for predictions, providing relatively reliable results by capturing players' stroke characteristics, it was initially designed for badminton, considering only landing points and techniques as key features. However, table tennis must also consider additional critical factors such as speed and spin. These elements are vital for accurate stroke prediction in table tennis, and relying solely on landing points and technique features may not fully capture the complexity of table tennis dynamics. Therefore, when applying ShuttleNet to table tennis stroke forecasting, adjustments are necessary to integrate more table tennis-specific features, including speed and spin, to enhance the model's predictive capability.

A review of the literature on table tennis stroke prediction reveals that while Wu and Koinke (2020) have explored landing point prediction using deep learning methods, primarily through LSTM networks for temporal prediction in human pose estimation, and Wang et al. (2022a) used transformers for badminton stroke forecasting, predicting stroke techniques in table tennis remains largely underexplored. Previous studies on table tennis prediction have primarily focused on landing points and trajectories, with image analysis methods requiring high environmental and equipment standards, which present challenges for practical implementation.

This study aims to accurately predict table tennis techniques and stroke placements. To achieve this, we convert match records into an event-stream representation based on individual strokes, enhancing data consistency and readability, facilitating more efficient and accurate analysis and prediction. We propose a deep learning approach based on ShuttleNet for the stroke prediction framework. The main contributions of this study include incorporating key factors specific to table tennis - namely ball speed and spin, and modifying the model's loss function better to suit the analysis of table tennis tactics and strategies. Additionally, to provide more diverse perspectives in tactical analysis, we employ a moving window approach to capture short-term rally patterns within matches, enabling in-depth analysis of localized tactical scenarios and yielding more precise predictive outcomes.

Methodology

This study proposes a deep learning-based method for predicting table tennis strokes to analyze players' stroke strategies during matches. By forecasting players' potential stroke techniques and placements throughout a match and presenting these predictions graphically, we investigate the relationship between different players' techniques and tactical approaches. In the following, we first outline the preprocessing steps used to convert raw match data into sequential formats suitable for deep learning. The proposed stroke prediction model architecture is then described, which employs two transformer modules to extract features from rally dynamics and player styles, integrating them through a gated multimodal network. Finally, we present the methods used for predicting stroke techniques and placements, along with the associated loss functions.

Data Preprocessing

In sports data analysis, data is typically annotated manually by experts. Experts review match footage and convert the content into a standardized format. For example, in football match analysis, SPADL (Soccer Player Action Description Language) (Decroos et al., 2018) consolidates existing football match data and represents continuous movements in an event stream format, thereby enhancing data analysis and interpretation. In this study, we utilize table tennis match records as our dataset (Liu et al., 2024), which detail the players, stroke techniques, placements, and other information for each rally, providing rich material for analysis.

Due to manual annotation, data inconsistencies and missing values can occur. Additionally, existing representations are based on rallies, which involve recording multiple consecutive strokes, making it difficult to discern individual stroke information quickly. To address these issues, we refer to the BLSR (Badminton Language for Shot Rally) (Wang et al., 2022b) used in badminton, designing a preprocessing method to convert match records into an event stream representation based on individual strokes, thereby improving data consistency and readability, and enhancing the efficiency and accuracy of subsequent analysis and prediction tasks. The representation methods are shown in Table 1, which references the BLSR format and combines the match record's description of table tennis with a stroke-based representation unit.

In a match, there are multiple games, each subdivided into rallies, and each rally consists of various strokes. Each stroke involves interactions between two players. This structure is similar to a corpus in natural language, so we treat each stroke as a word and each rally as a sentence based on the relationships in natural language corpora, sentences, and words. Each rally $R^{(i)}$ can be seen as a stroke sequence $\{s_1^{(i)}, s_2^{(i)}, \ldots, s_{N(i)}^{(i)}\}$, where each stroke $s_n^{(i)}$ contains the following six types of information:

(a) Player: The player who performed the stroke.

(b) Technique: The technique used for the stroke.

(c) Forehand/Backhand: Hit the ball with forehand or backhand technique.

(d) Spin: The type of spin is applied to the ball.

(e) Speed: The speed of the stroke.

(f) Spot: The location where the ball lands on the table.

Information for each rally $R^{(i)}$ can be described as follows:

(a) Player A Score: The score of the player who strikes first in the rally.

(b) Player B Score: The score of the player who strikes second in the rally.

(c) Get-point Player: The player who won the rally.

(d) End Reason: The reason the rally ended, such as out of bounds or net.

