T3Set: A Multimodal Dataset with Targeted Suggestions for LLM-based Virtual Coach in Table Tennis Training

Ji Ma
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, Zhejiang, China
zjumaji@zju.edu.cn

Jiale Wu
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, Zhejiang, China
jialewu2022@zju.edu.cn

Haoyu Wang
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, Zhejiang, China
wang.haoyu@zju.edu.cn

Yanze Zhang
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, Zhejiang, China
yanzezhang@zju.edu.cn

Xiao Xie
Department of Sports Science
Zhejiang University
Hangzhou, Zhejiang, China
xxie@zju.edu.cn

Zheng Zhou
Department of Sports Science
Zhejiang University
Hangzhou, Zhejiang, China
zheng.zhou@zju.edu.cn

Hui Zhang
Department of Sports Science
Zhejiang University
Hangzhou, Zhejiang, China
zhang_hui@zju.edu.cn

Jiachen Wang*
Department of Sports Science
Zhejiang University
Hangzhou, Zhejiang, China
wangjiachen@zju.edu.cn

Yingcai Wu
State Key Lab of CAD&CG
Zhejiang University
Hangzhou, Zhejiang, China
ycwu@zju.edu.cn

Abstract

Coaching is critical for learning table tennis skills. However, amateur table tennis players often lack access to professional coaches due to high costs and a limited number of coaches. While recent multimodal large language models show promise as virtual coaches, most of the existing approaches merely rely on video analysis, which is not comprehensive enough. In table tennis, many important kinematic details (e.g., strength, acceleration) cannot be captured by videos. They can only be tracked using sensors. To address this gap, we present **T3Set** (**T**able **T**ennis **T**raining Set), a multimodal dataset that synchronizes inertial measurement unit (IMU) data from sensors mounted on 32 players' rackets with video recordings. The sensor data has 16 dimensions and a sample rate of 100Hz. This dataset covers 7 fundamental techniques across 380 training rounds, totaling 8655 annotated strokes, with 8395 targeted suggestions from coaches. The key features of T3Set include (1) temporal alignment between sensor data, video data, and text data. (2) high-quality targeted suggestions which are consistent with predefined suggestion taxonomy. Based on T3Set, we propose a novel two-stage framework that effectively integrates motion perception with generative reasoning as a virtual coach. Our method quantitatively outperforms baseline methods. The dataset, code, and documentation are available at https://github.com/jima-cs/T3Set.

*Jiachen Wang is the corresponding author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737407

CCS Concepts

• **Information systems** → *Multimedia databases*; • **Applied computing**;

Keywords

table tennis; multimodal dataset; virtual coach

ACM Reference Format:
Ji Ma, Jiale Wu, Haoyu Wang, Yanze Zhang, Xiao Xie, Zheng Zhou, Hui Zhang, Jiachen Wang, and Yingcai Wu. 2025. T3Set: A Multimodal Dataset with Targeted Suggestions for LLM-based Virtual Coach in Table Tennis Training. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737407

KDD Availability Link:
The dataset of this paper has been made publicly available at https://doi.org/10.5281/zenodo.15516143. The code of the model and scripts has been made publicly available at https://doi.org/10.5281/zenodo.15522785.

1 Introduction

Coaches are vital for learning table tennis skills. They can effectively identify the shortcomings during the training process and provide targeted suggestions for improvement. However, due to the high costs and the limited number of coaches, general amateurs can hardly hire experienced coaches for training. A virtual coach provides an opportunity for amateurs to obtain high-quality suggestions with ease. With the development of large language models (LLMs), various virtual coaches have been established by leveraging the excellent dialogue and reasoning capabilities of LLMs. For example, Zhang et al. [56] have explored generating training suggestions for fitness, and Zhang et al. [57] developed an agent to conduct

intelligent assessments of racket sports videos through video comprehension. While these virtual coaches can provide suggestions, they are not comprehensive when applied to table tennis.

Table tennis training emphasizes precise body control and minute movement changes, which can be reflected by important kinematic information such as grip posture, racket angle, and stroke strength. Such kinematic information should be comprehensively captured by both cameras and sensors. However, existing virtual coaches give suggestions only based on videos and conversations or even conversations alone, missing the focus on sensor data. Intricate and kinematic details, such as strength and racket angle [5, 22] are not captured and analyzed, leading to inaccurate or off-target suggestions[16, 27]. A virtual coach that can fuse video data and sensor data for accurate and targeted suggestion generation is needed for table tennis training.

Constructing such a virtual coach demands a high-quality dataset including video data, sensor data, and well-annotated suggestion data in table tennis for model training. However, to our knowledge, no existing dataset can satisfy this requirement. To fill this blank and facilitate the research on virtual coaches, we propose T3Set, the first multimodal dataset for virtual coach construction in **T**able **T**ennis **T**raining. We mainly solved two challenges when constructing T3Set. The first is **accurate data alignment**. Sensor data, which tracks player movements, must align with the video view, and coach feedback must be linked to specific moments in the training session. However, these three types of data are recorded at various frequencies (e.g., sensor: 100HZ, camera: 60fps, suggestion: 6-7/minute). In addition, the sensor and the camera use different time systems. They may not share the same clock or time reference, and the differences between the two clocks or time references may change each time when recording data. All these issues make it difficult to match data from all sources accurately.

The second is **quantifiable evaluation**. Existing evaluation metrics for virtual coaches[56, 57] focus on subjective scoring. They hire coaches/players to score each suggestion generated by the virtual coach to evaluate the performance of suggestion generation. While this method can provide an approximate assessment of a virtual coach's ability, it is not scalable enough. Ma et al. [27] define several error types existing in the training process and let the virtual coach judge whether specific errors exist in a particular training process. This method is scalable and can evaluate the virtual coach's performance quantitatively by computing the error detection accuracy. However, it cannot directly evaluate the quality of the suggestions generated by the virtual coach. A quantifiable evaluation method should be proposed.

To address the first challenge, we developed a three-phase alignment process. First, we segmented the videos and sensor signals into clips, each of which contained a complete training round. Second, we detected stroke (each hit in table tennis is called a *stroke*) sequences from both kinds of clips, respectively. We designed an alignment algorithm based on linear regression to align the video clips and corresponding sensor clips. We manually align dirty alignment cases by checking the videos. Third, given that suggestions were recorded in audio during video checking, we converted the audio into text, and based on the timestamps of the audio, we manually aligned the textual suggestions with the other two modalities.

To address the second challenge, we analyzed all suggestions and summarized a suggestion taxonomy. Based on the taxonomy, we converted each suggestion into a structured format to support the quantitative assessment of the suggestion quality.

We further designed a two-stage framework, SenseCoach, to validate the usability of T3Set. At Stage 1, we proposed a fusion model incorporating cross-key attention mechanisms to integrate video data and sensor data and generate suggestion keys. At stage 2, we input suggestion keys into LLM for reasoning and suggestion generation. To our knowledge, We are the first to establish a high-quality dataset involving aligned video data, sensor data, and suggestion data in table tennis training scenarios. Based on the dataset, we explored LLM's ability to generate targeted suggestions. We hope that our dataset and SenseCoach can inspire research on virtual coach systems. We summarize our contributions as follows:

- We establish a multimodal dataset with aligned video-sensor-text data in table tennis training.
- We propose a suggestion taxonomy to evaluate the quality of suggestions quantitatively.
- Using this dataset, we proposed a two-stage framework, SenseCoach, to generate targeted suggestions.
- We conducted experiments to evaluate the performance of our framework.

2 Related Work

2.1 LLM-Based Virtual Coach

Before the LLM era, early works like swimming coaches [40], running assistants [44], and fitness trainers [54] were constructed based on rule-based adaptation mechanisms and predefined content templates. These systems established foundational motion analysis capabilities yet lacked contextual understanding and dynamic reasoning. LLM has had a huge impact on domains like sports[7, 26] and sparked a lot of research on text[34, 45], interaction[49, 53], and multimodality[15, 55], etc. LLM With the advent of LLMs, text-centric systems emerged that leveraged conversational interfaces [47] and RAG-enhanced knowledge injection [10], though their inability to process physical motion data. This limitation motivated the integration of visual understanding, where multimodal LLMs reveal progress in fitness coaching [56], racket sports analysis [57], and badminton analysis [8]. Recent work by Ma et al.[27] defined several error types and utilized the LLM to detect specific errors and give error analysis. While capable of visual understanding, these works ignore sensor data as many kinematic details can only be captured by sensors and other equipment. Moreover, they rely on expert scoring and cannot directly conduct a quantitative evaluation of the quality and feasibility of the suggestions. Though Ashutosh et al. [2] addresses actionability with pose and video data, it still misses sensor data. Therefore, we propose a method that combines video data and sensor data, establishing a new method for generating targeted suggestions as an LLM-based virtual coach.

