F^{3}Set: Towards Analyzing Fast, Frequent, and Fine-grained Events from Videos

Zhaoyu Liu[1,2], Kan Jiang[2], Murong Ma[2], Zhe Hou[3], Yun Lin[4], Jin Song Dong[2]
[1]Ningbo University [2] National University of Singapore [3] Griffith University
[4] Shanghai Jiao Tong University
{liuzy, jiangkan}@nus.edu.sg, murongma@u.nus.edu
z.hou@griffith.edu.au, lin_yun@sjtu.edu.cn, dcsdjs@nus.edu.sg

Abstract

Analyzing Fast, Frequent, and Fine-grained (F^3) events presents a significant challenge in video analytics and multi-modal LLMs. Current methods struggle to identify events that satisfy all the F^3 criteria with high accuracy due to challenges such as motion blur and subtle visual discrepancies. To advance research in video understanding, we introduce F^{3}Set, a benchmark that consists of video datasets for precise F^3 event detection. Datasets in F^{3}Set are characterized by their extensive scale and comprehensive detail, usually encompassing over 1,000 event types with precise timestamps and supporting multi-level granularity. Currently F^{3}Set contains several sports datasets, and this framework may be extended to other applications as well. We evaluated popular temporal action understanding methods on F^{3}Set, revealing substantial challenges for existing techniques. Additionally, we propose a new method, F^{3}ED, for F^3 event detections, achieving superior performance. The dataset, model, and benchmark code are available at https://github.com/F3Set/F3Set.

1 Introduction

Recognizing sequences of fast (fast-paced), frequent (many actions in a short period), and fine-grained (diverse types) events with precise timestamps (with a tolerance of 1-2 frames) is a challenging problem for both current video analytics methods and multi-modal large language models (LLMs). Despite advancements in fine-grained action recognition [29; 51; 44], temporal action localization [53; 6; 38; 52], segmentation [60; 31; 64; 2], and video captioning [56; 49; 42; 34], limited focus has been directed towards this problem. This task is critical for various real-world applications, such as sports analytics, where action forecasting [20; 58], strategic and tactical analysis [10; 41], and player performance evaluation [11; 48] depend on a *detailed* understanding of event sequences. Other examples include industrial inspection [40], crucial for detecting subtle irregularities in high-speed production lines to ensure quality and safety; computer vision in autonomous driving [24], essential for accurate and instantaneous vehicle control and obstacle detection; and surveillance [46], important for the precise identification of abnormal or sudden events to enhance security. However, existing methods and datasets foundational to their development only *partially* address the F^3 scenario.

To facilitate the study of F^3 events understanding, we propose a new benchmark, F^{3}Set, for precise temporal events detection and recognition. F^{3}Set datasets usually have a large number of event types (on the order of 1,000), annotated with exact timestamps, and offer multi-level granularity to capture comprehensive event details. Although F^3 is a general problem, creating such a dataset requires domain-specific knowledge for labeling and processing, thus, in this paper, we use tennis as a case study. We also introduce a general annotation pipeline and toolchain to support domain experts in creating new F^3 datasets. Using this pipeline, we have also been building datasets for table tennis and badminton, and a community of users are actively expanding these with other applications.

Unlike other video analysis tasks, tennis actions are characterized by their rapid succession and diversity as illustrated in Figure 1. Understanding detailed event attributes like shot direction, technique, and outcome is essential. For instance, analyzing patterns in serve directions (e.g., “T”, “body”, “wide”, defined in Appendix B) or success rates can reveal a player’s habits and skill levels, offering strategic insights for competitive advantage. This detailed analysis supports coaches and players in developing tailored strategies against different opponents. However, detecting F^3 events

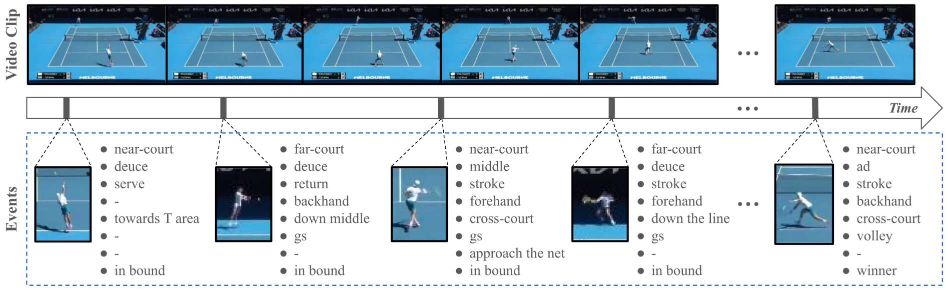


Figure 1: Example of detecting fast, frequent, and fine-grained events with precise moments.