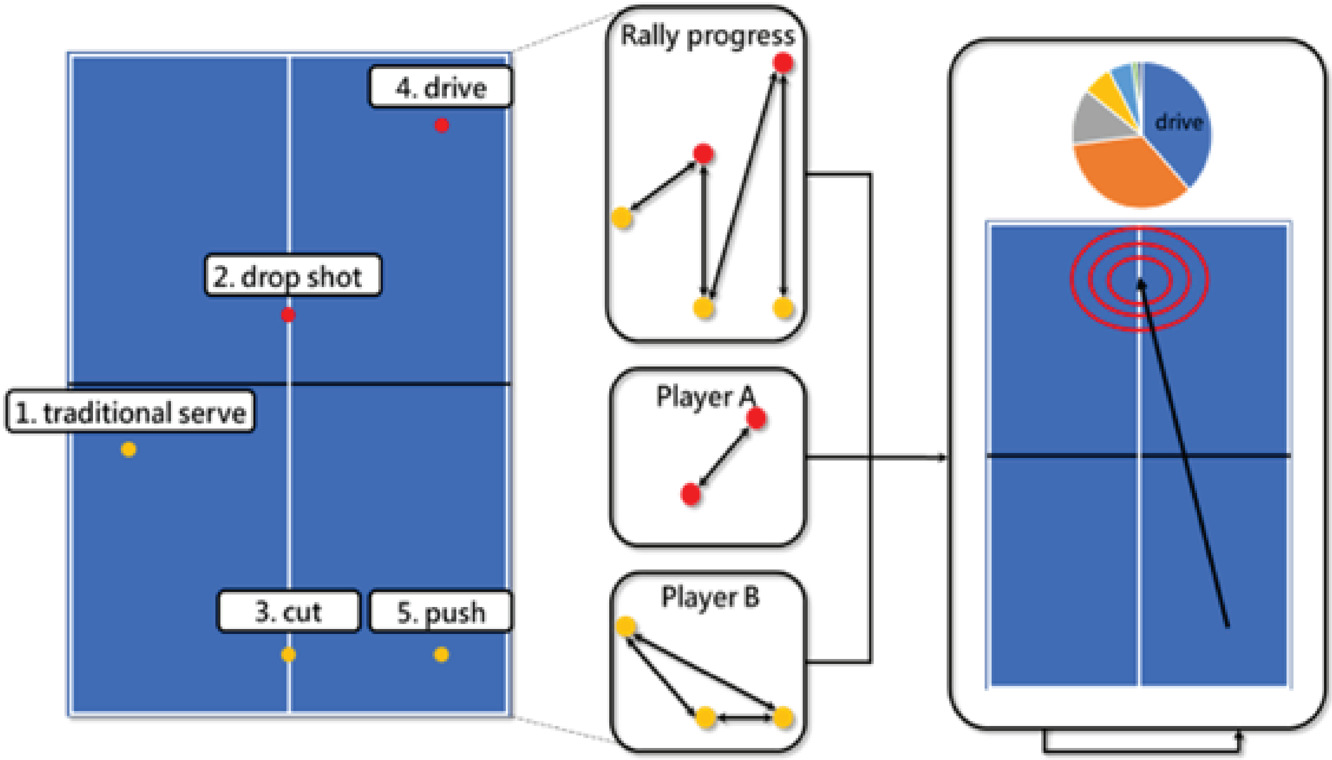
Deep Neural Network for Stroke Forecasting in Table

DOI 10.26773/mjssm.260306

Tennis

Each rally in table tennis matches consists of multiple strokes, with two players alternating hits. Thus, stroke prediction in table tennis is defined as a turn-based prediction problem (Wang et al., 2023). We first simulate future player behavior based on known rally information and player styles. Thus, we utilize two feature extractors to separately extract rally information and player styles and then predict the technique and landing point of the next stroke, as shown in Figure 1.

Figure 1. Diagram of a turn-based table tennis stroke forecasting prediction method



We designed a deep learning-based table tennis stroke prediction model, primarily utilizing the ShuttleNet (Wang et al., 2022a) architecture. The aim is to objectively forecast stroke techniques and landing points in table tennis matches. This framework includes two extractors for extracting rally progress and player styles, consisting mainly of transformers' encoders and decoders. The model replaces the self-attention mechanism with a type-area attention mechanism to better integrate the stroke technique and the landing point. A transformer-based rally extractor generates the rally context, while a transformer-based player extractor obtains the contexts of the two players. A position-aware gated fusion network further fuses these contexts to predict the technique and landing point of the next stroke.

Firstly, a rally in a table tennis match is represented as $R=\{S_1,S_2,\ldots,S_n\}$, composed of multiple strokes, which represent the known information for that rally. The i-th stroke is performed by a player and is denoted as $S_i=\langle p_i,t_i,a_1,r_i,s_i\rangle$, where it includes five key pieces of information: p_i represents the player who performed the stroke, t_i represents the stroke technique, a_i represents the ball's landing points, r_i represents the type of spin applied to the stroke, and s_i represents the speed of the stroke. The stroke prediction problem is defined as forecasting stroke techniques and landing points, given the known information for the previous τ strokes $\{\langle t_i,a_1\rangle\}_{(i=1)}^{\tau}$ combined with player, speed, and spin information, to predict the technique and landing point for the next *n* strokes $\{\langle t_i,a_1\rangle\}_{(i=\tau+1)}^{(\tau+n)}$.

The model can effectively capture complex strategy patterns in table tennis matches and accurately predict future stroke techniques and landing points with this structure. This enhances stroke prediction accuracy and provides robust support for subsequent tactical analysis and match strategy.

Embeddings Layer

In table tennis matches, each stroke includes technique, landing point, spin, speed, and player information. Before inputting this data into the model, it needs to be converted into vector representations so that the model can understand and process the input information. Therefore, we utilize word embedding techniques to effectively capture the relationships between input data, generating dense, low-dimensional vector representations that improve model processing efficiency and prediction accuracy.

Word embedding aims to convert a given input sequence into a representation that is model-readable. A standard method uses One-Hot Encoding, which converts input words into sequence vectors. The length of the vector corresponds to the number of input categories, with only one position being 1 and the rest being 0, where the 1 position corresponds to the index of the word in the vocabulary. However, this method produces large, sparse vectors, which leads to the curse of dimensionality and negatively impacts model performance. Additionally, one-hot Encoding does not capture the semantic relationships between words. Therefore, we utilize word embedding as a representation method, treating it as a lookup table where the input vectors are multiplied by a weight matrix to map into a lower-dimensional vector space, thereby capturing dependencies between words and providing contextual information.

To effectively utilize player information in technique and placement, we add speed, spin, and player information to the technique and landing point vectors, thereby better reflecting the importance of speed and spin in table tennis tactics. For the *i*-th stroke, the embedding layer e^i is calculated as follows:

$$e_i = \langle e_i^t, e_i^a \rangle = \langle t_i' + r_i' + s_i' + p_i', a_i' + r_i' + s_i' + p_i' \rangle \quad (1)$$

where t_i' is the technique vector obtained by projecting t_i' using $M_t \in R^{(N_t \times d)}$, N_t being the number of stroke technique categories; r_i' is the spinvector obtained by projecting r_i using $M_r \in R^{(N_r \times d)}$, N_r being the number of spin categories; s_i' is the speed vector obtained by projecting s_i using $M_s \in R^{(N_s \times d)}$, N_s being the number of speed categories; p_i' is the player vector obtained by projecting a_i using $M_p \in R^{(N_p \times d)}$, N_p being the number of player categories; and a_i' is the placement vector obtained by projecting a_i using $M_a \in R^{(N_a \times d)}$, N_a being the number of placement categories.