2.2 Multimodal Sports Datasets

The development of sports analytics facilitates the construction of various multimodal sports datasets. The survey by Xia et al. [52] summarized existing multimodal datasets for sports. However,

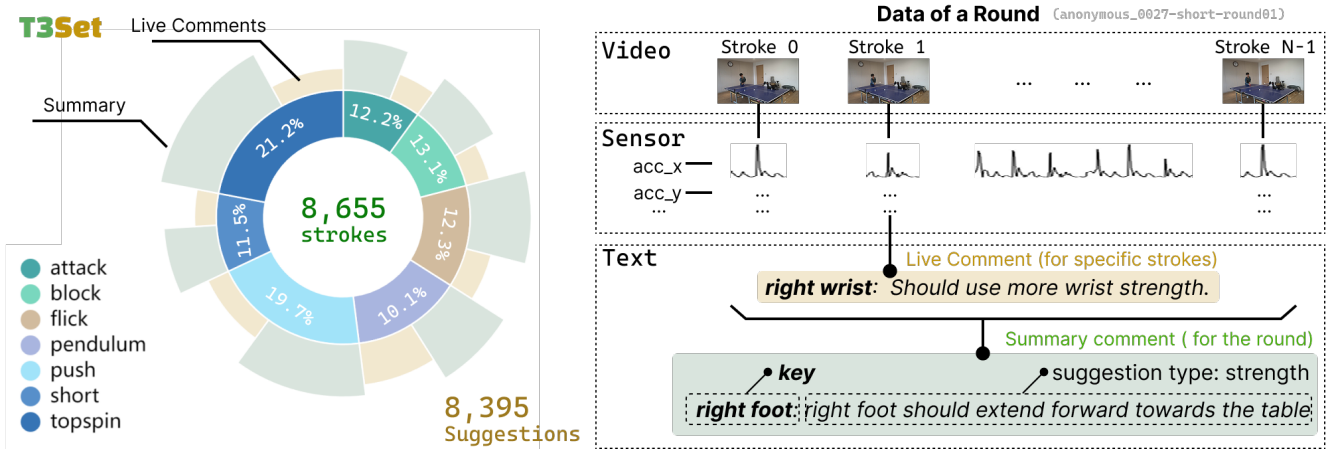

Figure 1: T3Set comprises data from 32 amateur players, encompassing 7 stroke techniques, 380 rounds, and includes a total of 8395 strokes and 8655 suggestions. According to the statistics, the 7 techniques collectively account for 8773 strokes. On average, each round contains 23.5 suggestion sentences and involves 8.18 keys.

according to the survey, many existing datasets [43, 51] predominantly serve **conventional analytical tasks** such as tactic analysis [41, 48], match simulation [50], and event prediction [42], which cannot support the construction of virtual coaching due to the lack of well-annotated suggestion data.

Some multimodal datasets have rich annotations for game summarization, such as MSMO[35], and so on [23, 30, 35, 39]. And some datasets hasve video-question-answer triplets for video understanding such as Sports-QA[24], EIGD [4], and X-VARS[21]. However, they rarely involve sensor data, which is important for kinematic details. As noted in a survey[16], aligning sensor streams (accelerometers, gyroscopes) with coaching knowledge remains unresolved.

Ego4D [18] and Ego-Exo4D [19] provide egocentric action data from Meta Aria glasses. The characteristics of the collection devices allow them to directly include aligned video data and sensor data, and their annotations can somewhat support virtual coach construction, but the data collected by this type of equipment is not suitable for racket sports dynamics. For table tennis datasets, there are only traditional video data [3, 58]or sensor data[9] available. So, we proposed a multimodal dataset with synchronized multimodal streams (video+sensor+suggestion text) for virtual coach construction in table tennis.

3 T3Set: Table Tennis Training Dataset

In this chapter, we will introduce the structures, categories, and features of T3Set, as well as the processing steps and alignment algorithm.

3.1 Background about table tennis training

In table tennis, hitting the ball once is referred to as a *stroke*. Each stroke features various techniques that lead to various directions and speeds of spin. Mastery of these techniques is crucial to performance in matches. Therefore, technique training is one of the most important components of table tennis training. Multi-ball training[20] is a popular method for technique training, where a coach/ball machine continuously feeds balls to the player, and the player practices consistent techniques to develop a stable technique performance. T3Set was constructed based on the multi-ball training scenario.

3.2 Dataset Overview

As shown in Figure 1, T3Set includes video data, sensor data, and text data of coaches' suggestions. The detailed structure and description of T3Set are shown in Figure 1. It covers seven commonly used techniques in table tennis (i.e., *attack, block, flick, pendulum, push, short, topspin*). The whole dataset consists of 32 amateur players, 380 multi-ball training rounds, 8,655 strokes, and a total of 8,395 pieces of professional suggestions from coaches.

The data structure is as follows:

- **Video Data** All videos were recorded using two UGREEN 2K webcams with OBS in *MKV* format, with a frame rate of 60hz and 1080p. According to the coach's observation habits, cameras are placed on the player's right-hand side and diagonally opposite. In the database, we label these positions as 'right' and 'remote' respectively.
- **Sensor Data** Sensor data are collected using *Wit-Motion BWT901BLE5.0* with 100hz sampling rate. Sensor data dimensions include acceleration(x,y,z), angular(x,y,z), angular speed(x,y,z), magnetic(x,y,z) and quaternion(1,2,3,4).
- **Text Data** We used GPT-4o to convert transcribed audio text into structured text. Each coach's suggestions for each round are stored separately in a JSON file with an index. Each suggestion is parsed into a structured object including these keys: *suggestion_key*, *suggestion_type*, *suggestion_content*, *start_time*, *end_time*, while the **stroke_index** associates with a specific stroke if it is live comments.

3.3 Data Collection

Raw data of T3Set were collected from table tennis training scenarios at the university.

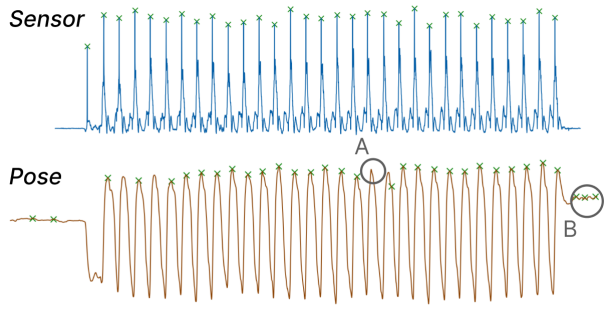

Figure 2: The stroke detection of sensor data and pose data. A: lack of detection; B: extraneous movements. The alignment task is special for our scene and data features. We could find the equal-distance, and the extraneous movements often occur at the start and the end.

Participants: We recruited 32 table tennis amateurs (18 male, 14 female) from university student populations. Participant ages ranged from 19 to 33 years (M=24.3, SD=3.2), with gaming experience spanning 4-18 years (M=10.4, SD=4.2). Professional training duration varied between 0-8 years (M=2.0, SD=2.0). This diverse range of age, experience, and training underscores the diverse backgrounds of the participants and their suitability to represent amateur players. All participants signed informed consent forms and authorized open-source usage. The experimental procedure was approved by the laboratory's ethics review.

Procedure: At the beginning of data collection, all participants signed informed consent. We used a ball machine instead of a coach to feed balls to the participants during the multi-ball stroke training to control the consistent quality and frequency of the fed balls. Participants were required to use specially designed rackets equipped with Inertial Measurement Units (IMU) in the handles. We also placed two cameras according to the recommendations from coaches to capture participants' motions. One on the participants' right-hand side and the other on the diagonal side. All strokes followed standardized parameters of each technique, including serving machine settings, participants' hitting positions, and target landing points. Each participant needed to perform 7 techniques, and for each technique, he/she needed to conduct 3 rounds, which contained about 30 strokes (20 for some cases). Each participant was asked to take a short break after completing each technique (3 rounds). During the collection, data exceptions occurred due to: (1) time/space constraints that prevented three full rounds, (2) occasional video recording gaps during expert labeling, and (3) sensor data loss caused by connectivity issues. Therefore, we totally obtained 8655 valid strokes.

3.4 Data Processing

We developed a three-phase alignment approach to processing the raw data, which includes: (1) Clip Segmentation and Pose Extraction, (2)Stroke detection and Cross-modality Alignment. (3) Coach Annotation and Manual Checking. We hope our process can serve as a reference for constructing similar sports datasets.

