BST: Badminton Stroke-type Transformer for Skeleton-based Action Recognition in Racket Sports

Jing-Yuan Chang

National Tsing Hua University

va6lue@gapp.nthu.edu.tw

Abstract

Badminton, known for having the fastest ball speeds among all sports, presents significant challenges to the field of computer vision, including player identification, court line detection, shuttlecock trajectory tracking, and player stroke-type classification. In this paper, we introduce a novel video segmentation strategy to extract frames of each player's racket swing in a badminton broadcast match. These segmented frames are then processed by two existing models: one for Human Pose Estimation to obtain player skeletal joints, and the other for shuttlecock trajectory detection to extract shuttlecock trajectories. Leveraging these joints, trajectories, and player positions as inputs, we propose Badminton Stroke-type Transformer (BST) to classify player stroke-types in singles. To the best of our knowledge, experimental results demonstrate that our method outperforms the previous state-of-the-art on the largest publicly available badminton video dataset, ShuttleSet, which shows that effectively leveraging ball trajectory is likely to be a trend for racket sports action recognition.

1. Introduction

In recent years, the rapid development of deep learning has catalyzed significant progress in sports analysis [11, 13, 14, 17, 23, 25, 26], aiming to provide athletes with objective statistical data to refine their techniques and devise effective strategies for continuous improvement. Badminton, one of the world's most popular sports and known for having the fastest ball speeds among all sports, presents challenges for computer vision, including player identification, court line detection, shuttlecock trajectory tracking, and player stroke-type classification.

For the task of player stroke-type classification, models from the field of Human Action Recognition (HAR) can be leveraged. The domain of HAR, which focuses on identifying the actions performed by individuals in videos, has evolved from directly analyzing RGB bounding boxes in the frames [28] to first extracting skeletal joints from these individuals and then analyzing them. This extraction process is called Human Pose Estimation (HPE) [3, 4, 18, 29, 38], and the remaining process is known as Skeleton-based Action Recognition (SAR) [9, 10, 12, 20–22, 24, 33, 36, 37]. With this approach, we can effectively filter out extraneous background details and enable a concentrated focus on the nuances of human motion. Nevertheless, even with these advanced SAR models [9, 22, 33, 37] (see Sec. 2.2 for more details), achieving high accuracy in a broadcast badminton dataset faces several challenges. These models are primarily designed to recognize human actions in everyday scenarios, where movements often exhibit significant variation, for instance, the contrast between the static action of drinking water and the dynamic motion of running. However, players' racket-swinging actions can be categorized under a general "hitting" action in broader classifications, regardless of the specific stroke-type. Furthermore, these actions are both rapid and brief, making it even more difficult to distinguish between different strokes. Additionally, as noted in [21], although some models support multi-person input, they do not fully take into account the relationship between interacting players. These limitations underscore the challenge of distinguishing subtle differences in movements while utilizing limited information and determining which of the two players executed a particular stroke.

Fortunately, in badminton singles, the shuttlecock trajectory can serve as an interactive medium between the two players. To the best of our knowledge, TemPose [16] is the first model to incorporate shuttlecock trajectory information as an auxiliary input. However, we argue that the shuttlecock trajectory plays an even more crucial role in recognizing badminton stroke actions. Consider a hypothetical scenario where the shuttlecock is completely removed from a badminton match video, leaving only the two players performing air swings. From a human perspective, would it still be possible to accurately determine the type of stroke performed based solely on the players' movements? This highlights the indispensable role of shuttlecock trajectory information in a robust badminton stroke recognition

model. Therefore, we treat the shuttlecock trajectory as a primary input in our model. The main contributions of this paper are summarized as follows:

- We introduce a novel badminton video segmentation strategy to extract frames that have high relevance to the target badminton stroke swung by the player.
- We propose Badminton Stroke-type Transformer (BST) models that outperform the previous state-of-the-art on the largest publicly available badminton video dataset, ShuttleSet [32].
- We show that effectively leveraging shuttlecock trajectory information can significantly improve the performance of badminton stroke-type classification.

2. Related Work

2.1. Badminton Analysis System

Anurag Ghosh *et al.* proposed an end-to-end framework [13] for automatic attribute tagging and analysis in badminton videos. Building on this, Wei-Ting Chen *et al.* introduced a more advanced end-to-end analysis system [5], which incorporates a visually appealing user interface for enhanced usability. In the domain of stroke classification, Wei-Ta Chu and Samuel Situmeang developed a method [7] specifically targeting the bottom player in singles matches. Similarly, Yongkang Zhao proposed another stroke classification approach [35] that also focuses on the bottom player, utilizing deep learning techniques to improve accuracy. Magnus Ibh *et al.* introduced TemPose [16] that can classify top and bottom player strokes in singles matches. For serve detection, See Shin Yue *et al.* presented a specialized model [34] tailored for badminton. Further advancing this area, the TrackNet series [1, 6, 15, 30] primarily focuses on shuttlecock tracking and provides 2D trajectory predictions for downstream analysis systems. Paul Liu and Jui-Hsien Wang proposed MonoTrack [23], an end-to-end system that not only tracks the shuttlecock but also extends its functionality by simulating 3D trajectories from 2D trajectory data.