from videos poses significant challenges, such as subtle visual differences, motion-induced blurring, and the need for precise event localization. Current video understanding methods are inadequately equipped to address these challenges. For instance, traditional fine-grained action recognition [51; 9] typically assigns a single label to an entire video rather than identifying a sequence of events. Temporal action localization (TAL) and temporal action segmentation (TAS) often depend on pre-trained or modestly fine-tuned input features [37; 15], which lack the specificity required to capture the subtle and domain-specific visual details necessary for recognizing diverse events with temporal precision. Some studies [23; 34] attempt to address these issues through *dense* frame sampling and end-to-end training. However, this makes targeted events temporally sparse (e.g., only a few events over hundreds of consecutive frames). As a result, long-term temporal correlation modules on dense visual features struggle to capture event-wise causal correlations effectively.

Moreover, Large Language Models (LLMs) [47; 54; 36] have expanded their capabilities to include multi-modal inference, encompassing text, visuals, and audio. Recognizing the potential, we conducted preliminary experiments on F^{3}Set using GPT-4 and observed that it understood basic video contexts, such as sports types, contextual information (e.g., court type and scoreboard), and simple actions. However, it struggles with understanding F^3 events and temporal relations between frames (e.g., shot directions). See Appendix A for details. Consequently, GPT-4 yields poor results compared to the other methods for F^3 problems, and we do not use it in the experiment. By introducing F^{3}Set, we hope it can help advance multi-modal LLM capabilities in F^3 video understanding in the future.

Leveraging F^{3}Set, we extensively evaluate existing temporal action understanding methods, aiming to reveal the challenges of F^3 event understanding. To provide guidelines for future research, we conduct a number of ablation studies on modeling choices. Addressing the shortcomings of existing methods, we also propose a simple yet efficient model, F^{3}ED, that is designed for F^3 event detection tasks and can be trained quickly on a single GPU. It outperforms existing models and can serve as a baseline for further development.

Contributions. The key contributions of this paper are as follows:

- We create F^{3}Set, a new benchmark with datasets that feature over 1,000 precisely timestamped event types with multi-level granularity, designed to challenge and advance the state-of-the-art in temporal action understanding.
- We introduce a general annotation toolchain that enables domain experts to create new F^3 datasets.
- We propose an end-to-end model named F^{3}ED, which can accurately detect F^3 event sequences from videos through visual features and contextual sequence refinement on a single GPU.
- We assess the performance of leading temporal action understanding methods on F^{3}Set through comprehensive evaluations and ablation studies and provide an analysis of the results.

2 Related Work

Existing F^3 related datasets. Although datasets have been developed for temporal action understanding, few focus on the F^3 events. Table 1 compares existing datasets with F^{3}Set by scale ("# Vid", "# Clips") and characteristics like action speed ("Evt. Len."), frequency ("Evt. / sec"), and granularity ("# Classes"), which correspond to "fast", "frequent", and "fine-grained" respectively. Datasets such as THUMOS14 [26] and Breakfast [28] focus on coarse-grained actions, where background context provides clear cues, and actions span seconds to minutes. In contrast, FineAction [39] and ActivityNet [4] cover a wide range of daily activities with diverse action categories, while FineGym [51] delves into detailed action types within gymnastics. Like FineGym, F^{3}Set emphasizes domain-specific

Table 1: Comparison of existing F^3 related datasets and F^3Set. “Evt. Len.” is the average duration of each event, and “# Evt. / sec” is the average number of events per second.

Datasets	# Vid.	# Clips.	Avg. Clip Len.	# Classes	Evt. Len.	# Evt. / sec
(a) Fine-grained						
FineAction [39]	-	16,732	149.5s	101	6.9s	0.3
ActivityNet [4]	-	19,994	116.7s	200	49.2s	0.01
FineGym [51]	303	32,697	50.3s	530	1.7s	0.3
(b) Fast						
CCTV-Pipe [40]	575	575	549.3s	16	< 0.1s	0.02
SoccerNetV2 [12]	9	9	99.6min	12	< 0.1s	0.3
(c) Frequent						
FineDiving [62]	135	3,000	4.2s	29	1.1s	$\sim$1
(d) Fast & Frequent						
ShuttleSet [59]	44	3,685	10.9s	18	< 0.1s	$\sim$1
P^2ANet [3]	200	2,721	360.0s	14	< 0.1s	$\sim$2
(d) Fast & Frequent & Fine-grained						
F^3Set	114	11,584	8.4s	1,108	< 0.1s	$\sim$1