3.4.1 Clip Segmentation and Pose Extraction. We segment and organize the video clip and sensor clip for each round, create an index, and use Mediapipe to extract poses from the videos. We use MediaPipe Pose [17] to extract the strokes from videos. The selection of MediaPipe Pose over higher-precision alternatives was motivated by its optimal balance between computational efficiency and accuracy, which is particularly crucial for real-world deployment scenarios. This design choice does not compromise the core technical contributions focusing on cross-modal alignment. We chose MediaPipe Pose's pose_world_landmarks. x,y,z are Real-world 3D coordinates in meters with the origin at the center between hips. Pose data is an intermediate product but not our featured modality in T3Set. By using stroke sequences in pose data, we align the sensor data with the video more precisely at the timestamp level.

3.4.2 Stroke Detection and Cross-modality alignment. We use sensor and pose data to perform stroke detection on the sensor and video data. Although there are some efficient ball trajectory detection algorithms available now[38], there are too many balls in multi-ball training scenarios, so we rely on expert observation for ball tracking.

Stroke Detection. Video-based action recognition has matured significantly in sports analytics. However, our experimental setup creates uniquely favorable conditions for signal processing approaches due to three inherent simplifications: (1) Fixed-interval stroke execution enforced by ball machine pacing, (2) Highly repetitive movement patterns. These controlled conditions allow reliable detection through waveform analysis of sensor and pose trajectories, as shown in Figure 2. We utilized *scipy.signal.find_peaks* in *Python* to detect stroke peak in signals of: Pose: Right wrist trajectory (most regular kinematic marker) Sensors: Technique-specific threshold on $\sqrt{a_x^2 + a_y^2 + a_z^2}$. Finally, we manually checked the results to ensure accuracy.

There are two types of errors in stroke detection. (1)Under-detection: Sensor occasionally missed strokes per sequence (Fig. 2A) (2)Over-detection: Pose data sometimes registered extraneous movements (Fig. 2B). What's more, the timestamp of the sensor and video is not aligned. Therefore, it is necessary to align and further match the number of strokes recognized by both modalities.

Cross-Modality Alignment. This paragraph presents the methodology for aligning video and sensor data. The alignment challenge stems from two temporal discrepancies: (1) IMU timestamps exhibit sub-second deviations (<1s), and (2) video recordings only provide second-level precision for start/end timestamps. This causes issues when trying to match peak times after completing stroke detection. Additionally, there may be missed strokes (Figure 2(A)) or false peaks (Figure 2(B)) during stroke detection. This creates significant alignment challenges between modalities. We first use a regression algorithm to match the data, then plot the matching results, and manually correct any low-quality matches.

The second step is to match the sensor sequence and pose sequence. Since the two sequences, the pose sequence extracted from video **P** and the sensor sequence **S**, describe the same series of stroke motion sequences, we can assume a fixed time offset between these two sequences, denoted as d.

Our objective is, given a tolerance δ, to adjust $\mathbf{P}$ by the offset d and check if the points in $\mathbf{P}$ fall within the bin $[s_j - \delta, s_j + \delta]$ for some s_i in $\mathbf{S}$. If this condition holds, match p_i with s_j. If multiple points match the same s_j, we select the smallest one since the points are close enough that it does not matter which one is chosen. The algorithm could be described as Algorithm 1. We assume that the

Algorithm 1: Matching Peaks from Video and Sensor Data

Input: Pose sequence $\mathbf{P}$ extracted from video, sensor sequence $\mathbf{S}$, tolerance δ, offset d
Output: A sequence of matched stroke pairs (p_i, s_j)

```
# P and S are in ascending order.
for each point s_j ∈ S do
    for each point p_i ∈ P do
        if p_i + d ∈ [s_j − δ, s_j + δ] then
            Match p_i with s_j;
            break;
        end
    end
end
return All matched stroke pairs
```

strokes in the middle portion of the timeline are less likely to be missing or incorrect. Therefore, we select n consecutive strokes around the center of the pose sequence extracted from the video and also select n consecutive strokes from the sensor data. Let $\mathbf{p}$ represent the chosen pose data and $\mathbf{s}$ represent the chosen sensor data, with the assumption $\mathbf{s} = \mathbf{p} + d$. We aim to find:

$$d^* = \underset{d}{\operatorname{argmin}} \sum_{i=1}^{n} (s_i - (p_i + d))^2, \quad s_i \in \mathbf{s},\ p_i \in \mathbf{p}$$

Here, $\mathbf{s}$ can be any continuous segment of the complete sensor data sequence $\mathbf{S}$, and we traverse the entire sequence $\mathbf{S}$ to identify the optimal d^* using Algorithm 1, thereby maximizing the number of matched points.

3.4.3 Coach Annotation and Manual Check. Our annotation process involved four domain experts: **Coach A**: International-certified referee and Grade-One athlete with 15+ years coaching experience. **Coach B**: University-level table tennis instructor who has rich teaching experience. **Coaches C & D**: Grade-One athletes with demonstrated coaching competency (>50 amateurs).

When coaches give suggestions, they should follow their usual approach to training suggestions: provide **live comments** while the video plays and give **summary suggestions** after watching. In summary suggestions, the focus should not be on explanations but on highlighting the main shortcomings in a round. All suggestions are recorded as audio and transferred into text. Experts' suggestions are synchronized to video timestamps during the annotation process. In the end, we manually checked all stroke detection, alignment, and suggestion annotation information.

3.5 Suggestion Taxonomy

3.5.1 Suggestion Keys and Suggestion Types. In discussions with Coach A and Coach B, we summarized the suggestions found in all annotation results and developed a hierarchical suggestion taxonomy. We categorized all suggestion keys from multi-ball training into 25 types and suggestion types into five subtypes. The **suggestion keys** are shown in the Appendix. We used GPT-4o to convert the semi-structured text annotated by the coaches into a structured format (Key-Type-Content), following our taxonomy. This enables us to support the quantitative evaluation of suggestions, including qualitative assessments of suggestions. We define a Suggestion Pair as **suggestion_pair**=(**suggestion_key**,**suggestion_type**). When calculating precision and other metrics later, we remove duplicates based on the predefined *key*, *type*, and *suggestion_pair* before computing the matches.

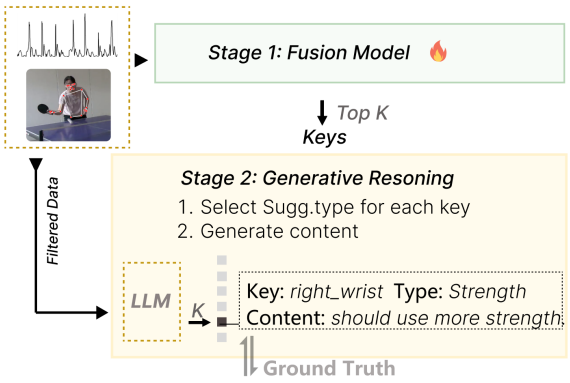

Figure 3: SenseCoach Architecture. Stage 1: Our fusion model is used to select the suggestion key. Stage 2: Combined with extracted data, LLM is utilized to generate suggestions according to each key.

4 Application: SenseCoach

Based on **T3Set**, we present SenseCoach (Figure 3), a two-stage framework that bridges multimodal sensing with LLM reasoning for sports training, which helps LLM generate more targeted training suggestions and validates the usability of our dataset.

4.1 Stage 1: Fusion Model for Key Selection

4.1.1 Problem Definition. Consider a table tennis training session (called a round) containing N consecutive strokes. Let each stroke s_i $(i = 1, ..., N)$ be represented as: $s_i = (\mathbf{P}_i, \mathbf{M}_i)$ Where:

- $\mathbf{P}_i \in \mathbb{R}^{33\times10\times3}$ denotes pose features (33 body joints × 10 timesteps × 3D coordinates)
- $\mathbf{M}_i \in \mathbb{R}^{16}$ contains sensor measurements from the IMU mounted to the rackets.

Given a sequence of strokes $\mathcal{S} = \{s_1, ..., s_N\}$, we formulate a *multi-label ranking task* that predicts relevant suggestion keys from a predefined set $\mathcal{K} = \{k_1, ..., k_{25}\}$. The ground truth is a subset $\mathcal{K}^* \subset \mathcal{K}$ that contains expert-annotated keys.

Our multi-modal fusion model processes table tennis stroke data through three core components, as illustrated in Figure 4. In this content, let B denote batch size, K_p represent the number of pose joints (with $K_p = 33$), K_s represent the number of sensor measurements (with $K_s = 16$), and H represent the hidden size (with $H = 64$).