2.2. Skeleton-based Action Recognition (SAR)

In this field, the development of Graph Convolutional Network (GCN) [19]-based models has been as rapid and competitive as that of Transformer [31]-based models.

Sijie Yan *et al.* proposed ST-GCN [33], the first model to integrate GCN and Temporal Convolutional Network (TCN) [2] for SAR. At the time, it was considered a groundbreaking development.

Yuxuan Zhou *et al.* introduced BlockGCN [37], which employs BlockGC (a custom-designed GCN) along with a multi-scale TCN. Before applying these components, they incorporated Dynamic Topological Encoding. Within BlockGC, they utilized a Static Topological Encoding based on the hop distance between joints and a learnable adjacency matrix to provide the model with greater flexibility in learning. Remarkably, BlockGCN achieved state-of-the-art performance without relying on any attention mechanisms and was published at *CVPR* 2024.

SkateFormer [9], proposed by Jeonghyeok Do and Munchurl Kim, builds upon the Transformer architecture with several modifications. They integrated GCN, TCN, and their custom-designed Skate-MSA into the self-attention module. The core idea of Skate-MSA is to transform the input data into four different types, which were constructed based on their observations of human motion patterns. Each type is then processed separately. Their work was later published at *ECCV* 2024.

Hongda Liu *et al.* proposed ProtoGCN [22], which is also based on a custom GCN and a multi-scale TCN. Additionally, they incorporated Class-Specific Contrastive Loss to enhance model training. Their approach achieved strong performance on several well-known public datasets in SAR.

3. Method

In this section, we present the key components of our method. In Sec. 3.1, we present a video segmentation strategy which segments a match video into clips. Each clip contains two or three strokes. In Sec. 3.2, we introduce a data preprocessing work which produces player poses, positions and shuttlecock trajectories from stroke clips. In Sec. 3.3, we present a Transformer-based architecture which takes poses as inputs and produces classification results.

3.1. Video Segmentation Strategy

In this subsection, we explain how we segment a match video into stroke clips. There are many rallies in a match video. A rally is a sequence of strokes that start from a serve and ends when the shuttlecock falls on the court and a point is awarded to a player. A stroke refers to a badminton swing that hits the shuttlecock. We first denote every rally i that contains $n^{(i)}$ strokes j to be a set like

$$R_i = \{S_j^{(i)}\}_{j=1}^{n^{(i)}} = \{S_1^{(i)}, S_2^{(i)}, \dots, S_{n^{(i)}}^{(i)}\},$$

for $i = 1, 2, \dots, r$, where r is the total number of rallies in the match video, and $j = 1, 2, \dots, n^{(i)}$. We now segment every rally into stroke clips. Suppose a match video contain N frames denoted by $F_1, F_2, \dots, F_N$. Some of the frames in the match video are hit frames, which are the frames in which the shuttlecock is in contact with, or closest to, the racket during a stroke. Thus, every stroke $S_j^{(i)}$ in rally R_i has a one-to-one correspondence with the hit frame number $h_j^{(i)}$. A fixed-width clipping strategy similar to the approach used in [35] define the j-th stroke clip in rally R_i to be

$$C_j^{(i)} = \{F_{h_j^{(i)}-t}, F_{h_j^{(i)}-t+1}, \dots, F_{h_j^{(i)}+t}\}, \quad (1)$$

for $j = 1, 2, \ldots, n^{(i)}$ and $1 \leq i \leq r$. In Eq. (1), t is ideally half of the width of every hit frame number. However, it is a fixed parameter, and clearly, all stroke clips defined in Eq. (1) realistically contain at least one hit frame and possibly many hit frames.