Although word embeddings provide rich semantic information for input sequences, transformers process all strokes in parallel and cannot capture the positional information of each stroke in the sequence. Therefore, position encoding is introduced to provide the relative position of each stroke within the sequence. According to the original paper (Wang et al., 2022), strokes that are closer together have more similar position encoding vectors. We can effectively learn the relative positions between strokes by adding the position encoding vectors to each stroke's word embedding vectors. The position encoding vectors are determined based on the order of strokes in the input sequence. This allows the model to retain the sequence order and learn the sequential relationships between strokes. The calculations for position encoding are shown in Equations (2) and (3):

$$pe_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \quad (2)$$

$$pe_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \quad (3)$$

where pos represents the position of the stroke in the sequence, d is the dimensionality of the word vectors, 2i represents the even dimensions, and 2i + 1 represents the odd dimensions. This encoding method ensures that different positions are not encoded as identical values across all dimensions, thereby giving each position a unique encoding.

Rally Progress Extractor

During a rally, players develop strategies to counter their opponents based on the current state of the game. We simulate this process using a rally extractor, as shown in Figure 2(a). The known match records are converted into sequence information, and the input to the rally encoder is represented as:

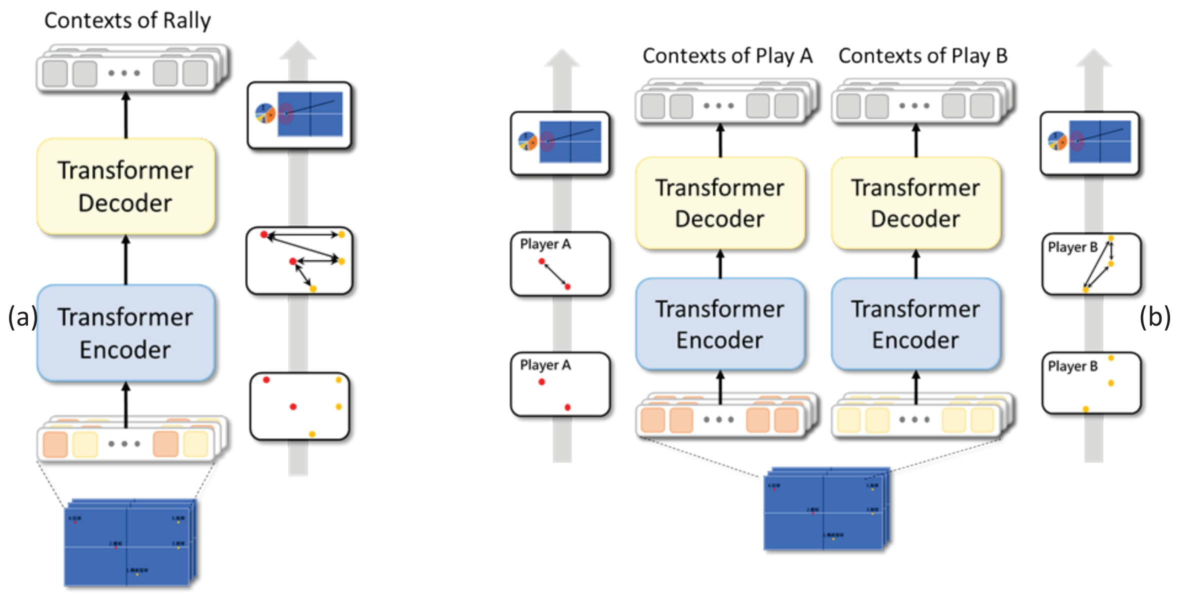
$$E_R = (\langle e_1^t + pe_1, e_1^a + pe_1 \rangle, \langle e_2^t + pe_2, e_2^a + pe_2 \rangle, \ldots) \quad (4)$$

where pe_i is the stroke embedding vector with added position encoding. The encoder in the transformer extracts features from each stroke, enabling the model to effectively learn the relationships between strokes and reflect the dynamic state of the game. The rally extractor primarily reflects the current state of the rally, identifying whether a player is currently in an offensive or defensive position. The decoder then uses the extracted rally information to generate contextual information for the rally, as shown in Equation (5):

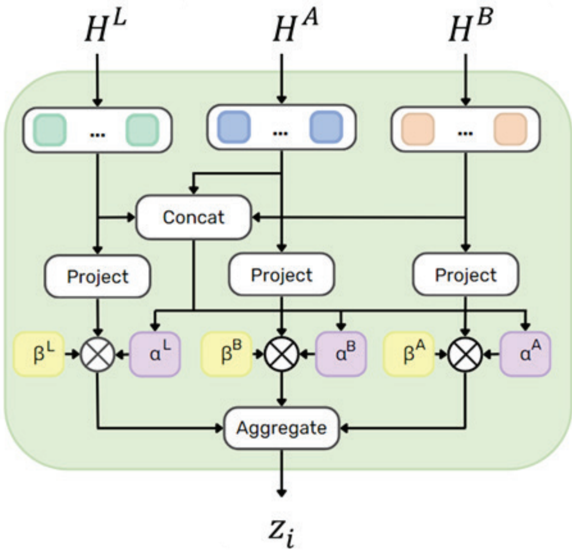
$$H^R = (h_{(\tau+1)}{}^R, h^{(\tau+2)R}, \ldots) \quad (5)$$

where $h_i^R \in R^d$ represents the contextual information generated by the encoder for the   -th stroke. The generated information is then input into the feature fusion network to infer the possible strategy for the next stroke.

Figure 2. Diagram of the network architecture. (a) rally progress extractor; (b) player style extractor (c) position-aware gated fusion network.