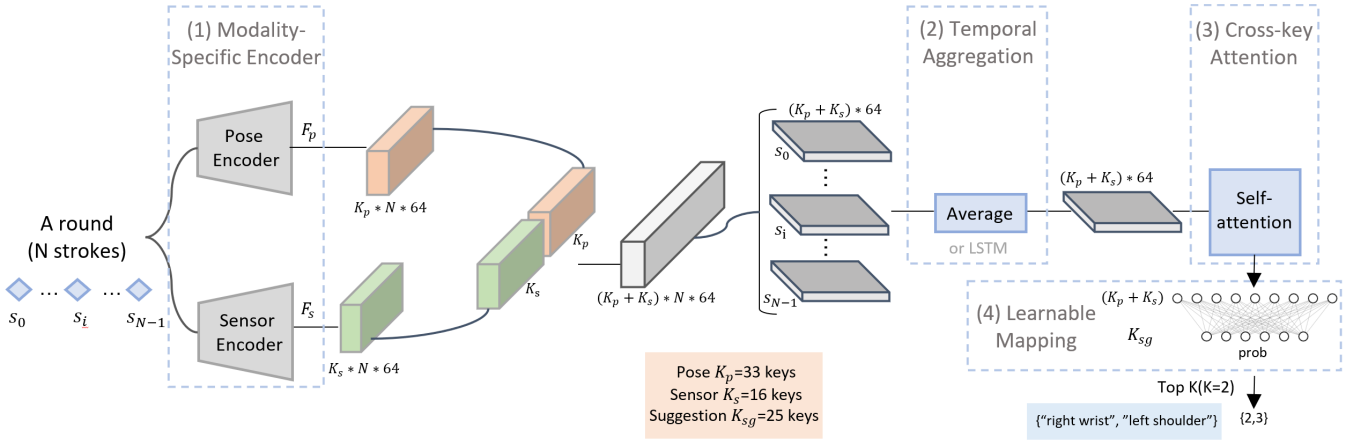

Figure 4: Multimodal Fusion Model Architecture. The model processes structured input from T3Set through three core components:(1) Modality-Specific Encoders: Pose Encoder and Sensor Encoder. (2) Temporal Aggregation: Mean pooling or LSTM. (3) Cross-Key Attention: Inter-modal feature alignment, (4)Learnable Key Mapping: Mapping layer from data keys to suggestion keys. Final predictions output top-K suggestion keys based on probability distribution ranking.

4.1.2 Modality-Specific Encoding. The pose and sensor data undergo separate feature extraction:
Pose Encoder handles 3D joint coordinates across 10 timesteps:

$$\mathbf{F}_p = f_{\text{pose}}(\mathbf{P}) \in \mathbb{R}^{B\times S\times K_p\times H} \tag{1}$$

Sensor Encoder processes 16 sensor measurements per stroke:

$$\mathbf{F}_s = f_{\text{sensor}}(\mathbf{M}) \in \mathbb{R}^{B\times S\times K_s\times H} \tag{2}$$

where f_{sensor} contains two linear layers with GELU activation and layer normalization. Then we concatenate the pose feature ($\mathbf{F}_p$) and sensor feature ($\mathbf{F}_s$).

4.1.3 Temporal Aggregation. We used average pooling for temporal aggregation, and we will compare their differences in the ablation study. Concatenated features $\mathbf{F} = [\mathbf{F}_s; \mathbf{F}_p] \in \mathbb{R}^{B\times S\times (K_s+K_p)\times H}$ are aggregated through **Average Pooling** (average of different strokes in a round).

$$\mathbf{F}_{\text{agg}} = \frac{1}{S}\sum_{i=1}^{S} \mathbf{F}_{:,i,:,:} \tag{3}$$

4.1.4 Cross-Key Attention. A 4-head self-attention mechanism discovers interactions between (K_p+K_s) data keys:

$$\mathbf{A} = \text{MultiHeadAtt}(\mathbf{F}_{\text{agg}}, \mathbf{F}_{\text{agg}}, \mathbf{F}_{\text{agg}}) \in \mathbb{R}^{B\times (K_s+K_p)\times H} \tag{4}$$

4.1.5 Learnable Key Mapping. We previously introduced the suggestion keys (subsubsection 3.5.1). In the pose data and sensor data, each has its own data dimensions. Pose data has 33 dimensions corresponding to the 33 skeletal joints in MediaPipe. Sensor data has 16 dimensions. We created a predefined mapping dictionary from suggestion keys to data keys based on expert advice and physical quantity correspondence. For example, for *angle of racket*, we map it to the sensor's *angle_y* at the peak of the strike, as the *angle_y* relates directly to the racket's tilt given the IMU installation direction. We also designed an additional layer of dimension processing for sensor dimensions, aggregating sensor dimensions such as *acc_x*, *acc_y*, and *acc_z* by taking the square root of their sum of squares. In this case, the sensor data dimensions become 3 instead of 16. We will discuss the differences of this design in the later ablation study. The number of suggestion keys is K_{sg}=25. A sparse projection matrix $\mathbf{W}_m \in \{0,1\}^{(K_s+K_p)\times K_{sg}}$ (initialized from expert knowledge) converts data key scores to suggestion keys:

$$\mathbf{z} = (\mathbf{A}\mathbf{W}_o)\mathbf{W}_m \in \mathbb{R}^{B\times K_{sg}} \tag{5}$$

where $\mathbf{W}_o \in \mathbb{R}^{H\times 1}$ produces initial data key scores. The matrix $\mathbf{W}_m$ remains either fixed or trainable based on configuration.

4.1.6 Training & Inference. Using BCEWithLogitsLoss for multi-label classification:

$$\mathcal{L} = -\sum_{j=1}^{K_{sg}} w_j [y_j \log \sigma(z_j) + (1-y_j)\log(1-\sigma(z_j))] \tag{6}$$

Top-k suggestions are generated by selecting the highest scoring keys from $\mathbf{z}$ without sigmoid transformation. The learning rate is set as 0.00001.

4.2 Stage 2: Suggestions Generation with LLM

In this part, we introduce stage 2, generating suggestions using the inference capabilities of LLMs. In stage 1, our fusion model selects the top-k suggestion keys. Based on these top-k keys, we map these keys to their corresponding multimodal dimensions and corresponding data evidence. This key-evidence bundle serves as the LLM's input context, with detailed prompt provided in the Appendix. In stage 2, the LLM performs two core tasks: selecting the predefined type according to the input key and generating suggestions grounded in sensor-video evidence. The output is a list of suggestions. Each suggestion generated by the virtual coach includes: (1) a suggestion key in the predefined list and (2) a suggestion type allowed in the predefined list. (3) improvement suggestion content(round-level rather than stroke-specific comments). We use *precision* to evaluate the ability to select the suggestion type and ROUGE-L[25] to assess the text similarity.

	TopK=6			TopK=7			TopK=8			TopK=9		
Model Name	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Full Model	**76.32**	50.22	60.57	73.50	56.42	63.84	70.89	62.19	66.26	68.71	67.82	68.26
w/o lr_mp layer	56.58	37.23	44.91	56.39	43.29	48.98	57.07	50.07	53.34	57.46	56.71	57.08
w/o sensor preprocess	76.10	50.07	60.40	73.68	56.57	64.00	70.89	62.19	66.26	69.30	68.40	**68.85**
w/o stroke-avg	74.78	49.21	59.36	72.56	55.70	63.02	70.89	62.19	66.26	69.30	68.40	68.85

Table 1: Ablation Study of Stage 1 Fusion Model. Evaluates key components: (1) Learnable mapping (lr_mp) vs fixed projection, (2) Sensor preprocessing with key-based aggregation, (3) Temporal aggregation using stroke-mean pooling vs LSTM. Metrics: Precision (P) and Recall (R) for suggestion key prediction.

5 Experiments

We conducted three core experiments to compare the ability of **Suggestion Key Selection**, **Suggestion Type Selection**, and **Suggestion Key-Type Pair Selection**. The fusion model is used to select suggestion keys. We designed an ablation study to compare the model structure design, as shown in Table 1; we tested various LLMs in Stage 2 on **Suggestion Type Selection**, as shown in Table 2. We validate our framework's effectiveness against baseline methods Table 3. The following subsections detail our experimental protocol, metric selection, and quantitative analysis.

5.1 Stage 1 Evaluation: Ablation Study

Building on the architecture described in subsection 4.1, we systematically evaluate three critical design choices:

(1) **Temporal Aggregation**: LSTM vs. Stroke Average.
(2) **Key Mapping**: Learned vs. predefined coach mappings
(3) **Sensor Encoding**: With/without kinematic preprocessing (dimension aggregation)

The dataset was partitioned into 60% training, 20% validation, and 20% test sets, ensuring temporal continuity within sessions. Table 1 presents the quantitative comparisons. We use the precision, recall, and F1 to measure the ability to select suggestion keys.

5.1.1 Result analysis. We analyze the results in Table 1 as follows. (1) **learning Key Mapping**: it learns the more accurate suggestion keys-data keys relationships than the predefined mapping we propose with coaches. (2) **Temporal Aggregation Strategy** Using the averaging of strokes is generally better than using LSTM for temporal aggregation, but we think the difference is not significant. Due to the nature of multi-ball training, the strokes in a round are basically detailed repetitive movements, which means repeating similar mistakes. So, the averaging method is more advantageous. (3) **Sensor Preprocessing**: Given the test set size (76 rounds), we think the performance difference is not significant, but the aggregated sensor dimensions are easier for the LLM to understand.