granularity with subtle visual differences but encounters additional challenges due to faster and more frequent actions. Besides, unlike FineGym’s typical single-player focus, F^3Set (e.g., tennis) features two players and a fast-moving ball, with both players rapidly moving across the court, occupying only small portions of the scene, thus increasing task difficulty. CCTV-Pipe [40] targets temporal defect detection in urban pipe systems, providing single-frame annotations for rapid event detection, though it is limited in frequency and event types. Research in the sports domain has explored the detection of fast and frequent actions. FineDiving [62] segments diverse diving events, while ShuttleSet [59] and P^2ANet [3] focus on identifying strokes in fast-paced racket sports. Volleyball [25] and NSVA (basketball) [61] focus on team sports understanding and video captioning, while SoccerNetV2 [12] ball action spotting task focus on identifying the timing and type of ball-related actions. However, these datasets typically cover coarser event types and are limited to specific F^3 aspects.

In contrast, our proposed F^3Set is characterized by 1) *rapid* events that occur instantaneously, 2) *high frequency* of approximately one event per second, and 3) *extensive granularity* with a larger number of detailed event classes. These attributes introduce novel challenges.

F^3 event understanding Detecting F^3 events poses unique challenges due to their rapid temporal dynamics, high occurrence rates, and subtle visual distinctions, requiring precise temporal and contextual understanding. Fine-grained action detection has been explored in tasks covering diverse daily activities [4; 39], using features extracted by video encoders pre-trained on datasets like Kinetics-400 [27] and a detection head for classification. However, such pre-trained extractors often miss domain-specific nuances. Domain-specific methods in FineGym [51] and FineDiving [51] utilize end-to-end training to incorporate domain knowledge. These methods often encode videos into non-overlapping snippets or downsample frames, yielding coarse temporal features insufficient for detecting fast-paced events spanning only 1–2 frames. Related works such as ShuttleSet [59] and P^2ANet [3] address fast and frequent event detection in racket sports by employing end-to-end models that extract frame-wise features and use detection heads (e.g., BMN [35] or GRU [8]) to classify each frame. To address class imbalance, the loss weight of the foreground classes is set higher than the background during training [23]. While these approaches achieve precise temporal spotting, their scalability to larger action classes is limited by challenges like long-tail class distributions and inadequate modeling of event-wise correlations. Our proposed F^3ED overcomes these issues through frame-wise dense processing, a multi-label classification head to handle minor event differences and class imbalances, and a contextual module to refine predictions by leveraging event-wise causal relationships, enhancing both precision and robustness in F^3 event detection.

3 F^3SET: A BENCHMARK DATASET FOR F^3 EVENT DETECTION

Recognizing the limitations in existing video datasets for F^3 event understanding, we introduce F^3Set, a new benchmark for precise temporal F^3 events detection and recognition. Given the need for

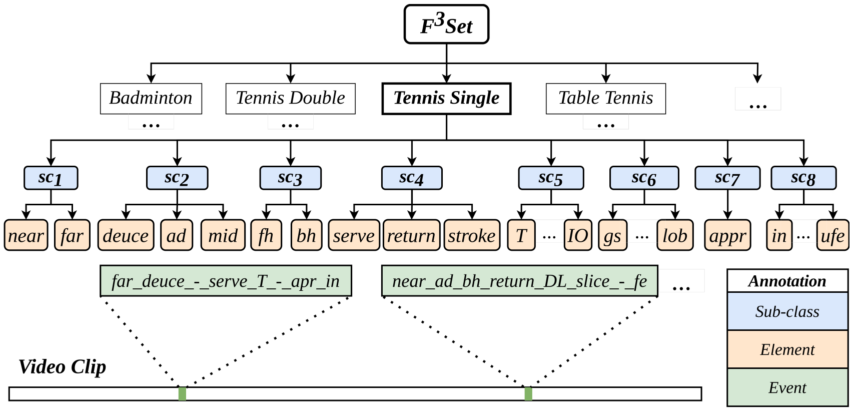


Figure 2: Breakdown of F^{3}Set event class annotation.

domain-specific expertise in creating F^3 datasets, this section uses **tennis** as a **case study** to illustrate F^{3}Set's event descriptions, construction process, and key properties. We also propose a general annotation pipeline and toolchain that empowers domain experts to develop new F^3 datasets for diverse applications. Applying the same approach, we have also built F^3 datasets for **other domains**, including tennis doubles, badminton, and table tennis (see link).