 DOI 10.26773/mjssm.260306



(c)

Player Style Extractor

In table tennis matches, players must devise strategies based on the information from previous strokes and aim to minimize their opponents' advantages while maximizing their own. Therefore, we utilize a player style extractor, as shown in Figure 2(b), to capture each player's playing style and consider their stroke preferences, which enables us to predict the next stroke more accurately for different players. The input sequence data is divided into two sub-sequences according to the players, as represented by Equations (6) and (7):

$$E_A=(e_1,e_3,\ldots) \tag{6}$$
$$E_B=(e_2,e_4,\ldots) \tag{7}$$

where E_A is the sequence for Player A and E_B is the sequence for Player B. The two sequences are first added with position encoding. Two separate encoders extract the stroke style for each player, and two decoders generate contextual information for both players, as shown in Equations (8) and (9):

$$H_A=(h_{(\tau+1)}^{A}, h_{(\tau+1)}^{A}, h_{(\tau+2)}^{A}, h_{(\tau+2)}^{A},\ldots) \tag{8}$$
$$H_B=(0, h_{(\tau+1)}^{B}, h_{(\tau+1)}^{B}, h_{(\tau+2)}^{B}, h_{\tau+2)}^{B},\ldots) \tag{9}$$

where $h_i^A \in R^d$ represents the context generated by the encoder for the -th stroke of the serving player, while $h_i^B \in R^d$ represents the context for the receiving player. Since the serving player performs the first stroke and odd-numbered strokes, the receiving player's sequence has the first stroke filled with zeros and the odd-numbered strokes filled with the contexts of the previous stroke.

The rally progress and player style extractor used in this study employs a transformer as the backbone network to extract features, primarily consisting of an encoder and a decoder. Initially, the encoder transforms the input vectors of each stroke into a set of high-dimensional feature vectors. Subsequently, the decoder generates feature vectors predicting outcomes based on these vectors and uses the prediction results of the previous stroke as input into the decoder.

The encoder is the main component within the transformer, primarily converting input sequences into high-dimensional vector representations. It achieves this by learning the semantic and structural information of the input data, providing rich contextual information for the subsequent operations of the decoder. The encoder comprises six identical sub-layer stacks, each consisting of two main modules: the multi-head self-attention mechanism and a feed-forward network.

The input sequence first undergoes the multi-head self-attention mechanism, which captures global relationships by calculating attention weights between each position in the sequence and every other. This information is then subjected to a nonlinear transformation through a feed-forward network to extract features further, allowing for better learning of complex features and semantic information. Each sub-layer includes residual connections and layer normalization, which help alleviate the vanishing gradient problem and enhance model training efficiency. After processing through multiple encoder layers, the enriched information extracted from the input sequence is represented as high-dimensional vectors, which are then passed to the decoder to generate the final output sequence.

Attention modules (Vaswani et al., 2017) have achieved significant success in deep learning, particularly within the Transformer, which is widely used due to the powerful performance of its self-attention mechanism. The main goal is to calculate the similarity between each element in the input sequence and reweight the results of other elements based on these similarities. The multi-head self-attention, as the technological core of the Transformer, can dynamically learn the semantic features. Typically, it computes attention weights through query, key, and value vectors to determine which features the model should focus on. Attention scores are normalized using activation functions and ultimately used to generate the output of the attention module.

However, in the stroke strategies in table tennis matches, the technique of each stroke mainly depends on the player's previous stroke, while the landing point is determined based on the opponent's past playing habits. Since traditional self-attention mechanisms can only focus on the same position, the model tends to focus on a single position when predicting stroke types and landing points, which is insufficient for fully capturing complex contextual information. To address this issue, Wang et al. (2022a) proposed the Type-Area Attention Layer (TAA), which separately calculates the importance of stroke technique and landing point, then combines them to generate the final attention scores.

Feedforward neural networks play a critical role in enhancing the feature vector representation capabilities and task performance of query indices. Each encoder layer follows the multi-head self-attention mechanism. It consists of two fully connected layers that enable independent feature transformations at each position. This network can be combined with the global context captured by the self-attention mechanism, allowing the encoder to possess stronger learning capabilities when processing sequential data. Additionally, nonlinear activation functions are employed between layers to capture more complex features and patterns. By processing input data from different positions independently in a consistent manner, the network ensures uniformity in feature extraction from the input sequence, as shown in Equation (10):

$$FFN(x)=max(0,xW_1+b_1)W_2+b_2 \tag{10}$$

where x represents the output from the self-attention mechanism, W_1 and W_2 are the trainable weight matrices of the first and second layers, respectively, and b_1 and b_2 are bias vectors. The first fully connected layer maps the input dimension from d_{model} = 512 to a higher dimension d_{ff} = 2048. After applying the ReLU activation function, the second fully connected layer projects the dimension back to d_{model}. Feedforward neural networks effectively enhance the model's feature representation capabilities through local nonlinear transformations. Combined with the self-attention mechanism, this significantly improves the performance of the transformer model in handling stroke sequence processing tasks.

The decoder's primary function is to generate new output sequences step-by-step based on the previously generated output sequences. Similar to the encoder, the input sequence is first added with positional encoding to retain the order information of each position in the sequence. The input vectors are then passed through multiple decoder layers, primarily including three modules: the Masked Multi-Head Attention Layer, the Multi-Head Attention Layer, and the feedforward neural network.

The masked multi-head attention Layer (Vaswani et al., 2017) introduces a masked mechanism to prevent the model from attending to future positions in the sequence during the generation process, ensuring that each step's prediction is based solely on the parts that have already been generated. The Multi-Head Attention Layer facilitates interactive learning between the decoder's current known output information and the encoder's output. It calculates the similarity between the input sequence and the current sequence context by using the output from the previous layer as the query matrix, and the encoder's output as the key and value matrices, thereby improving the accuracy of predictions. The feedforward neural network consists of two fully connected layers and utilizes nonlinear activation functions for transformation, further extracting features and generating the final predicted vector representations. Additionally, each sub-layer is followed by residual connections and layer normalization, which help mitigate the vanishing gradient problem and enhance training efficiency. Through these modules, the decoder can fully leverage the contextual information provided by the encoder while generating the output sequence step-by-step, thereby improving understanding and prediction accuracy of the input sequence and significantly enhancing the performance and flexibility in stroke sequence prediction tasks.