5.2 Stage 2 Evaluation: LLM Ability Test

According to our suggestion taxonomy, if LLM wants to generate suggestions, it should 1) select the suggestion key, 2) select the suggestion type, and 3) generate suggestion content. Our evaluation framework assesses LLMs' capacity to classify suggestion types. For the prediction precision loss in stage 1, corrections should be made in stage 2 to ensure the evaluation of the LLM's ability on the suggestion type.

model-name	version	P@6(%)	**P@6-S1L(%)**	R-L(%)
LLaMa 3.3 70B[29]	-	38.82	**51.64**	18.96
Claude 3.5 Haiku[1]	241022	**39.04**	51.18	16.80
DeepSeek-R1[14]	-	38.67	50.91	15.63
GPT-4o[31]	240806	36.40	48.49	20.02
o1-preview[32]	240912	36.62	48.07	18.89
LLaMa 3.1 70B[28]	-	35.31	45.72	18.37
Qwen2.5-32B Instruct[36]	-	34.65	45.37	22.09
Claude 3.5 Sonnet[1]	241022	33.11	44.21	19.68
LLaMa 3.1 405B[28]	-	32.46	43.93	19.31
Gemini 2.0 Flash[12]	2502	33.55	43.51	19.55
Qwen2.5-72B Instruct[36]	-	32.02	41.86	22.82
DeepSeek-V3[13]	-	30.26	40.94	21.76
Qwen2.5-Max[37]	240919	28.95	37.35	**23.34**

Table 2: Type Selection Precision in Stage 2. This ability is evaluated through P@6-S1L: Type classification accuracy under perfect key assumption.

5.2.1 Evaluation Protocol and Metric Design. **P (Precision)**: Correct key-type pair selection. **P-S1L**: Type classification accuracy under perfect key assumption. **R-L(Rouge-L)**: Content similarity between generated and expert suggestions. We evaluated using precision rather than recall or F1 because the model or the LLM has not learned about the number of expert suggestions (key-type pairs), so using them for evaluation would not be meaningful. For example, the model is asked to generate top-k=6 suggestions, which include 6 (key, type) pairs. In the ground truth, after taking the union of suggestions from the coach, there are 12 (key,type) pairs. If 4 of the model's 6 suggestions are in these 12 pairs, to calculate precision, we have TP=4 and FP=2, so the precision is 66.67%. In the FP, one is due to an incorrect key selection. After removing it, when calculating P-S1L, we have TP=4 and FP=1, so P-S1L is 80.0%.

5.2.2 Result analysis. Here, we compared the various LLMs' ability to select suggestion types (as shown in Table 2). For example, Deepseek models outperformed others in type selection. ROUGE-L

Model	Type	Select-Key			Select (key,Type)	Select-Type	Content
		P@6(%)	R@6(%)	F1@6(%)	P@6(%)	P@6-S1L(%)	Rouge-L(%)
Claude-3.5-Haiku	Our Method	**75.66**	**47.69**	**57.71**	**39.04**	51.18	16.80
	text-only	56.58	35.95	43.39	30.48	**53.79**	16.56
LLaMa3.3 70B	Our Method	73.68	46.43	56.19	38.82	51.64	18.96
	text-only	32.46	20.13	24.46	14.47	44.63	16.31
DeepSeek-R1	Our Method	75.56	47.11	57.38	38.67	50.91	15.63
	text-only	56.36	35.53	42.99	27.19	47.26	15.39
GPT-4o	Our Method	73.68	46.43	56.19	36.40	48.49	**20.02**
	text-only	60.09	38.18	46.05	31.36	49.85	17.89
GPT o1-preview	Our Method	**75.66**	**47.69**	**57.71**	36.62	48.07	18.89
	text-only	52.63	32.46	39.61	26.10	49.25	17.19
Gemini-1.5-Pro[11]	video	67.54	42.58	51.55	18.20	26.21	18.89
VideoChat-Flash-qwen2-7B[33]	video	9.87	6.38	7.64	5.92	33.99	11.27
-	Random Selection	32.00	24.00	27.43	9.77	30.53	-

Table 3: Performance compared with baseline. Comparison with: (1) Text-Only LLM (structured data prompting), (2) Video LLM (video input+structured sensor data). Evaluated on: (a) Key Selection (b) Full Suggestion (key+type accuracy) (c) Type Selection.

scores showed marginal variance, which shows that content metrics poorly reflect targeting specificity.

5.3 Baseline Comparison

We evaluate our two-stage framework against two baseline approaches: **Text-Only LLM**: Processes structured T3Set data through standard prompting, **Video LLM (vLLM)**: Give structured sensor data and the raw video. (The prompts are given in the Appendix).

Due to time issues and difficulty deploying large open-source vLLMs, we only tested this Gemini-1.5pro (SOTA on many benchmarks) and VideoChat-7B (good in MVBench and Chardes-STA).

5.3.1 Result Analysis. We analyze the results in Table 3 as follows. (1) **Key Selection**. Our fusion model achieved 75.66% precision vs. 66.67% (VLLM) and 56.97% (Text-Only LLM). Random selection baseline: 32.0%. (2) In generating suggestions (**selecting the suggestion pair**, meaning both key and type), our full pipeline precision reached 39.03%, surpassing Text-Only LLM's 31.36% and VLLM's 18.20%. Our method (stage 1 + stage 2) also outperforms the based large language model used in stage 2. (3) **ROUGE-L** scores showed minimal variation.

5.4 Case study

We introduce two suggestion generation cases along with corresponding feedback from coaches.

Angle of Racket. We show experts the suggestions in the round anonymous_0024-short-round_00. Coach C's suggestion in annotation is *"You shouldn't make the racket perpendicular to the table."* The LLM's suggestion is *"Adjust the angle of the racket to achieve better spin and control during strokes."* Both suggestions focus on the angle of the racket.

Coach A thought while both suggestions had nuance differences, both of them were valid. He pointed out that the angle of the racket at the stroke moment is a variable that is difficult to observe. Coaches usually judge based on the ball trajectory combined with their experience. Here, using sensor data key *angle_y* can provide a reasonable suggestion key(it's correlated with racket face orientation).

Center of Gravity. In anonymous_0026-pendulum-round_02, most of the suggestions from coaches about the center of gravity were related to *position*, such as *"Center of gravity is too high"*. LLM's suggestions also focused on the center of gravity. However, most of these suggestions are about the stability of the center of gravity. This difference reveals the limitation of our framework where the suggestion type is selected by LLM and LLM lacks enough domain knowledge.

6 Discussion

In this section, we discuss the limitations of this work and potential research opportunities based on our dataset.

6.1 Limitation

The limitations of our work primarily stem from the constraints in computational resources, which have impacted our ability to fine-tune LLM directly using our dataset. As a result, while our framework demonstrates the potential for enhancing reasoning ability and domain-specific coaching capabilities aligned with human expertise, the lack of fine-tuning means that the model's performance may not be fully optimized for suggestion generation, as we can find in Section 5.4. In the future, we will try to find more computational resources and try more alternatives to the framework to enhance the quality of generated suggestions. Due to limitations in the scope of previous data collection, the diversity of this dataset does not include children and elderly individuals. We will collect more diverse samples in the future.

6.2 Research opportunities

Better virtual coach: Our dataset, which integrates video, sensor data, and suggestions, offers a promising foundation for designing an enhanced virtual coach for table tennis training. We have prepared and cleaned the text data, making it easy to transfer into the QA format using Python scripts and be used in the supervised fine-tuning(SFT) or retrieval augmentation(RAG). By leveraging T3Set, we can create a coach that goes beyond basic motion analysis to provide targeted and actionable suggestions tailored to specific training scenarios. However, there is still significant room for improvement in the quality of these suggestions. Both the accuracy and the intuitiveness of the suggestions can be further refined to align with players' needs and optimize their training experience.

Future long-term experiment: In this work, we specifically aim at the training scene, not real match conditions. The robustness and adaptability in the multi-ball scenario is valid and highly accurate. Real-world adoption is feasible after solving the alignment of different modalities. We will build an LLM-based coaching app and conduct more experiments and assess how AI-generated suggestions impact actual skill improvement and player experience, validating real-world effectiveness. We hope other researchers can utilize this dataset to conduct long-term studies to validate real-world training efficacy beyond technical accuracy metrics, or transition from training scenarios to competition scenarios.