Intuitively, consecutive strokes are highly correlated, and correlation between far-away strokes drops very rapidly. Thus, to increase the stroke classification accuracy, we would like to provide additional strokes to the classifier. On the other hand, we would not like to provide too many strokes to the classifier, as far-away strokes are much less correlated with, and can be even irrelevant to, the target stroke being classified. Our goal is for each stroke clip to include information only from the previous and next strokes swung by the opponent. Achieving this goal is difficult with a fixed-width clipping strategy using a fixed t. For this reason, we propose a variable-width clipping strategy as follows. First, define

$$\hat{h}_{j-1}^{(i)} = \begin{cases} h_{j-1}^{(i)}, & \text{if } j > 1 \\ h_j^{(i)} - t, & \text{if } j = 1, \end{cases}$$
$$\hat{h}_{j+1}^{(i)} = \begin{cases} h_{j+1}^{(i)} + \epsilon, & \text{if } j < n^{(i)} \\ h_j^{(i)} + t, & \text{if } j = n^{(i)}, \end{cases} \quad (2)$$

and a new stroke clip $\hat{C}_j^{(i)}$ in rally R_i to be

$$\hat{C}_j^{(i)} = \{F_{\hat{h}_{j-1}^{(i)}}, F_{\hat{h}_{j-1}^{(i)}+1}, \ldots, F_{\hat{h}_{j+1}^{(i)}}\}. \quad (3)$$

If the target stroke being classified is the first or last stroke in a rally, we set the lower or upper bound of the stroke clip to be the same value as used in Eq. (1). In Eq. (2), ϵ is a small parameter less than t. Looking at this strategy from the perspective of shuttlecock trajectory, it captures clean and complete trajectories in three stages: (1) the shuttlecock flying towards the target player's court, (2) the shuttlecock trajectory after being hit by the target player, and (3) the direction of the shuttlecock trajectory after the opponent's response. These stages respectively represent (1) the target player's observation and reaction phase, (2) the actual trajectory of the target stroke, and (3) the opponent's response pattern. This allows the model to not only judge based on (2) but also understand the target player's common strategies after going through (1) and infer the target stroke-type back from (3). The reason that the stroke information in (3) is less than that in (1), which means that ϵ cannot be set too large, is to preserve the characteristic of keeping the target hit frame close to the middle of the stroke clip under the ideal condition in Eq. (1).

We remark that the dataset (Sec. 4.1) we used to train our model contains hit frame and rally information. To classify a match video without hit frame and rally information, we can use a model called HitNet proposed in MonoTrack [23]

to detect hit frames. To detect rallies, we note that time intervals between successive rallies are in general much larger than the time differences between two typical successive hit frames in one rally. In practice, we also set some limits on $\hat{h}_{j-1}^{(i)}$ and $\hat{h}_{j+1}^{(i)}$ to prevent the model from accessing irrelevant information caused by remaining unclean data.

3.2. Model Inputs Extracted from Clips

Now, we have a set of stroke clips extracted from the match videos. The model inputs we need are player poses, shuttlecock trajectories, and player positions.

For the player poses and positions, we extract human poses from the clips using MMPose [8] toolbox at first. The models we utilized in MMPose are RTMPose [18] for 2D pose estimation and MotionBERT [38] for further 3D pose estimation. (We choose 2D poses for the input of the model. See Sec. 5 for more details.) However, there are more than two people, including the two players, spectators, and referees, in the clips. To extract the exact two player poses, we need the court lines to determine the player positions to filter out the irrelevant poses. Fortunately, the court lines information is available in our dataset (Sec. 4.1). Even if the court lines are not available, we can use the algorithm proposed in MonoTrack [23] to extract the court lines.

For the shuttlecock trajectories, we utilize the 2D shuttlecock positions generated by TrackNetV3[1] [1, 6] for each clip. Despite the existence of the 3D shuttlecock trajectory physical model MonoTrack [23], the first work to estimate 3D shuttlecock trajectories over 2D, precise 3D estimation remains challenging.

3.3. Badminton Stroke-type Transformer (BST)

The architecture of BST is derived from a Transformer-based model TemPose [16], which, to our knowledge, is the state-of-the-art badminton stroke classification model. A key distinction is that BST shifts its focus from player poses to shuttlecock trajectory information. Below, we describe several versions of BST, as follows: BST-0 serves as a prototype for the BST series. BST-0 takes player poses and shuttlecock trajectory information as its inputs, whereas BST-1, BST-2, and BST-3 incorporate additional player position information.

3.3.1. BST-0

Initially, the player poses[2] and shuttlecock positions are processed through a TCN module. After the outputs from this TCN are passed through a Transformer Encoder, individual class tokens with spatiotemporal representations are obtained, as shown in Fig. 1.

---

[1] There are two versions of TrackNetV3 developed by different authors. We choosed the one using attention for faster trajectory generating.

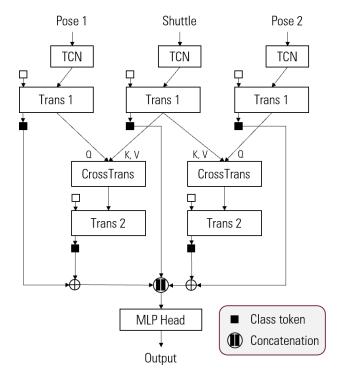[2] In practice: Top player pose is Pose 1; Bottom player pose is Pose 2.