The Masked Multi-Head Attention Mechanism (Vaswani et al., 2017) is primarily used within the self-attention mechanism of the decoder to prevent the model from focusing on positions in the output sequence that have not yet been generated during the inference process. When calculating attention scores, the masked multi-head attention mechanism adds a mask to obscure the attention values between the current and subsequent non-generated positions. This ensures that each step's output can only be predicted based on the sequence generated up to that point, thus avoiding interference from future information.

The attention mechanism first computes the dot product of the query, key, and value matrices. Then, it applies the masking matrix to the output, setting the attention weights of the future sequence (upper right part of the matrix) to zero. This ensures that the model can only attend to the input of the current and previous steps at each generation step.

The cross-attention mechanism in the decoder is based on the masked multi-head self-attention module. Its primary function is to attend to both the encoder's output and the output from the previous multi-head self-attention module of the decoder when generating each output. This approach effectively leverages the feature information learned by the encoder to better capture the relationship between the input sequence and the output being generated at the current step of the decoder. The decoder first computes a query matrix based on its input, then utilizes the encoder's output to create the key and value matrices. Using the cross-attention mechanism, the decoder acquires contextual information enriched with features learned from the encoder. This enables the model to consider the relationship between the input and the generated sequence at each step, resulting in more accurate predictions.

Position-aware gated fusion network

In the transformer model, the decoder maps the generated high-dimensional feature matrix to a vector of the same length as the number of prediction classes through a linear layer, used for the final prediction. However, players predict the next stroke strategy in a table tennis match based on their opponent's stroke strategy and the current state of the rally. Therefore, it is necessary to process the player and rally information individually and consider the importance of this information comprehensively. A traditional linear layer alone may transform this information into a high-dimensional feature matrix. Still, it does not sufficiently account for the varying importance of this information across different rallies.

To address this issue, we employed a position-aware gated fusion network (Wang et al. 2022a), which aims to more effectively fuse information from different sources while considering their importance in each rally. The Position-Aware Gated Fusion Network utilizes Gated Multimodal Units (Arevalo et al., 2017), enabling the model to dynamically adjust the weight of each information source during the fusion process, accurately reflecting the relative importance of each piece of information in the current context, as shown in Figure 2(c). Firstly, the rally context h_i^R output from the rally progress extractor and the context of both players h_i^A and h_i^B output from the player style extractor are mapped to the hidden vector space, as shown in Equations (11)-(13):

$$\tilde{h}_i^A = \delta_t\left(h_i^A W^A\right) \tag{11}$$

$$\tilde{h}_i^B = \delta_t\left(h_i^B W^B\right) \tag{12}$$

$$\tilde{h}_i^L = \delta_t\left(h_i^L W^L\right) \tag{13}$$

Here, $\delta_t(\cdot)$ represents the tanh activation function, and W^A ,W^B and W^R are learnable matrices. Next, the weight of each

 DOI 10.26773/mjssm.260306

context is calculated to convey its importance in the current rally, as shown in Equations (14)–(16):

$$\alpha^A = \delta_s\left([\,\tilde{h}_i^A,\, \tilde{h}_i^B,\, \tilde{h}_i^L\,]W^A\right) \tag{14}$$

$$\alpha^B = \delta_s\left([\,\tilde{h}_i^A,\, \tilde{h}_i^B,\, \tilde{h}_i^L\,]W^B\right) \tag{15}$$

$$\alpha^R = \delta_s\left([\,\tilde{h}_i^A,\, \tilde{h}_i^B,\, \tilde{h}_i^L\,]W^R\right) \tag{16}$$

Here, $\delta_s(\cdot)$ the sigmoid activation function is denoted, and W^A, W^B and W^R are learnable weight matrices. Finally, the projected vectors, information weights, and position weights are multiplied elementwise and summed up to obtain the final fused output, as shown in Equation (17):

$$z_i=\delta_s\,(\beta_i^A \otimes \alpha^A \otimes \tilde{h}_i^A+\beta_i^B \otimes \alpha^B \otimes \tilde{h}_i^B+\beta_i^R \otimes \alpha^R \otimes \tilde{h}_i^R) \tag{17}$$

In this equation, $\otimes$denotes element–wise multiplication, and β_i^A,β_i^B and β_i^R represent the learned position information weights. This method dynamically adjusts the strategy for fusing information between rallies and players. It introduces complex nonlinear transformations to enhance the prediction accuracy of the following stroke type and landing point. It offers significant advantages in multi-source information fusion. improving prediction accuracy.

Technique and Spot Prediction

In transformer models, the final layer is usually a fully connected layer for classification or generation tasks. The activation function of its output is softmax, which mainly converts the model output into a probability distribution for classification. In the table tennis stroke prediction model, the embedded vector p_i of the player hitting the -th ball is combined with the result z_i from the fusion network. The softmax function then converts each vector element into a probability value between [0,1]. The class with the highest probability is the model's prediction for the technique and landing point of the following stroke in the sequence, as shown in Equations (18)-(19):

$$\hat{t}_{(i+1)}=\mathrm{softmax}((z_i+p_{i+1})W^t \tag{18}$$

$$\hat{a}_{(i+1)}=\mathrm{softmax}((z_i+p_{i+1})W^a \tag{19}$$

Here, $\hat{t}_{(i+1)}$ represents the predicted probability distribution of the stroke technique for the next hit, and $\hat{a}_{(i+1)}$ denotes the predicted probability distribution of the landing point for the next hit. W^t and W^a are learnable matrices.

In deep learning models, an appropriate loss function is selected according to the task requirements to evaluate the model's learning effectiveness and establish measurement standards to quantify the difference between the ground truth and the predicted results. In the task of stroke forecasting in table tennis, the model mainly has two outputs: technique and spot, both of which are multi-class outputs. Cross-entropy loss, commonly used in deep learning for classification tasks, enables the comparison of the predicted probability distribution with the actual labels, accurately reflecting the prediction results, especially in multi-class classification problems.