Expanding to other sports, domains, and modalities: Based on our practice, there is significant potential for generating similar datasets in other sports. Sports like basketball, football, or badminton also involve complex kinematic information and require targeted suggestions [6, 46]. By collecting multimodal data from these sports, we can develop virtual coaches tailored to the specific demands and techniques of each sport. We believe that T3Set fills the gap of the lack of a multi-modal dataset for racket sports training. This aligned dataset will bring more possibilities for sports visual analytics and provide more opportunities for developing advanced training tools and interfaces. Moreover, other modalities, such as biomechanical data, would allow for richer datasets and a deeper understanding of player performance. This would help create more comprehensive and adaptable coaching systems that can be applied across a wide range of sports, pushing the boundaries of personalized, AI-driven sports training.

7 Conclusion

In this paper, we introduce T3Set, a multimodal dataset that combines aligned video, sensor data, and suggestion data for table tennis training. Based on this dataset, we propose a two-stage framework to generate more targeted coaching suggestions. While the current system shows promise, further improvements in the accuracy and intuitiveness of the suggestions are needed. Our approach has the potential to advance virtual coaching and provide players with a more personalized, data-driven training experience.

Acknowledgments

The work was supported by NSFC (62421003, U22A2032). We thank all reviewers for their insightful feedback and comments, and all participants in the data collection.

References

[1] Anthropic. 2024. Introducing computer use, a new Claude 3.5 Sonnet, and Claude 3.5 Haiku. Retrieved February 17, 2025 from https://www.anthropic.com/news/3-5-models-and-computer-use

[2] Kumar Ashutosh, Tushar Nagarajan, Georgios Pavlakos, Kris Kitani, and Kristen Grauman. 2024. ExpertAF: Expert Actionable Feedback from Video. doi:10.48550/arXiv.2408.00672 arXiv:2408.00672

[3] Jiang Bian, Xuhong Li, Tao Wang, Qingzhong Wang, Jun Huang, Chen Liu, Jun Zhao, Feixiang Lu, Dejing Dou, and Haoyi Xiong. 2024. P2ANet: A Large-Scale Benchmark for Dense Action Detection from Table Tennis Match Broadcasting Videos. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 4 (Jan. 2024), 23 pages. doi:10.1145/3633516

[4] Henrik Biermann, Jonas Theiner, Manuel Bassek, Dominik Raabe, Daniel Memmert, and Ralph Ewerth. 2021. A Unified Taxonomy and Multimodal Dataset for Events in Invasion Games. In *Proceedings of the 4th International Workshop on Multimedia Content Analysis in Sports*. 1–10. doi:10.1145/3475722.3482792

[5] Peter Blank, Benjamin H. Groh, and Bjoern M. Eskofier. 2017. Ball speed and spin estimation in table tennis using a racket-mounted inertial sensor. In *Proceedings of the ACM International Symposium on Wearable Computers*. 2–9. doi:10.1145/3123021.3123040

[6] Anqi Cao, Xiao Xie, Runjin Zhang, Yuxin Tian, Mu Fan, Hui Zhang, and Yingcai Wu. 2025. Team-Scouter: Simulative Visual Analytics of Soccer Player Scouting. *IEEE Transactions on Visualization and Computer Graphics* 31, 1 (2025), 1–11. doi:10.1109/TVCG.2024.3456216

[7] Liqi Cheng, Dazhen Deng, Xiao Xie, Rihong Qiu, Mingliang Xu, and Yingcai Wu. 2024. SNIL: Generating Sports News From Insights With Large Language Models. *IEEE Transactions on Visualization and Computer Graphics* (2024), 1–14. doi:10.1109/TVCG.2024.3392683

[8] Shang-Hsuan Chiang, Lin-Wei Chao, Kuang-Da Wang, Chih-Chuan Wang, and Wen-Chih Peng. 2024. BADGE: BADminton Report Generation and Evaluation with LLM. doi:10.48550/arXiv.2406.18116 arXiv:2406.18116

[9] Che-Yu Chou, Zheng-Hao Chen, Yung-Hoh Sheu, Hung-Hsuan Chen, Min-Te Sun, and Sheng K. Wu. 2025. TTSwing: A Dataset for Table Tennis Swing and Racket Kinematics Analysis. *Scientific Data* 12, 1 (2025), 339. doi:10.1038/s41597-025-04680-y

[10] Cristian Comendant. 2024. *Large Language Model-Based Sport Coaching System Using Retrieval-Augmented Generation and User Models*. B.S. thesis. University of Twente.

[11] Google DeepMind. 2024. Gemini 1.5 Pro Model Card. Retrieved February 17, 2025 from https://www.promptthub.us/models/gemini-1-5-pro

[12] Google DeepMind. 2025. Gemini 2.0 is now available to everyone. Retrieved February 17, 2025 from https://blog.google/technology/google-deepmind/gemini-model-updates-february-2025/

[13] DeepSeek. 2024. Introducing DeepSeek-V3. Retrieved February 17, 2025 from https://api-docs.deepseek.com/news/news1226

[14] DeepSeek. 2025. DeepSeek-R1 Release. Retrieved February 17, 2025 from https://api-docs.deepseek.com/news/news250120

[15] Zikun Deng, Haoming Chen, Qing-Long Lu, Zicheng Su, Tobias Schreck, Jie Bao, and Yi Cai. 2025. Visual Comparative Analytics of Multimodal Transportation. *Visual Informatics* 9, 1 (2025), 18–30. doi:10.1016/j.visinf.2025.01.001

[16] Emilio Ferrara. 2024. Large Language Models for Wearable Sensor-Based Human Activity Recognition, Health Monitoring, and Behavioral Modeling: A Survey of Early Trends, Datasets, and Challenges. *Sensors* 24, 15 (2024), 5045. doi:10.3390/s24155045

[17] Google. 2024. MediaPipe Pose. Retrieved Februry 17, 2025 from https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/pose.md

[18] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolář, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. 2022. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18995–19012. doi:10.1109/CVPR52688.2022.01842

[19] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal,

Bikram Boote, Eugene Byrne, Zach Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, Maria Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina Gonzalez, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, Anush Kumar, Robert Kuo, Sach Lakhavani, Miao Liu, Mi Luo, Zhengyi Luo, Brighid Meredith, Austin Miller, Oluwatumininu Oguntola, Xiaqing Pan, Penny Peng, Shraman Pramanick, Merey Ramazanova, Fiona Ryan, Wei Shan, Kiran Somasundaram, Chenan Song, Audrey Southerland, Masatoshi Tateno, Huiyu Wang, Yuchen Wang, Takuma Yagi, Mingfei Yan, Xitong Yang, Zecheng Yu, Shengxin Cindy Zha, Chen Zhao, Ziwei Zhao, Zhifan Zhu, Jeff Zhuo, Pablo Arbelaez, Gedas Bertasius, Dima Damen, Jakob Engel, Giovanni Maria Farinella, Antonino Furnari, Bernard Ghanem, Judy Hoffman, C.V. Jawahar, Richard Newcombe, Hyun Soo Park, James M. Rehg, Yoichi Sato, Manolis Savva, Jianbo Shi, Mike Zheng Shou, and Michael Wray. 2024. Ego-Exo4D: Understanding Skilled Human Activity from First- and Third-Person Perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19383–19400. doi:10.1109/CVPR52733.2024.01834

[20] Yaodong Gu, Changxiao Yu, Shirui Shao, and Julien S. Baker. 2019. Effects of table tennis multi-ball training on dynamic posture control. *PeerJ* 6 (2019), e6262. doi:10.7717/peerj.6262

[21] Jan Held, Hani Itani, Anthony Cioppa, Silvio Giancola, Bernard Ghanem, and Marc Van Droogenbroeck. 2024. X-VARS: Introducing Explainability in Football Refereeing with Multi-Modal Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3267–3279. doi:10.1109/CVPRW63382.2024.00332

[22] Yoichi Iino and Takeji Kojima. 2009. Kinematics of table tennis topspin forehands: effects of performance level and ball spin. *Journal of Sports Sciences* 27, 12 (2009), 1311–1321. doi:10.1080/02640410903264458

[23] Abdullah Aman Khan, Jie Shao, Waqar Ali, and Saifullah Tumrani. 2020. Content-Aware Summarization of Broadcast Sports Videos: An Audio–Visual Feature Extraction Approach. *Neural Processing Letters* 52, 3 (2020), 1945–1968. doi:10.1007/s11063-020-10200-3

[24] Haopeng Li, Andong Deng, Qiuhong Ke, Jun Liu, Hossein Rahmani, Yulan Guo, Bernt Schiele, and Chen Chen. 2024. Sports-QA: A Large-Scale Video Question Answering Benchmark for Complex and Professional Sports. doi:10.48550/arXiv.2401.01505 arXiv:2401.01505

[25] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of summaries, In Text Summarization Branches Out. *Proceedings of the ACL Workshop: Text Summarization Braches Out 2004*, 10. doi:10.48550/arXiv.1803.01937