3.1 F^{3}SET EVENT DESCRIPTION

We use **tennis** to illustrate F^3 event descriptions, introducing key lexicon and defining F^3 events. Datasets have been built for **other F^{3}domains**, including tennis doubles, badminton, and table tennis, with similar event definitions. Details are in Appendix C.

Lexicon. A tennis court is divided into deuce, middle, and ad regions. The initial shot, a "serve," targets the T, Body (B), or Wide (W) areas. A "return" follows if the receiver's shot lands in bounds. Subsequent shots, or "strokes," can be directed "cross-court" (CC), "down the line" (DL), "down the middle" (DM), "inside-in" (II), or "inside-out" (IO) using either "forehand" (fh) or "backhand" (bh). Players may "approach" (apr) the net on shorter balls. Shot techniques include "ground stroke/top spin" (gs), "slice," "volley," and "lob," with outcomes: "in-bound," "winner," "forced error," or "unforced error.". More detailed definitions can be found in Appendix B.

F^3 events. Formally, each event (tennis) consists of 8 *sub-classes*, denoted as $sc_1, sc_2, ..., sc_8$:

sc_1 – *hit by which player*: (1) near- or (2) far-end player;

sc_2 – *hit from which court location*: (3) deuce, (4) middle, or (5) ad court;

sc_3 – *hit at which side of the body*: (6) forehand or (7) backhand;

sc_4 – *shot type*: (8) serve, (9) return, or (10) stroke;

sc_5 – *shot direction*: (11) T, (12) B, (13) W, (14) CC, (15) DL, (16) DM, (17) II, or (18) IO;

sc_6 – *shot technique*: (19) gs, (20) slice, (21) volley, (22) lob, (23) drop, or (24) smash;

sc_7 – *player movement*: (25) approach;

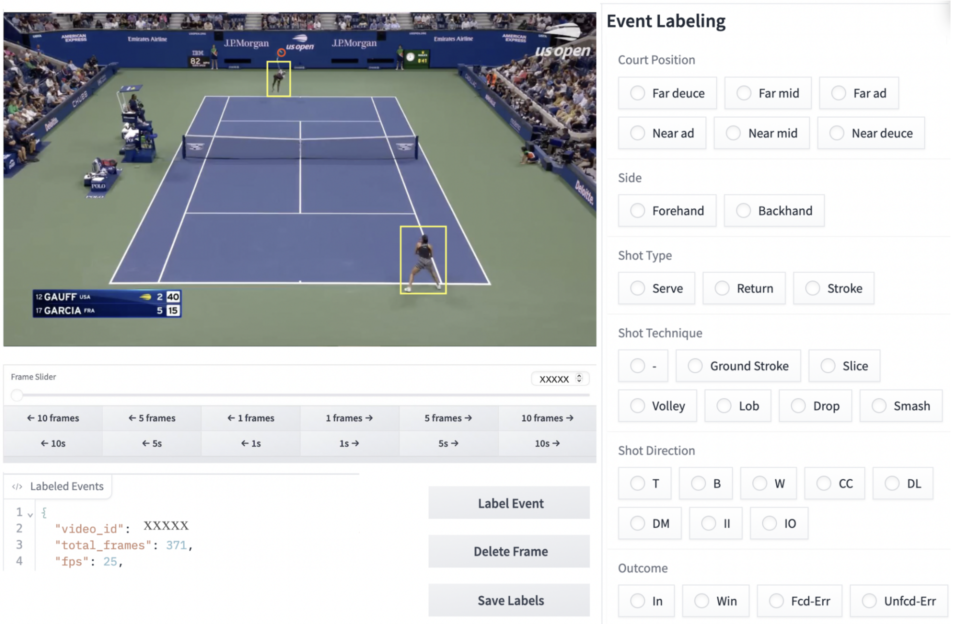
sc_8 – *shot outcome*: (26) in, (27) winner, (28) forced error, or (29) unforced error.

Altogether, there are 29 *elements* and 1,108 *event types* based on various combinations (Figure 2).

Similarly, for other domains, badminton contains 6 *sub-classes*, 28 *elements* and 1008 *event types*; table tennis contains 7 *sub-classes*, 23 *elements* and 1296 *event types*; and tennis doubles contain 26 *elements* and 744 *event types*. Compared to existing racket sports video datasets [3; 59], F^{3}Set offers additional dimensions, such as shot direction and outcomes, which are crucial for identifying playing patterns and success rates. Please refer to Appendix C for more details.