When predicting the type of the next stroke, multi-class cross-entropy loss is used to measure the difference between the predicted stroke probability and the actual stroke type, which effectively handles classification problems with multiple categories, as shown in Equations (20)–(21):

$$L_{type}=-\Sigma_{r=1}^{|R|}\,\Sigma_{i=r+1}^{|S_r|}\,t_i\log_{(\hat{t}_i)} \tag{20}$$

$$L_{area}=-\Sigma_{r=1}^{|R|}\,\Sigma_{i=r+1}^{|S_r|}\,a_i\log_{(\hat{a}_i)} \tag{21}$$

The total loss function consists of two parts: L_{type}, which is the loss for stroke technique prediction, and L_{area}, which is the loss for landing point prediction. Here, S_r represents the input stroke sequence, and R represents all the rally sequences. The total loss value is obtained by summing these two losses, as shown in Equation (22):

$$L=L_{type}+L_{area} \tag{22}$$

Using this loss function, the model can simultaneously optimize the prediction for stroke type and landing point, improving its performance and accuracy in table tennis stroke prediction tasks. This approach effectively handles the complex dynamics in table tennis matches and helps us better understand and simulate player behavior patterns.

4. Experimental Results and Discussion
Experimental Environment

Due to the high computational demand of training and inference for deep learning networks, the experiment used a GPU to accelerate processing. The hardware setup consisted of an Intel(R) Core(TM) i9-11900K CPU @ at 3.50GHz, an NVIDIA GeForce RTX 4060 GPU, 64GB of memory, and the Windows 10 operating system. The development was mainly performed using Python and the PyTorch framework.

Datasets

The table tennis match data collected in this study were provided by the "Intellectual Tactical System in Competitive Table Tennis" system database (Liu et al., 2024). We selected matches between world-ranked table tennis player Lin Yun-Ju and his opponents as a case analysis, using a dataset comprising 23 matches, totaling 121 games, 2,225 rallies, and 10,517 shots, with an average of 4.7 shots per rally. In the shot prediction task, 80% of the rallies in each match were used as training data, while the remaining 20% were used for inference.

In table tennis, there are 18 stroke techniques. Fourteen of them are categorized as rally strokes: drive, counter-drive, smash, twist, fast drive, fast push, flick, pimple's long push, pimple's fast push, long push, drop shot, chop, block, and lob. The remaining four—traditional serve, hook serve, reverse pendulum serve, and squat serve—are classified as serve techniques. The table tennis table is divided into two halves by a net, with each side further segmented into nine landing zones: forehand short (1), middle short (2), backhand short (3), forehand mid-long (4), middle mid-long (5), backhand mid-long (6), forehand long (7), middle long (8), and backhand long (9). Stroke speed is categorized as slow, medium and fast, and spin types include topspin, backspin, no spin, sidespin-top, and sidespin-back. Hitting type is classified into forehand and backhand.

Evaluation Metrics

Accuracy typically refers to the ratio of correctly predicted samples to the total number of samples. However, in trajectory prediction, where multiple valid tactical options exist, simple accuracy may be too stringent a criterion for evaluation. Thus, we adopted Top-k Accuracy as the evaluation metric, as shown in Equation (23).

$$\text{Acc@}k = \frac{1}{N}\sum_{i=1}^{N} I\Big(y_i \in \hat{y}_i^{(k)}\Big) \tag{23}$$

Top-k Accuracy is commonly used in multi-class classification tasks, measuring whether the actual class appears in the model's top k predictions. This metric accommodates the tactical variability of table tennis and provides a more comprehensive assessment of model performance under various scenarios.

Comparisons

We applied the proposed prediction method to table tennis trajectories, enhancing ShuttleNet (Wang et al., 2022a) by incorporating speed and spin—two critical elements in the game of table tennis. These features significantly influence ball trajectories. We evaluated our model on datasets from one world-ranked Taiwanese player, Lin Yun-Ju.

Table 1 shows the results of Lin Yun-Ju's dataset comparing models with and without speed and spin. 'Spot' refers to landing prediction, and 'Technique' to stroke type prediction. Acc@1, Acc@3, and Acc@5 represent Top-1, Top-3, and Top-5 accuracy, respectively. ShuttleNet (Ori) refers to our baseline without speed/spin; ShuttleNet (Adv) includes them. Results show improved accuracy with speed/spin included. For landing prediction, improvements were minimal due to aggressive strategies favoring long strokes. Stroke prediction accuracy improved by about 5.4%.

Table 1. Evaluation Comparison of the Lin Yun-Ju Dataset

	Landing Area (Spot)			Stroke Type (Technique)		
Methods	Acc@1	Acc@3	Acc@5	Acc@1	Acc@3	Acc@5
ShuttleNet (Ori)	41.3%	94.4%	98.1%	51.4%	77.2%	91.8%
ShuttleNet (Adv)	41.4%	94.5%	98.2%	56.8%	77.3%	94.9%

We conducted experiments with observed sequence lengths τ = 2, 3, and 4 to investigate how the known stroke sequence length affects performance. Table 2 shows that with Lin Yun-Ju's dataset, Acc@1 was highest for τ = 2. Accuracy dropped for τ = 4, likely due to fewer long rallies in the dataset.