Figure 1. Architecture of BST-0

The Transformer modules in this paper are more similar to that used in TemPose [16] than the original Transformer [31]. At the beginning, a learnable class token is prepended to the input sequence for summarization, and a learnable positional embedding is added to the input as well. We can formulate our Transformer Encoder as follows:

$$\tilde{X}^{(l)} = X^{(l-1)} + \text{MHSA}(\text{split}(\text{LN}(X^{(l-1)}))) \quad (4)$$

$$X^{(l)} = \tilde{X}^{(l)} + \text{FFN}(\text{LN}(\tilde{X}^{(l)})) \quad (5)$$

where $X^{(l)} \in \mathbb{R}^{S \times D_{model}}$ is the output to the l-th layer, S is the sequence length, D_{model} is the dimension used in the model, LN denotes the layer normalization, MHSA is the multi-head self-attention mechanism, and FFN is the feed-forward network. Before passing through the MHSA, the input $X^{(l)}$ which simplifies here to X is splitted into:

$$Q = [\, Q_1 \, \cdots \, Q_H \,] = XW_Q \in \mathbb{R}^{S \times (D_A \times H)} \quad (6)$$

$$K = [\, K_1 \, \cdots \, K_H \,] = XW_K \in \mathbb{R}^{S \times (D_A \times H)} \quad (7)$$

$$V = [\, V_1 \, \cdots \, V_H \,] = XW_V \in \mathbb{R}^{S \times (D_A \times H)} \quad (8)$$

where $W_Q, W_K, W_V \in \mathbb{R}^{D_{model} \times (D_A \times H)}$ are the weight matrices, D_A is the hyperparameter representing the dimension used in the attention modules, and H is the number of

the heads. Further in the MHSA, we formulate as follows:

$$\text{MHSA}(Q, K, V) = [\, \cdots \text{SA}(Q_h, K_h, V_h) \cdots \,] W_O \quad (9)$$

$$\text{SA}(Q_h, K_h, V_h) = softmax \left(\frac{Q_h(K_h)^{\text{T}}}{\sqrt{D_A}} \right) V_h \quad (10)$$

where $W_O \in \mathbb{R}^{(D_A \times H) \times D_{model}}$ represents the linear transformation from the dimenstion D_A to the original model dimension D_{model}.

Next, a key concept of our approach is ensuring that the model focuses on extracting critical information from the shuttlecock trajectory. To achieve this, we design a Cross Transformer layer with a Multi-Head Cross Attention (MHCA), which differs from MHSA in one key aspect: in the attention mechanism, K and V are derived from the shuttlecock trajectory latent, while Q is generated from the player pose latent or the player position latent, respectively, to perform cross-attention. As a result, the subsequent outputs follow two distinct pathways. One pathway is dedicated to the trajectory information related to the first player poses, while the other pathway focuses on the trajectory information related to the second player poses. Both pathways are fed into a second Transformer Encoder.

Finally, the model generates logits through an MLP (multi-layer perceptron) head, which are converted into probability values using the softmax function. The model is trained using cross-entropy loss.

3.3.2. Pose Postion Fusion (PPF)

In subsequent models, player position information was incorporated as an additional input to the model. X is preprocessed by a Pose Position Fusion (PPF) module to generate a fused representation of pose and position, as shown in Fig. 2. The output of the MLP in the PPF module multiplies back with the original poses and becomes the coefficients on the player poses.
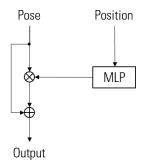

Figure 2. Pose Postion Fusion (PPF)

3.3.3. BST-1

In BST-1 shown in Fig. 3, a Clean Gate module is designed to refine the information derived from the shuttlecock trajectory by leveraging the interaction strength between the

two players and the shuttlecock. This helps the model filter out noise in the trajectory caused by the opponent's strokes. The Clean Gate module shown in Fig. 4 takes three inputs: (1) the class token representing shuttlecock information, (2) the fusion class token of the shuttlecock with the player one, and (3) the fusion class token of the shuttlecock with the player two. The input (2) and (3) are processed through min pooling and a MLP to generate values, which are then used to adjust the shuttlecock information class token, input (1), accordingly.
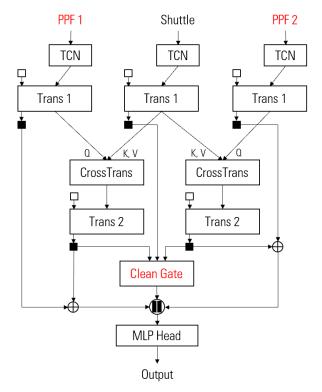