[26] Ziao Liu, Xiao Xie, Moqi He, Wenshuo Zhao, Yihong Wu, Liqi Cheng, Hui Zhang, and Yingcai Wu. 2025. Smartboard: Visual Exploration of Team Tactics with LLM Agent. *IEEE Transactions on Visualization and Computer Graphics* 31, 1 (2025), 23–33. doi:10.1109/TVCG.2024.3456200

[27] Wenlong Ma, Yang Liu, Qng Yi, Xutao Liu, Wei Xing, Rongji Zhao, Huan Liu, and Rongzhi Li. 2025. Table tennis coaching system based on a multimodal large language model with a table tennis knowledge base. *PLOS ONE* 20 (2025). doi:10.1371/journal.pone.0317839

[28] Meta. 2024. Introducing Llama 3.1:Our most capable models to date. Retrieved February 17, 2025 from https://ai.meta.com/blog/meta-llama-3-1/

[29] Meta. 2024. Model Cards&Prompt formats Llama 3.3. Retrieved February 17, 2025 from https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/

[30] Pulkit Narwal, Neelam Duhan, and Komal Kumar Bhatia. 2023. A novel multimodal neural network approach for dynamic and generic sports video summarization. *Engineering Applications of Artificial Intelligence* 126 (2023), 106964. doi:10.1016/j.engappai.2023.106964

[31] OpenAI. 2024. Hello GPT-4o. Retrieved February 17, 2025 from https://openai.com/index/hello-gpt-4o/

[32] OpenAI. 2024. Introducing OpenAI o1-preview. Retrieved February 17, 2025 from https://openai.com/index/introducing-openai-o1-preview/

[33] OpenGVLab. 2024. VideoChat-Flash-Qwen2-7B. Retrieved February 17, 2025 from https://huggingface.co/OpenGVLab/VideoChat-Flash-Qwen2-7B_res448

[34] Ruixiao Peng, Yu Dong, Guan Li, Dong Tian, and Guihua Shan. 2025. TextLens: Large Language Models-Powered Visual Analytics Enhancing Text Clustering. *Journal of Visualization* (2025). doi:10.1007/s12650-025-01043-y

[35] Jielin Qiu, Jiacheng Zhu, William Han, Aditesh Kumar, Karthik Mittal, Claire Jin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Ding Zhao, et al. 2024. MMSum: A Dataset for Multimodal Summarization and Thumbnail Generation of Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21909–21921. doi:10.1109/CVPR52733.2024.02069

[36] Qwen. 2024. Qwen2.5:A Party of Foundation Models! Retrieved February 17, 2025 from https://qwenlm.github.io/blog/qwen2.5/

[37] Qwen. 2025. Qwen2.5-Max:Exploring the Intelligence of Large-scale MoE Model. Retrieved February 17, 2025 from https://qwenlm.github.io/blog/qwen2.5-max

[38] Zhang Rong. 2024. Optimization of Table Tennis Target Detection Algorithm Guided by Multi-Scale Feature Fusion of Deep Learning. *Scientific Reports* 14, 1 (Jan. 2024), 1401. doi:10.1038/s41598-024-51865-3

[39] Melissa Sanabria, Frédéric Precioso, and Thomas Menguy. 2021. Hierarchical Multimodal Attention for Deep Video Summarization. In *Proceedings of the 25th International Conference on Pattern Recognition*. 7977–7984. doi:10.1109/ICPR48806.2021.9413097

[40] Nina Schaffert, André Engel, Sebastian Schlüter, and Klaus Mattes. 2019. The Sound of the Underwater Dolphin-Kick: Developing Real-Time Audio Feedback in Swimming. *Displays* 59 (2019), 53–62. doi:10.1016/j.displa.2019.08.001

[41] Zhuoyong Shi, Yetao Jia, Guoqing Shi, Kexin Zhang, Longmeng Ji, Dinghan Wang, and Yong Wu. 2024. Design of Motor Skill Recognition and Hierarchical Evaluation System for Table Tennis Players. *IEEE Sensors Journal* 24, 4 (2024), 5303–5315. doi:10.1109/JSEN.2023.3346880

[42] Ian Simpson, Ryan J. Beal, Duncan Locke, and Timothy J. Norman. 2022. Seq2Event: Learning the Language of Soccer Using Transformer-based Match Event Prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3898–3908. doi:10.1145/3534678.3539138

[43] Maria Skublewska-Paszkowska, Pawel Powroznik, Edyta Lukasik, and Jakub Smolka. 2024. Tennis Patterns Recognition Based on a Novel Tennis Dataset – 3DTennisDS. *Advances in Science and Technology Research Journal* 18, 6 (2024), 159–176. doi:10.12913/22998624/191264

[44] Georgia Sovatzidi and Dimitris K. Iakovidis. 2024. FCMCoach: Personalized Virtual Coaching Based on Fuzzy Cognitive Maps. *IEEE Access* 12 (2024), 174413–174423. doi:10.1109/ACCESS.2024.3501675

[45] Tan Tang, Yanhong Wu, Junming Gao, Kejia Ruan, Yanjie Zhang, Shuainan Ye, Yingcai Wu, and Xiaojiao Chen. 2024. ArtEyer: Enriching GPT-based Agents with Contextual Data Visualizations for Fine Art Authentication. *Visual Informatics* 8, 4 (2024), 48–59. doi:10.1016/j.visinf.2024.11.001

[46] Feng Tian, Shuting Ni, Xiaoyue Zhang, Fei Chen, Qiaolian Zhu, Chunyi Xu, and Yuzhi Li. 2024. Enhancing Tai Chi Training System: Towards Group-Based and Hyper-Realistic Training Experiences. *IEEE Transactions on Visualization and Computer Graphics* 30, 5 (2024), 1–11. doi:10.1109/TVCG.2024.3372099

[47] Alain Vázquez, Asier López Zorrilla, Javier Mikel Olaso, and María Inés Torres. 2023. Dialogue Management and Language Generation for a Robust Conversational Virtual Coach: Validation and User Study. *Sensors* 23, 3 (2023), 1423. doi:10.3390/s23031423

[48] Jiachen Wang, Dazhen Deng, Xiao Xie, Xinhuan Shu, Yu-Xuan Huang, Le-Wen Cai, Hui Zhang, Min-Ling Zhang, Zhi-Hua Zhou, and Yingcai Wu. 2021. TacValuer: Knowledge-based Stroke Evaluation in Table Tennis. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3688–3696. doi:10.1145/3447548.3467104

[49] Jiazhe Wang, Xi Li, Chenlu Li, Di Peng, Arran Zeyu Wang, Yuhui Gu, Xingui Lai, Haifeng Zhang, Xinyue Xu, Xiaoqing Dong, Zhifeng Lin, Jiehui Zhou, Xingyu Liu, and Wei Chen. 2024. AVA: An Automated and AI-driven Intelligent Visual Analytics Framework. *Visual Informatics* 8, 2 (2024), 106–114. doi:10.1016/j.visinf.2024.06.002

[50] Jiachen Wang, Kejian Zhao, Dazhen Deng, Anqi Cao, Xiao Xie, Zheng Zhou, Hui Zhang, and Yingcai Wu. 2020. Tac-Simur: Tactic-based Simulative Visual Analytics of Table Tennis. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 407–417. doi:10.1109/TVCG.2019.2934630

[51] Wei-Yao Wang, Yung-Chang Huang, Tsi-Ui Ik, and Wen-Chih Peng. 2023. ShuttleSet: A Human-Annotated Stroke-Level Singles Dataset for Badminton Tactical Analysis. doi:10.48550/arXiv.2306.04948 arXiv:2306.04948

[52] Haotian Xia, Zhengbang Yang, Yun Zhao, Yuqing Wang, Jingxi Li, Rhys Tracy, Zhuangdi Zhu, Yuan-fang Wang, Hanjie Chen, and Weining Shen. 2024. Language and Multimodal Models in Sports: A Survey of Datasets and Applications. doi:10.48550/arXiv.2406.12252 arXiv:2406.12252

[53] Yilin Ye, Jianing Hao, Yihan Hou, Zhan Wang, Shishi Xiao, Yuyu Luo, and Wei Zeng. 2024. Generative AI for Visualization: State of the Art and Future Directions. *Visual Informatics* 8, 2 (2024), 43–66. doi:10.1016/j.visinf.2024.04.003

[54] Fatma Youssef, Victor Parque, and Walid Gomaa. 2023. VCOACH: A Virtual Coaching System Based on Visual Streaming. *Procedia Computer Science* 222 (2023), 207–216. doi:10.1016/j.procs.2023.08.158

[55] Ziyue Yuan, Shuqi He, Yu Liu, and Lingyun Yu. 2023. MEinVR: Multimodal Interaction Techniques in Immersive Exploration. *Visual Informatics* 7, 3 (2023), 37–48. doi:10.1016/j.visinf.2023.06.001