Table 2. Sequence Length Evaluation Comparison of the Lin Yun-Ju Dataset

Model	Observed Strokes(τ)	Landing Area			Stroke Type		
		Acc@1	Acc@3	Acc@5	Acc@1	Acc@3	Acc@5
ShuttleNet (adv)	2	42.8%	93.4%	97.6%	57.2%	76.6%	88.9%
	3	41.4%	94.5%	98.2%	56.8%	77.3%	91.8%
	4	40.9%	94.8%	98.1%	35.7%	77.9%	90.9%

According to a study by Hung et al. (2020), the adoption of the 40+ABS plastic ball in table tennis since 2017 has once again impacted the development ecosystem of the sport. The advantage of determining the outcome through serve and attack within the first three strokes has gradually diminished, while rally exchanges have increased. Therefore, enhancing the ability to predict tactical and technical performance in prolonged rallies would better align with the actual competitive scenarios of modern table tennis matches. From the data results of this elite player mentioned above, it can be observed that the optimal prediction stroke count for Lin is 2 strokes; his accuracy begins to decline from the fourth stroke onward. Upon review, this may be due to the relatively fewer data points for rallies extending beyond 4 strokes. This suggests that, although the study collected data from 23 matches for the player, it is still insufficient. To more accurately predict the patterns of prolonged rallies, future research should continue to track and expand the match data of players, thereby bringing greater value to the tactical applications of players.

We also examined sliding window sizes (k = 2, 3, 4) to predict the next stroke. Table 3 shows that for Lin Yun-Ju, stroke prediction was best at k = 2, while landing prediction peaked at k = 4. This may be due to his consistent rear-court play, where larger windows better capture player tendencies.

Table 3. Window Size Evaluation Comparison of the Lin Yun-Ju Dataset

	Window (k)	Landing Area			Stroke Type		
		Acc@1	Acc@3	Acc@5	Acc@1	Acc@3	Acc@5
ShuttleNet (adv)	2	40.9%	90.8%	97.4%	59.1%	91.0%	97.9%
	3	43.1%	92.5%	97.9%	58.4%	90.4%	96.4%
	4	45.0%	92.7%	97.7%	51.0%	87.3%	95.0%

The results of this study are also quite in line with actual competition scenarios. Regarding match tactics, elite players typically focus on short or mid-long serves, intending first to control the receiver, making it difficult for them to employ aggressive attack techniques. For the receiver, they might respond with drop shot, push, or back twist attacks. However, after the

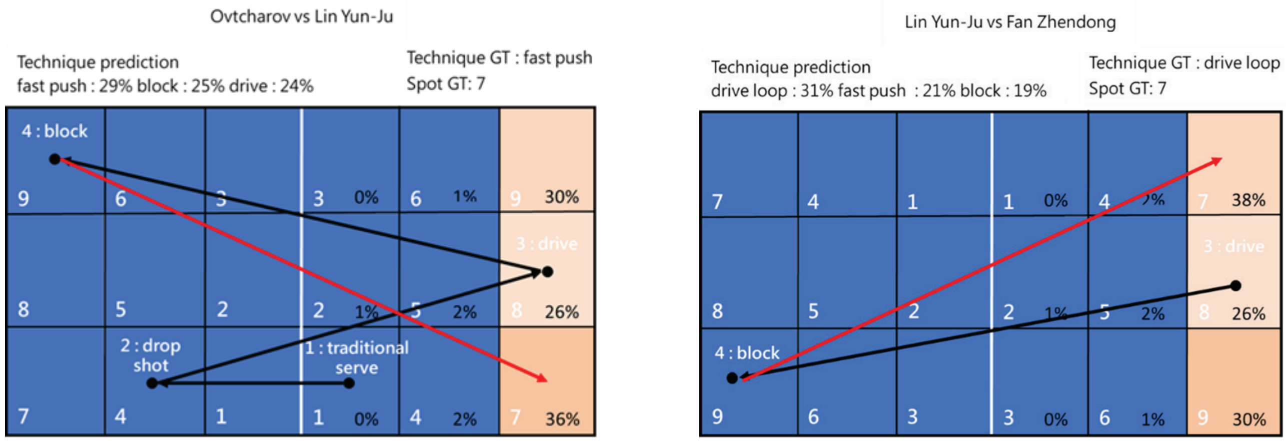
 DOI 10.26773/mjssm.260306

third stroke, both sides gradually reduce the use of control-focused techniques, leading to what is known as dynamic back-and-forth offensive exchanges. The hitting points for this style of play often occur in the rear court. From this phenomenon, we can generally infer that dynamic offensive exchanges begin by the third stroke in matches between top players.

Prediction Results

Our trained model predicts both the stroke type and landing zone based on a sequence of preceding strokes and visualizes the top three most probable outcomes. Figure 3 shows the prediction result of the proposed method. Figure 3(a) is used as a representative example. It illustrates a rally in which Ovtcharov serves to the “forehand short” area (zone 1), followed by a “drop shot” to zone 4 from Lin Yun-Ju, a “drive loop” to zone 8 by Ovtcharov, and a “block” to zone 9 by Lin Yun-Ju. The model predicted the next stroke as a“fast push” (red line) to zone 7 by Ovtcharov, which corresponded with the actual outcome. Figure 3 (b) shows the predictions for the fifth stroke, based on the data from the third and fourth strokes. In Figure 3 (b) (Lin Yun-Ju vs. Fan Zhendong), the predicted drive loop to zone 7 matched the real result

Figure 3. DPrediction result of the proposed method. (a) the match between Ovtcharov and Lin Yun-Ju in the 5th stroke: (b) the match between Fan Zhendong and Lin Yun-Ju in the 5th stroke



(a) (b)

From the above, we learned from the accuracy of predictions regarding the player Lin in the 5th stroke that our proposed method can almost accurately predict the techniques and landing positions they are likely to execute in the next shot. To ensure the reasonableness of the prediction results actual competition scenarios, after interpretation by two national-level table tennis coaches, it was noted that the predicted ball trajectory aligns well with the competition's offensive and defensive strategies, thereby proving the effectiveness of our proposed method. This outcome aligns with the findings of Wang et al. (2022), which suggest that players’ stroke selection strategies are influenced by both their overall playing style and the current rally situation. In real match scenarios, when two offensive players face each other, if one initiates an aggressive technique (e.g., a drive), the other may respond with a high-difficulty counterattack (e.g., a counter-drive) if they are confident. However, if the player's strategy leans more conservative, the likelihood of responding with a defensive block increases significantly.