3.2 F^{3}SET DATASET CONSTRUCTION

Video collection. For tennis, we collected publicly available high-resolution singles matches (2012–2023) from YouTube, including Grand Slams, Olympics, and major ATP/WTA tournaments. The dataset includes various court surfaces (hard, clay, grass), male and female players, and both right- and left-handed competitors. These videos feature complete rallies, match footage, and detailed player data. Similar criteria were used for tennis doubles, badminton, and table tennis videos.



```
{
  "video_id": XXXXX
  "total_frames": 371,
  "fps": 25,
```

Figure 3: An interface of the labeling tool. The panel on the right is application-customizable.

Annotation pipeline and toolchain. After data collection, we use a three-stage annotation process designed to maximize automation and minimize manual effort. This pipeline is adaptable to various sports broadcast videos and broader domains:

(1) Video segmentation: The first stage is to segment a full broadcast video into shorter clips using a context-aware scene detector [1] that automatically identifies jump cuts within the video.

(2) Clip selection: The second stage is to select targeted clips (e.g., clips contain tennis rallies) using a Siamese network to compare each clip with a "base image" indicative of the scene of interest.

(3) F^3 event annotation: The final stage is to identify the precise event moments (e.g., frames when a player hits the ball) and record the corresponding event types through an annotation tool.

The first two steps are automated and applicable to a range of sports videos, facilitating the efficient breakdown of lengthy videos into relevant clips. For the final phase, we developed an interactive annotation interface, shown in Figure 3. The tool allows users to navigate clips quickly (e.g., 1-second increments) or review them frame by frame, enabling efficient identification of key events (e.g., hitting moments). It supports selecting shot types and identifying court positions through direct clicks on the video, with each click displayed for immediate verification. Object-level detection can assist the process, and a foolproof design minimizes errors from accidental clicks or misjudgments. This tool is adaptable to other sports by incorporating domain-specific knowledge, broadening its applicability.

Our annotation team consists of 8 members. We provided them with specialized training and rigorous pre-tests before beginning the official annotation work, along with supporting materials such as slides and demonstrations. Each annotator was assigned an equal portion of the dataset, totaling 1,450 clips (rallies) each. The manual labeling takes roughly 30 hours to finish all 1,450 clips. Following the initial annotation phase, we conducted multiple rounds of cross-validation involving random sampling of rallies and quality checks among annotators to ensure the accuracy of the event-based labels. In cases where conflicting annotations arose, annotators were asked to input the labels they believed to be correct. The final label was determined based on a majority vote among the annotators.

3.3 F^{3}SET DATASET STATISTICS AND PROPERTIES

Key statistics for F^{3}Set tennis dataset are summarized in Table 2. Statistics for other F^3 datasets, including badminton, table tennis, and tennis doubles, are provided in the Appendix D. We employ a training, validation, and testing split of 3:1:1, with the training and validation sets drawn from the same video sources, while the test set features clips from distinct videos.

Event Timestamp. Unlike typical TAL and TAS tasks, where an action spans several frames or seconds, the duration of actions in racket sports is often ambiguous. Thus, stroke actions are defined as instantaneous events, recording only the moment of ball-racket contact [55] as shown in Figure 1.

Table 2: Summary of F^{3}Set tennis dataset statistics.

Category	Details
Matches	114 broadcast matches
Players	75 (30 men, 45 women)
Handedness	68 right-handed, 7 left-handed
Frame Rate (FPS)	25–30 FPS

Category	Details
Clips	11,584 rallies
Avgerage Clip Duration	8.4 sec
Total Shots	42,846
Shots Per Rally	1 to 34

Multi-level granularity. Depending on the requirements of the analytics task, F^{3}Set can focus on a subset of sub-classes, enabling flexible granularity. We define a parameter $G \in \mathcal{P}(\{sc_1, \dots, sc_8\})$, where $\mathcal{P}(\{sc_1, \dots, sc_8\})$ is the power set of $\{sc_1, \dots, sc_8\}$, to select sub-classes and form different levels of granularity. We define 3 granularity levels using F^{3}Set tennis as an example. At the coarse level, $G_{\text{low}} = \{sc_1, sc_3, sc_4, sc_8\}$ includes 4 sub-classes, 11 elements, and 38 event types. This level captures essential but broad information. At a finer level, $G_{\text{mid}} = \{sc_1, \dots, sc_6\}$ consists of 6 sub-classes, 24 elements, and 365 event types. This granularity provides more detailed event representations. At the most detailed level, $G_{\text{high}} = \{sc_1, \dots, sc_8\}$ encompasses all 8 sub-classes, 29 elements, and 1,108 event types. This level is ideal for precise and comprehensive event analysis. This multi-level granularity enhances F^{3}Set's flexibility for diverse real-world tasks.