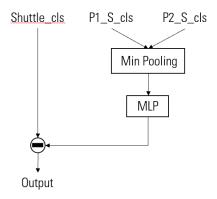Figure 3. Architecture of BST-1

Figure 4. Clean Gate

3.3.4. BST-2

In BST-2, shown in Fig. 5, we compare the cosine similarity between input (1) and (2) with that between input (1) and (3). The difference between these two values gives $\tilde{\alpha}$. Since cosine similarity ranges from -1 to 1, $\tilde{\alpha} \in [-2, 2]$. We normalize it using $\alpha = (\tilde{\alpha} + 2)/4$ so that $\alpha \in [0, 1]$. α denotes the coefficient on the target player information, so $1 - \alpha$ is the coefficient on his/her opponent's information. This ensures that the player with a stronger correlation to the overall shuttlecock trajectory exerts a greater influence on the input of the MLP head.
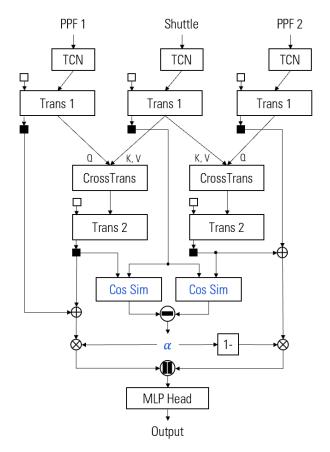

Figure 5. Architecture of BST-2

3.3.5. BST-3

BST-3, as shown in Fig. 6, combines the benefits of BST-1, which focuses on purifying shuttlecock information, and BST-2, which emphasizes the information derived from the target player.

4. Experiments

In this section, we present the dataset used in our experiments and compare our results with those obtained by several existing methods, including ST-GCN [33], Block-
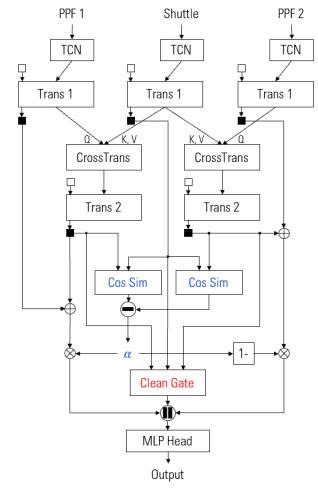
Figure 6. Architecture of BST-3

GCN [37], SkateFormer [9], ProtoGCN [22], and Tem-Pose [16].

4.1. Dataset

ShuttleSet[3] [32], the largest publicly available badminton video dataset, contains 104 sets, 3,685 rallies, and 36,492 strokes across 44 matches played between 2018 and 2021, featuring 27 top-ranking male and female singles players. After excluding incorrectly labeled and problematic data, a total of 40 matches, 33429 strokes remained for analysis. We set 30 matches for training, 5 matches for validation, and 5 matches for testing.

The original ShuttleSet dataset consists of 19 distinct stroke categories, including a "None" type. One category, driven flight, containing fewer than 50 strokes across the entire dataset was merged into the "None" category by us. Additionally, since the dataset does not differentiate between strokes played by the top or bottom player within each cat-

---

---

egory (except for "None"), the number of stroke categories (except for "None") was doubled. This adjustment resulted in a total of 35 categories, including "None".

4.2. Results

Tab. 1 demonstrates that even the most advanced SAR models struggle to achieve satisfactory accuracy. Without incorporating additional inputs, such as shuttlecock trajectories and player positions, significant performance improvements are difficult to attain. However, integrating such information is not straightforward for these models. This highlights the need for specialized models like TemPose [16], which are specifically designed to handle these additional inputs.

Initially, by focusing solely on player position fusion, TemPose-PF achieves results that surpass those of TemPose-V. The reason for this improvement is that player positions imply some information about the shuttlecock trajectory, that is, the players are always chasing the shuttlecock during a rally. On the other hand, integrating only shuttlecock positions, TemPose-SF leads to a substantial improvement of over two percentage points. Furthermore, combining both types of fusion further enhances performance. Notably, our architecture BST, achieves a modest improvement over TemPose when using a fixed-width clipping strategy.

Finally, the incorporation of our segmentation strategy results in even greater performance gains across all BST models, significantly exceeding the performance of TemPose, as shown in Tab. 2.