Moreover, previous studies have indicated that the first five strokes are a crucial factor in determining the outcome of a match (Tsai et al., 2023). In this study, we found that Lin Yun-Ju's average number of strokes in the match was 4.7 strokes, corresponding to approximately five strokes per point. This result aligns closely with the rhythm of offensive players, making it significant to predict the trajectory of the fifth stroke. However, unlike previous prediction models, here we attempt to use only the player’s preceding two strokes (i.e., the 3rd and 4th strokes) as predictive information for the trajectory of the fifth stroke. The aim is to effectively predict the player’s subsequent tactical execution with minimal intelligence-gathering effort, thereby saving considerable data collection time to enhance intelligence-gathering and analysis efficiency. The results revealed that our model achieved excellent predictive performance even with only the preceding two strokes as input. For example, the system’s top-ranked prediction that Lin Yun-Ju would hit a “drive loop” to Fan Zhendong’s zone 7 on the fifth stroke was entirely accurate.

5. Conclusion

This paper employs a transformer-based architecture to propose a novel approach for predicting stroke techniques and landing points in table tennis. Converting match data into a stroke-level event stream improves input consistency and interpretability, facilitating more effective learning. The proposed dual-encoder-decoder framework captures rich contextual features from rallies and player profiles, while incorporating critical factors including the ball speed and spin, to model strategic behavior accurately. Landing point prediction is formulated as a classification task, and a sliding window mechanism is employed to capture short-term trajectory patterns, enabling fine-grained analysis of local tactical scenarios. Prediction results are visualized to assist coaches and players in strategic decision-making.

Although the current model is trained on a specific dataset, future work will enhance its generalizability by integrating vision-based technologies, such as pose estimation and ball tracking, to automate the annotation of technical actions. This will support the collection of more diverse and representative datasets, improving the model's applicability across a broader range of players and match contexts. Ultimately, this research advances intelligent sports analytics in table tennis, providing practical tools for performance evaluation and tactical planning.

References

Arevalo, J., Solorio, T., Montes-y-Gomez, M., & Gonzalez, F.A. (2017). Gated multimodal units for information fusion. *arXiv preprint*, arXiv:1702.01992.

Chiang, S., & Denes, G. (2023). Supervised learning for table tennis match prediction. *arXiv preprint* arXiv:2303.16776. https://doi.org/10.48550/arXiv.2303.16776

Decroos, T., Van Haaren, J., & Davis, J. (2018). Automatic discovery of tactics in spatio-temporal soccer match data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 223-232. (London, August 19-23, 2018). https://doi.org/10.1145/3219819.3219832

Giuliari, F., Hasan, I., Cristani, M., & Galasso, F. (2021). Transformer networks for trajectory forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy, 2021, pp. 10335-10342, doi: 10.1109/ICPR48806.2021.9412190.

Huang, W., Lu, M., Zeng, Y., Hu, M., & Xiao, Y. (2021). Technical and tactical diagnosis model of table tennis matches based on BP neural network. *BMC Sports Science, Medicine and Rehabilitation, 13*(1), 54. https://doi.org/10.1186/s13102-021-00281-y

Hung, C. C., Chou, T. C., & Hsu, M. H. (2020). The impact of 40+ competitive table tennis connecting techniques on the tactics of high-level athletes. *Quarterly of Chinese Physical Education Society of Physical Education, 34*(4), 273-286. DOI: 10.6223/qcpe.202012_34(4).0006

Kulkarni, K. M., & Shenoy, S. (2021). Table tennis stroke recognition using two-dimensional human pose estimation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Nashville, TN, USA, 2021, pp. 4571-4579, doi: 10.1109/CVPRW53098.2021.00515

Liu, J.-W., Hsu, M. H., Lai, C. L., & Wu, S. K. (2024). Using video analysis and artificial neural network to explore association rules and influence scenarios in elite table tennis matches. *The Journal of Supercomputing*, 80(4), 5472–5489. https://doi.org/10.1007/s11227-023-05399-6

Liu, J. W., Hsu, M. H., Lai, C. L., Wu, S. K. (2025). A novel framework for table tennis match analysis: combining 3S theory, video analysis, and big data analytics. *Journal of Mechanics in Medicine and Biology, 25*(5), 2540045. https://doi.org/10.1142/S0219519425400457

Tsai, A. L., Hsu, M. H., & Chiu, C.H. (2023). Analyzing the Impact of the Competitive Performance of Olympic Medal-Winning Table Tennis Players from the Perspective of Victory and Defeat. *Sports Coaching Science,71*, 35-50. DOI:10.6194/SCS.202309(71).0004

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000-6010.

Wang, W.-Y., Shuai, H.-H., Chang, K.-S., & Peng, W.-C. (2022a). Shuttlenet: Position-aware fusion of rally progress and player styles for stroke forecasting in badminton. In *Proceedings of the AAAI Conference on Artificial Intelligence, 36*(4), 4219-4227. https://doi.org/10.1609/aaai.v36i4.20341

Wang, W.-Y., Chan, T.-F., Peng, W.-C., Yang, H.-K., Wang, C.-C., &Fan, Y.-C. (2022b). How is the stroke? Inferring shot influence in badminton matches via long short-term dependencies. *ACM Transactions on Intelligent Systems and Technology, 14*(1), 1–22. https://doi.org/10.1145/3545311

Wang, W.-Y., Peng, W.-C., Wang, W., & Yu, P.S. (2023). ShuttleSHAP: A turn-based feature attribution approach for analyzing forecasting models in badminton. *arXiv preprint*, arXiv:2312.10942.

Wu, E., & Koike, H. (2020). Futurepong: Real-time table tennis trajectory forecasting using pose prediction network. In *Proceedings of the Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA, 25–30 April 2020; pp. 1–8. https://doi.org/10.1145/3334480.3382853

Zhang, Z., Xu, D., & Tan, M. (2010). Visual measurement and prediction of ball trajectory for table tennis robot. *IEEE Transactions on Instrumentation and Measurement, 59*(12), 3195–3205. https://doi.org/10.1109/TIM.2010.2061450