3.4 Ethical Considerations

F^{3}Set is constructed from publicly available sports broadcasts, ensuring compliance with ethical and legal standards. We do not redistribute video content, providing only YouTube links to maintain adherence to copyright policies. The dataset focuses on professional players in public tournaments, avoiding private or off-court data and ensuring it is used strictly for academic research. While anonymization is not applied, as these players are public figures, we emphasize that the dataset should not be used for non-research purposes. A more detailed discussion on privacy, consent, and bias mitigation is provided in Appendix E.

4 Our Proposed Approach: F^{3}ED

Acknowledging the challenges and limitations of existing approaches, we propose a simple yet effective method named **F**ast **F**requent **F**ine-grained **E**vent **D**etection network (F^{3}ED), illustrated in Figure 4. It is designed for F^3 event detection and can serve as a baseline for further development.

Problem formulation. Let $X \in \mathbb{R}^{H \times W \times 3 \times N}$ denote the input, consisting of N RGB frames of size $H \times W$. The output is a sequence of M event-timestamp pairs $((E_1, t_1), \dots, (E_M, t_M))$, where E_i is the event type with C classes and t_i is the corresponding timestamp for $i \in \{1, \dots, M\}$. Additionally, each event E_i can also be expressed as a vector $[e_{i,1}, \dots, e_{i,K}]$, with each element $e_{i,j} \in \{0, 1\}$ indicating the presence or absence of the j^{th} element in event E_i, where j is an integer $j \in \{1, \dots, K\}$. The parameter K, which defines the *number of elements* in each event vector.

Video Encoder (VE). The first stage of both baselines and our model will extract spatial-temporal frame-wise features. The video encoder (VE) consists of a visual backbone, followed by a bidirectional GRU to capture long-term visual dependencies: $\mathbf{F}_{emb} = \text{VE}(X)$, with $\mathbf{F}_{emb} \in R^{N \times d'}$.

Event Localizer (LCL). Utilizing the frame-wise features $\mathbf{F}_{emb}$, the event localizer (LCL) employs a fully connected network with a Sigmoid activation function to perform dense binary classification, aiming to accurately identify specific event instances. For an N-frame clip, the output is represented as $(\hat{p}_1, \dots, \hat{p}_N)$, where each $\hat{p}_i$ denotes the probability that an event occurs at the corresponding timestamp: $(\hat{p}_1, \dots, \hat{p}_N) = \textit{Sigmoid}(\text{LCL}(\mathbf{F}_{emb}))$. Ground truth labels $(p_1, \dots, p_N)$ with $p_i \in \{0, 1\}$ are used to compute the discrepancy between the predicted probabilities and the actual values using binary cross-entropy loss as: $L_{LCL} = \frac{1}{N} \sum_{i=1}^{N} p_i \cdot log(\hat{p}_i) + (1 - p_i) \cdot log(1 - \hat{p}_i)$.

Multi-label Event Classifier (MLC). Upon detecting events, we proceed to categorize them into specific types using a multi-label classification module (MLC). This module, a fully connected network, takes the identified event features f_i from $\mathbf{F}_{emb}$ as inputs to predict the event types: $\hat{E}_i = \textit{Sigmoid}(\text{MLC}(f_i)) = [\hat{e}_{i,1}, \dots, \hat{e}_{i,K}]$, where K denotes the number of elements, f_i represents the features for the event at the i^{th} frame, $\hat{E}_i$ is the predicted event type, and $\hat{e}_{i,j} \in [0, 1]$ is the probability of $\hat{E}_i$ containing the j^{th} element. For a video clip with M events, the ground truths are