5. Discussion

5.1. Classifying Difficulty on ShuttleSet

Although our model achieved the state-of-the-art performance on ShuttleSet [32], the accuracy of it was still not high enough. We suspect that this is primarily due to the excessive granularity of certain categories. The normalized confusion matrices of BST-0 are shown in Fig. 7. The 0th to 16th strokes are executed by the top player, the 17th to 33rd strokes are hit by the bottom player, and the 34th stroke belongs to the "None" category. Observing on both confusion matrices, the model struggles to differentiate between similar stroke-types, such as the 2nd and 3rd strokes (top smash and top wrist smash) and the 19th and 20th strokes (bottom smash and bottom wrist smash). A closer inspection reveals that the model often misclassifies the 1st stroke (top return net) as the 13th stroke (top defensive return drive) that shown in the left matrix. Additionally viewing the diagonal elements in the right matrix, the 13th stroke (top defensive return drive) has the worst recall. These results suggest that the model has difficulty distinguishing between subtle variations in the strokes with similar hitting characteristics. We observed the same phenomenon when employing other

Model	Publication	Modality	SP	PP	Acc	Macro-F1	Acc-2	NumParams
ST-GCN [33]	AAAI 2018	J	×	×	0.728	0.656	0.897	3.08M
BlockGCN [37]	CVPR 2024	J	×	×	0.715	0.627	0.886	1.50M
SkateFormer [9]	ECCV 2024	J	×	×	0.712	0.627	0.859	2.38M
ProtoGCN [22]	(arXiv 2024)	J	×	×	0.723	0.637	0.897	4.11M
TemPose-V [16]	CVPRW 2023	J	×	×	0.720	0.642	0.892	1.62M
TemPose-V [16]	CVPRW 2023	J+B	×	×	0.730	0.645	0.895	1.62M
TemPose-PF		J+B	×	✓	0.738	0.677	0.905	1.68M
TemPose-SF		J+B	✓	×	0.7602	**0.6922**	0.9196	1.65M
BST-0		J+B	✓	×	**0.7626**	0.6908	**0.9206**	1.81M
TemPose-TF [16]	CVPRW 2023	J+B	✓	✓	0.7683	0.7027	0.9242	1.71M
BST-1		J+B	✓	✓	0.7681	0.7004	0.9252	1.85M
BST-2		J+B	✓	✓	0.7689	0.7023	0.9260	1.79M
BST-3		J+B	✓	✓	**0.7695**	**0.7043**	**0.9267**	1.85M

Table 1. Model comparison on the testing set without our segmentation strategy. J and B in the modality denote the players' joints and bones. SP and PP represent additional inputs of shuttlecock and player positions. TemPose-PF and TemPose-SF are intermediate products derived by us from the original model, positioned between TemPose-V and TemPose-TF. The suffix "F" denotes the fusion of the additional inputs. Acc-2 denotes top-2 accuracy.

Model	Modality	SP	PP	Acc	Macro-F1	Acc-2
TemPose-SF	J+B	✓	×	0.7534	0.6833	0.9194
BST-0	J+B	✓	×	**0.7655**	**0.6976**	**0.9306**
TemPose-TF [16]	J+B	✓	✓	0.7580	0.6902	0.9214
BST-1	J+B	✓	✓	0.7704	0.7026	**0.9355**
BST-2	J+B	✓	✓	0.7687	0.7000	0.9312
BST-3	J+B	✓	✓	**0.7710**	**0.7042**	0.9340

Table 2. Model comparison on the testing set after our segmentation strategy.

models as well.

5.2. Player 2D Joints vs. 3D Joints

While we explored using 3D player joints as input to our model, the performance was inferior to that achieved with 2D joints. We attribute this to the fact that the HPE model is trained on a general HPE dataset, encompassing a broader range of poses than those specific to badminton players. Additionally, the 3D joints are estimated from the 2D joints, which may further contribute to a bias towards general human poses, rather than the specialized poses of badminton players. We can see an example of this in Fig. 8. This issue would mislead the model to predict the wrong action, since the 3D pose is not accurate enough to represent the player's pose in the badminton game, as shown in Tab. 3.

5.3. Limitations and Future Work

Our method has an obvious limitation: it relies on accurate hit frame detection and shuttlecock trajectory tracking mod-

els. Fortunately, these issues are expected to be resolved in the future, similar to how the transition from HAR to SAR was facilitated by advanced HPE models, which enabled SAR to perform more effectively. Moreover, the existing 2D shuttlecock trajectory tracking models are quite mature, and we believe that there will be more 3D tracking models in the future. Although this paper primarily focuses on stroke classification in singles badminton, the potential applicability of our method to other racket sports, such as tennis and table tennis, or even doubles, is worth exploring.

6. Conclusion

In this paper, we first introduce a novel segmentation strategy that segments a match video into clips containing two to three badminton strokes. We propose BST for badminton stroke classification. Through experiments, we show that our method outperforms existing state-of-the-art methods, highlighting the importance of the shuttlecock information.
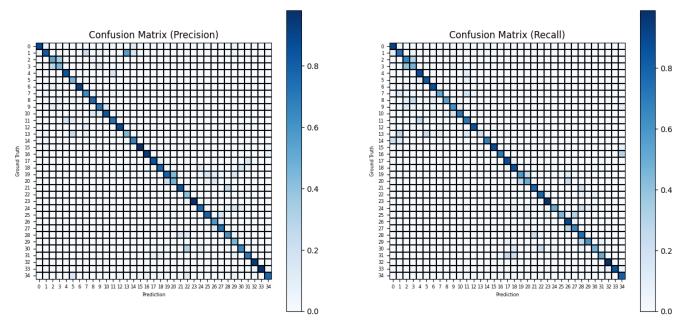
Figure 7. Normalized confusion matrices of BST-0 on our dataset. The sum of the elements in each column in the left matrix is equal to 1, and the sum of the elements in each row in the right matrix is equal to 1.