[56] Guodong Zhang, Guanchong Li, Hansen Li, Yuqin Su, and Yun Li. 2024. GPT-4 as a Virtual Fitness Coach: An Evaluation of Its Effectiveness in Providing Weight Loss and Fitness Guidance (Preprint). doi:10.2196/preprints.65470

[57] Jiawen Zhang, Dongliang Han, Shuai Han, Heng Li, Wing-Kai Lam, and Mingyu Zhang. 2025. ChatMatch: Exploring the Potential of Hybrid Vision–Language Deep Learning Approach for the Intelligent Analysis and Inference of Racket Sports. *Computer Speech & Language* 89 (2025), 101694. doi:10.1016/j.csl.2024.101694

[58] Tianjian Zou, Jiangning Wei, Bo Yu, Xinzhu Qiu, Hao Zhang, Xu Du, and Jun Liu. 2024. Fast Moving Table Tennis Ball Tracking Algorithm Based on Graph Neural Network. *Scientific Reports* 14, 1 (Nov. 2024), 29320. doi:10.1038/s41598-024-80056-3

A Appendix

A.1 Suggestions Keys

The **suggestion keys** include *center_of_gravity*, *right_wrist*, *right_shoulder*, *right_forearm*, *time_of_striking_ball*, *angle_of_racket*, *left_foot*, *right_foot*, *waist*, *right_upper_arm*, *left_shoulder*, *left_elbow*, *right_elbow*, *left_wrist*, *backswing_of_racket*, *left_hand*, *right_hand*, *left_leg*, *right_leg*, *left_knee*, *right_knee*, *right_finger*, *upper_body*, *grip_of_racket*, and *others*. The **suggestion types** include *position*, *strength*, *stability*, *racket*, and *others*.

A.2 Prompts

A.2.1 prompt for stage 2 evaluation in table 2. Our prompt includes two parts. One is the system prompt, which describes the task, and the other is the data prompt, which contains the data we give to LLMs. If the LLM interface supports the system prompt, our system prompt will be used as system prompt input, and our data prompt will does not support the system prompt, these two parts will be concatenated in the following manner and used as input: "system: content" + <system prompt> + "user_content" + <data prompt>.

Here is the **system prompt**:
You are a table tennis coach. I have recorded video data and sensor data of a table tennis technique training session.

This session is called 'multi-ball' training, meaning the player strikes the ball using the same technique repeatedly for N times in a round.

Your task is to analyze the suggestions I provide, each with a 'suggestion_key' and its related 'data_key' as well as the corresponding data of this 'data_key'. I will give you M suggestion_key, you MUST response M suggestions according to the suggestion_key I gave you.

<data-description> Here is the explanation of the sensor data: - "acc_peak_exp_sqrt": The peak value of the acceleration in the x, y, z directions. - "agl_y_peak" : The peak value of the angle in the y direction. - "agl_spd_peak_exp_sqrt" : The peak value of the angular speed in the x, y, z directions. Here is the explanation of the pose data: The input data is a sequence of pose data, each with 3 attributes, describing the position of the joint </data-description> <task> Based on this information, you need to:

1. Read the 'suggestion_key' and related data. 2. Select a 'suggestion_type' from the predefined types. 3. Generate a 'suggestion_content' sentence for each 'suggestion_key'. So, your suggestions number should equal to the number of input 'suggestion_key'. 4. Output the suggestion in JSON format. ATTENTION! you should copy the 'suggestion_key' value but not the 'data_key'.

</task> <task-requirements> The 'content' part should be a sentence describing the suggestion. The predefined suggestion keys are: [Predefined suggestions keys] you should just copy the 'suggestion_key' from input but not change or create a new one. The predefined types for 'suggestion_type' are: - position: Indicates that the position of this key is incorrect. - strength: Indicates that the force exertion or explosiveness of this key is incorrect or insufficient. - stability: Indicates that the stability of this key is incorrect. - racket: Indicates that the relationship between this key and the racket is incorrect. - others: Represents other types of suggestions related to this key.

Please ensure that the output is in the JSON format as shown in the example: <output format> ```json "suggestions_list": ["content": "suggestion content", "suggestion_type": "suggestion_type", "suggestion_key": "suggestion_key"] ``` </output format> </task-requirements>

Here is our **data prompt**:

You are expected to generate [input length] suggestions according to the the [input length] suggestion_keys: [input suggestion keys].

We retrievaled the following data of this round for each data_key. Here is the sequence of the pose data.We provide only the data of stroke moment, and the average value + Standard deviation around the peak.[key:key relevant data]

A.2.2 prompt for LLM text-only baseline ability test. In our text-only pipeline precision evaluation("text-only" part in table 3), We use the system prompt + data prompt policy similar to stage2 evaluation in table 2 and "Our Method" part in table 3.

Here is our **system prompt**:

You are a table tennis coach. I have recorded video data and sensor data of a table tennis technique training session.

This session is called 'multi-ball' training, meaning the player strikes the ball using the same technique repeatedly for N times in a round.

Your task is to analyze the data I provide, each with a sequence of pose data and the corresponding sensor data, describing the behavior of the player around the stroke moments. You will also know the technique of the stroke as well.

<data-description>

Here is the explanation of the sensor data: - acc_x: Acceleration along the x-axis of the imu - agl_speed_x: Angular speed along the x-axis of the imu - agl_x: Angle along the x-axis of the imu - mgt_x: Magnetic field along the x-axis of the imu - quat_1: Quaternion component 1 </data-description>

<task> Based on the data, you need to provide the most important [topk] suggestions for the player to improve his technique. Each suggestion must describe only one aspect of the player's behavior and how to improve it. The suggestions must contain 3 parts: 'content', 'suggestion_key' and 'suggestion_type'. The 'suggestion_key' part in each suggestion must be unique and should be one of the predefined types. </task> <task-requirements> The 'content' part should be a sentence describing the suggestion.

The 'suggestion_key' part should be the key of the data that the suggestion is based on. The predefined types for'suggestion_key' are: [Predefined suggestion keys], you must not create a new one.

The 'suggestion_type' part should be the type of the suggestion. The predefined types for 'suggestion_type' are: Based on this information, you need to: - position: Indicates that the position of this key is incorrect. - strength: Indicates that the force exertion or explosiveness of this key is incorrect or insufficient. - stability: Indicates that the stability of this key is incorrect. - racket: Indicates that the relationship between this key and the racket is incorrect. - others: Represents other types of suggestions related to this key.

Please ensure that the output is in the JSON format as shown in the example:

<output format>

```json "suggestions_list": ["content": "suggestion content", "suggestion_type": "suggestion_type", "suggestion_key": "suggestion_key" # you should copy the 'suggestion_key' but not the 'data_key'.] ``` </output format> </task-requirements>

Here is our **data prompt**:

Here is the sequence of the sensor data. We provide only the data of stroke moment, and the average value + Standard deviation around the peak.

The input data is a sequence of sensor data, each with 16 attributes: [16 sensor attributes]

round_meta_info:[meta_info], stroke_number:[stroke number]

[sensor data of the stroke moment + static information around the stroke] Here is the sequence of the pose data. We provide only the data of stroke moment, and the average value + Standard deviation around the peak. The input data is a sequence of pose data, each with 3 attributes, describing the position of the joint: [3 position attributes] round_meta_info:[meta_info], stroke_number:[stroke number] [pose data of the stroke moment + static information around the stroke]

A.2.3 prompt for data cleaning. We cleaned the data with the assistance of LLMs. We leveraged LLMs to translate the original comments from coaches into English and summarize the comments into predefined keys and suggestion types. We use the system prompt to explain the task, describe the input format, and input coach comments record in JSON format as a user prompt.

Here is our **system prompt**:

"<task>You are a table tennis coach using English, now here is a JSON file containing some advice commenting a video,

which consists of a sequence of "start", "end", "text", "index", "comment_zh" and "comment_en", summarize the comments in the form 'joint:suggestion', the 'joint' here means the part of body the line of json comment on, and you should aggregate and clean the occurring keys, mapping them to the predefined categories in: NewAllowedSuggestionKey = [Predefined key list] that is, add a new field 'suggestion_type', which allows five values: position: Indicates that the position of this key is incorrect. Strength: Indicates that the force exertion or explosiveness of this key is incorrect or insufficient. Stability: Indicates that the stability of this key is incorrect. Racket: Indicates that the relationship between this key and the racket is incorrect. others: Represents other types of suggestions related to this key.Don't classify them as 'others' unless you really need to. Each record must only comment on one suggestion key, which means if the previous comments contain multiple 'joint:suggestion' pairs, you should split them separately. </task> <addition></addition> <example1>Here is an example, input: [input example] output:[output example] </example1>

<notation>every record in the output need to follow the structure : [output format] Other irrelevant variables should remain unchanged. Focus on processing "comment_en"</notation>"
```