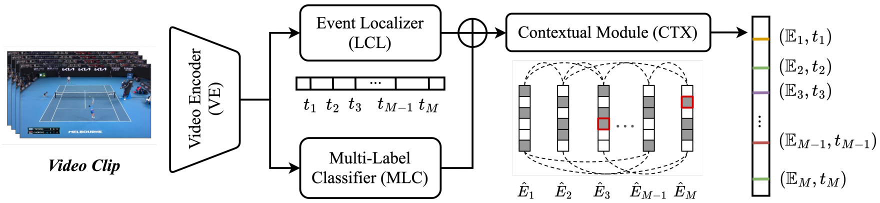


Figure 4: Overview of F^{3}ED. RGB images are processed by VE to capture frame-wise spatial-temporal features, which are passed to LCL to identify event timestamps and MLC to predict labels. Outputs from LCL and MLC are combined ('plus' symbol) to form an event representation sequence and refined by CTX module. 'Red squares' represent errors from purely visual predictions.

given as $(E_1, \ldots, E_M)$ with each E_i represented as a vector of K elements $[e_{i,1}, \ldots, e_{i,K}]$. The loss can be represented by $L_{MLC} = \frac{1}{M}\sum_{i=1}^{M}(\frac{1}{K}\sum_{j=1}^{K} e_{i,j} \cdot log(\hat{e}_{i,j}) + (1 - e_{i,j}) \cdot log(1 - \hat{e}_{i,j}))$.

Contextual module (CTX) Video encoders often struggle to extract insightful visual features from fast-paced videos due to motion blur, and objects of interest, such as players, may only occupy a small portion of the frame. This can result in the loss of crucial visual details for fine-grained action classification, particularly when resizing images to 224×224. Selecting the best-predicted event types naively might, therefore, produce invalid event sequences. To address this, we introduce a contextual module (CTX), designed to concurrently learn contextual knowledge from event sequences during end-to-end training: $(\mathbb{E}_1, \ldots, \mathbb{E}_M) = \text{CTX}(\hat{E}_1, \ldots, \hat{E}_M)$. CTX employs a bidirectional GRU to process the predicted event sequence $\hat{E}$ and outputs a refined sequence $\mathbb{E}_i = [\mathbb{e}_1, \ldots, \mathbb{e}_k]$, integrating both visual-based predictions and contextual correlations across events. The loss is calculated for each refined event: $L_{CTX} = \frac{1}{M}\sum_{i=1}^{M}(\frac{1}{K}\sum_{j=1}^{K} e_{i,j} \cdot log(\mathbb{e}_{i,j}) + (1 - e_{i,j}) \cdot log(1 - \mathbb{e}_{i,j}))$.

5 Experiments

In this section, we benchmark existing temporal action understanding methods, including TAL, TAS, and TASpot, on the F^{3}Set dataset and conduct a series of ablation studies.

Evaluation metrics. The evaluation metrics used in our work are carefully chosen to comprehensively assess both the temporal precision and classification accuracy of detected events, which are critical for F^3 event detection. These metrics align with evaluation standards in similar tasks. **Edit Score** [30] measures the similarity between predicted and ground truth event sequences using Levenshtein distance, capturing errors in event sequence structure, such as missing, additional, or misordered events. This metric is particularly valuable for evaluating models where the temporal order and completeness of event sequences are essential [22]. **Mean F1 Score with Temporal Tolerance** evaluates both classification and temporal localization accuracy [23; 22]. By considering a prediction correct only when its timestamp aligns within a strict temporal tolerance (e.g., ± 1 frame) and its class correctly identifies, this metric ensures that models are assessed on their ability to achieve precise temporal spotting alongside accurate classification. Given the long-tail distribution of event types in the dataset, where some events are extremely rare, we report two variants of the mean F1 score to ensure a balanced evaluation: $F1_{evt}$, the average F1 score across all event types, and $F1_{elm}$, the average F1 score across all elements, which typically presents a more balanced distribution.

Baselines. Existing temporal action understanding frameworks typically incorporate two key components: a *video encoder* for visual feature extraction and a *head module* for specific tasks such as detection or segmentation. Applying these models directly to our study presents challenges, as they generally utilize a two-stage training process—employing a static, pre-trained video encoder for feature extraction and training only the head module. This approach often fails to capture fine-grained, domain-specific events due to its reliance on temporally coarse, non-overlapping, or downsampled video segments. To address these limitations, we have adapted these temporal action understanding methods to develop new baselines better suited for detecting F^3 events. Given the rapid pace and short duration of tennis shots, it is crucial to utilize frame-wise feature extraction [7] (discussed in Section 5.2). Besides, end-to-end training with video encoder fine-tuning is required to capture the subtle event differences. Moreover, the classification of some sub-classes (e.g., shot direction, outcome) demands long-term temporal reasoning to integrate information from subsequent frames

Consequently, we focus on three established feature extractors: TSN [57], TSM [33], and SlowFast [19], which are known for their efficiency in frame-wise feature extraction and end-to-end training.