(a) 2D pose   (b) Rear view of the 3D pose   (c) Front view of the 3D pose   (d) Top view of the 3D pose
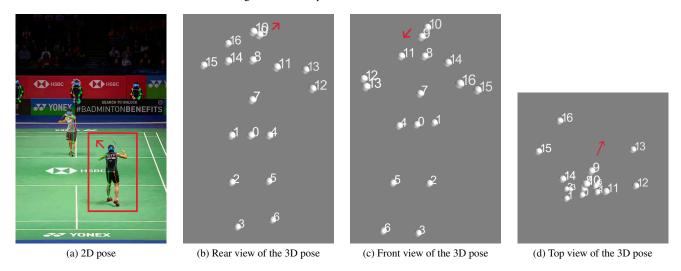
Figure 8. Example of a 2D pose vs. 3D pose in a badminton video frame. The ninth joint represents the human nose. The red arrows indicate the directions the bottom player was facing. We can see the bottom player was facing her opponent in fact, but the 3D pose exhibits a clear error in the facing direction. We suspect that this 3D model will often assume that if both forearms are pointing roughly forward, it will lead the model to assume that the face is also pointing in that direction. (The visualizations in (b), (c), and (d) are generated by the Mayavi [27] python package.)

Model	Accuracy	Macro-F1	Accuracy-2
TemPose-SF	**0.7602** / 0.7458	**0.6922** / 0.6696	**0.9196** / 0.9066
BST-0	**0.7626** / 0.7498	**0.6908** / 0.6768	**0.9206** / 0.9080
TemPose-TF [16]	**0.7683** / 0.7563	**0.7027** / 0.6852	**0.9242** / 0.9164
BST-3	**0.7695** / 0.7560	**0.7043** / 0.6806	**0.9267** / 0.9155

Table 3. Model with 2D vs. 3D poses comparison on the testing set without our segmentation strategy. The value on the left side of the slash is the result of the model with 2D poses, and the right side is the result of the model with 3D poses.

References

[1] Alenzenx. Tracknetv3: Beyond tracknetv2 ,and "first" tracknet using attention. https://github.com/alenzenx/TrackNetV3, 2023. 2, 3

[2] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018. 2

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 1302–1310, 2017. 1

[4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE TPAMI*, 43(1): 172–186, 2021. 1

[5] Wei-Ting Chen, Hsiang-Yun Wu, Yun-An Shih, Chih-Chuan Wang, and Yu-Shuen Wang. Exploration of player behaviours from broadcast badminton videos. *Computer Graphics Forum*, 42(6):e14786, 2023. 2

[6] Yu-Jou Chen and Yu-Shuen Wang. Tracknetv3: Enhancing shuttlecock tracking with augmentations and trajectory rectification. In *ACM International Conference on Multimedia in Asia*, New York, NY, USA, 2024. Association for Computing Machinery. 2, 3

[7] Wei-Ta Chu and Samuel Situmeang. Badminton video analysis based on spatiotemporal and stroke features. In *ACM on International Conference on Multimedia Retrieval (ICMR)*, pages 448–451, New York, NY, USA, 2017. Association for Computing Machinery. 2

[8] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 3

[9] Jeonghyeok Do and Munchurl Kim. Skateformer: skeletal-temporal transformer for human action recognition. In *ECCV*, pages 401–420. Springer, 2025. 1, 2, 6, 7

[10] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Dg-stgcn: Dynamic spatial-temporal modeling for skeleton-based action recognition. *arXiv e-prints*, art. arXiv:2210.05895, 2022. 1

[11] Javier Fernández and Luke Bornn. Soccermap: A deep learning architecture for visually-interpretable analysis in soccer. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track*, pages 491–506, Cham, 2021. Springer International Publishing. 1

[12] Zhimin Gao, Peitao Wang, Pei Lv, Xiaoheng Jiang, Qidong Liu, Pichao Wang, Mingliang Xu, and Wanqing Li. Focal and global spatial-temporal transformer for skeleton-based action recognition. In *ACCV*, pages 155–171, Berlin, Heidelberg, 2022. Springer-Verlag. 1

[13] Anurag Ghosh, Suriya Singh, and C. V. Jawahar. Towards structured analysis of broadcast badminton videos. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 296–304, 2018. 1, 2

[14] Mei-Ling Huang and Yun-Zhi Li. Use of machine learning and deep learning to predict the outcomes of major league baseball matches. *Applied Sciences*, 11(10), 2021. 1

[15] Yu-Chuan Huang, I-No Liao, Ching-Hsuan Chen, Tsì-Uí İk, and Wen-Chih Peng. Tracknet: A deep learning network for tracking high-speed and tiny objects in sports applications. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8, 2019. 2

[16] Magnus Ibh, Stella Grasshof, Dan Witzner, and Pascal Madeleine. Tempose: a new skeleton-based transformer model designed for fine-grained motion recognition in badminton. In *CVPRW*, pages 5199–5208, 2023. 1, 2, 3, 4, 6, 7, 8

[17] Kan Jiang, Jiayu Li, Zhaoyu Liu, and Chen Dong. Court detection using masked perspective fields network. In *IEEE 28th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 342–345, 2023. 1

[18] Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose, 2023. 1, 3

[19] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. 2

[20] Jungho Lee, Minhyeok Lee, Dogyoon Lee, and Sangyoun Lee. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In *ICCV*, pages 10444–10453, 2023. 1

[21] Meng Li, Yaqi Wu, Qiumei Sun, and Weifeng Yang. Two-stream proximity graph transformer for skeletal person-person interaction recognition with statistical information. *IEEE Access*, 12:193091–193100, 2024. 1

[22] Hongda Liu, Yunfan Liu, Min Ren, Hao Wang, Yunlong Wang, and Zhenan Sun. Revealing key details to see differences: A novel prototypical perspective for skeleton-based action recognition. *arXiv preprint arXiv:2411.18941*, 2024. 1, 2, 6, 7

[23] Paul Liu and Jui-Hsien Wang. Monotrack: Shuttle trajectory reconstruction from monocular badminton video. In *CVPRW*, pages 3512–3521, 2022. 1, 2, 3

[24] Woomin Myung, Nan Su, Jing-Hao Xue, and Guijin Wang. Degcn: Deformable graph convolutional networks for skeleton-based action recognition. *IEEE TIP*, 33:2477–2490, 2024. 1

[25] Masato Nakai, Yoshihiko Tsunoda, Hisashi Hayashi, and Hideki Murakoshi. Prediction of basketball free throw shooting by openpose. In *New Frontiers in Artificial Intelligence*, pages 435–446, Cham, 2019. Springer International Publishing. 1

[26] Nadav Oved, Amir Feder, and Roi Reichart. Predicting in-game actions from interviews of nba players. *Computational Linguistics*, 46(3):667–712, 2020. 1

[27] Prabhu Ramachandran and Gaël Varoquaux. Mayavi: a package for 3d visualization of scientific data. *CoRR*, abs/1010.4891, 2010. 8

[28] Amir Shahroudy, Tian-Tsong Ng, Yihong Gong, and Gang Wang. Deep multimodal feature analysis for action recognition in rgb+d videos. *IEEE TPAMI*, 40(05):1045–1058, 2018. 1

[29] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 1

[30] Nien-En Sun, Yu-Ching Lin, Shao-Ping Chuang, Tzu-Han Hsu, Dung-Ru Yu, Ho-Yi Chung, and Tsì-Uí İk. Tracknetv2: Efficient shuttlecock tracking network. In *International Conference on Pervasive Artificial Intelligence (ICPAI)*, pages 86–91, 2020. 2

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*. Curran Associates, Inc., 2017. 2, 4

[32] Wei-Yao Wang, Yung-Chang Huang, Tsi-Ui Ik, and Wen-Chih Peng. Shuttleset: A human-annotated stroke-level singles dataset for badminton tactical analysis. *CoRR*, abs/2306.04948, 2023. 2, 6

[33] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *AAAI*, 32(1), 2018. 1, 2, 5, 7

[34] See Shin Yue, Raveendran Paramesran, and Ganesh Krishnasamy. Ready-to-serve detection in badminton videos. In *International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–5, 2024. 2

[35] Yongkang Zhao. Automatic shuttlecock motion recognition using deep learning. *IEEE Access*, 11:111281–111291, 2023. 2

[36] Yuxuan Zhou, Zhi-Qi Cheng, Chao Li, Yifeng Geng, Xuansong Xie, and Margret Keuper. Hypergraph transformer for skeleton-based action recognition. *arXiv preprint arXiv:2211.09590*, 2022. 1

[37] Yuxuan Zhou, Xudong Yan, Zhi-Qi Cheng, Yan Yan, Qi Dai, and Xian-Sheng Hua. Blockgcn: Redefining topology awareness for skeleton-based action recognition. In *CVPR*, 2024. 1, 2, 6, 7

[38] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Motionbert: A unified perspective on learning human motion representations. In *ICCV*, 2023. 1, 3