Table 3: Experimental results on F^{3}Set (tennis) with 3 levels of granularity. Full table in Appendix G.

		F^{3}Set (G_{high})			F^{3}Set (G_{mid})			F^{3}Set (G_{low})		
Video encoder	Head arch.	F1$_{evt}$	F1$_{elm}$	Edit	F1$_{evt}$	F1$_{elm}$	Edit	F1$_{evt}$	F1$_{elm}$	Edit
TSN [57]	MS-TCN [18]	15.9	59.8	53.5	23.2	60.9	65.8	45.7	70.4	72.8
	ActionFormer [65]	18.4	60.6	55.2	24.8	61.9	67.3	48.7	70.6	72.2
	E2E-Spot [23]	24.7	65.3	60.1	31.5	66.2	71.0	53.5	73.6	75.0
SlowFast [19]	G-TAD [63]	23.0	66.1	64.0	29.6	66.5	74.2	53.3	76.0	77.9
	ActionFormer [65]	28.7	70.0	67.6	35.5	70.9	76.4	59.3	77.1	81.5
	E2E-Spot [23]	25.9	69.4	65.7	33.8	70.4	75.4	55.5	76.5	79.5
I3D [5]	E2E-Spot [23]	22.7	59.7	68.7	27.1	60.7	74.2	51.9	67.7	78.3
VTN [45]	E2E-Spot [23]	14.8	58.3	56.7	20.0	59.4	68.2	39.7	63.1	73.1
TSM [33]	MS-TCN [18]	21.7	67.3	58.6	30.4	69.5	73.0	50.2	74.0	75.3
	ASformer [64]	17.6	61.9	57.5	25.5	64.0	74.2	46.0	72.9	74.0
	G-TAD [63]	16.9	62.5	55.2	29.8	66.9	74.8	39.8	70.1	67.2
	ActionFormer [65]	22.4	65.7	60.3	31.0	68.2	74.7	52.4	73.8	74.9
	E2E-Spot [23]	31.4	71.4	68.7	39.5	72.3	77.9	60.6	78.4	82.1
TSM[33]	F^{3}ED	**40.3**	**75.2**	**74.0**	**48.0**	**76.5**	**82.4**	**68.4**	**80.0**	**87.2**

We pair each encoder with five representative head module architectures from existing methods: MS-TCN [18] and ASFormer [64] from TAS, G-TAD [63] and ActionFormer [65] from TAL and E2E-Spot [23] from TASpot, to establish a set of new baseline models for our study. To identify hitting moments and their respective event types, frame-wise dense *multi-class* classification is applied to identify each frame as either background or one of the event types.

Implementation details. We implement and train models on F^{3}Set in an end-to-end manner. The video encoder takes video clip X down-scaled and cropped to 224 × 224 to extract frame-wise visual features. Subsequently, each head module processes per-frame features to identify a sequence of F^3 events and their timestamps. For more implementation details, please refer to Appendix F.

5.1 Results and Analysis

The evaluation results presented in Table 3 provide several critical insights into the performance of various methods across different levels of granularity (G_{low}, G_{mid}, and G_{high}). A general trend emerges where performance decreases as granularity increases, underscoring the growing challenges associated with finer granularity. While certain methods demonstrate some robustness, the overall efficacy across all approaches remains suboptimal, particularly at higher levels of granularity, indicating the challenge of precise F^3 event detection task.

Simple 2D CNNs (e.g., TSN), which process frames independently, are inadequate for F^3 event detection due to their inability to capture critical spatial-temporal correlations between frames. Lacking temporal modeling, they struggle to distinguish visually similar events, resulting in poor performance, especially at higher granularity levels. Advanced video encoders such as I3D [5], SlowFast [19], and transformer-based VTN [45], which excel in other video understanding tasks, face significant challenges with F^{3}Set. These models process video data using techniques like non-overlapping snippets or frame downsampling, resulting in coarse temporal features. While effective for long-duration actions, such approaches struggle to detect the rapid, short-duration events in F^3, which rely on precise temporal cues spanning only 1–2 frames. This suggests that increasing video encoder complexity does not necessarily improve performance for fast-action detection in F^{3}Set. Notably, simpler models like TSM, paired with advanced 2D CNNs such as RegNet-Y [50], outperform these complex encoders. This highlights the importance of capturing subtle visual differences over short temporal spans, demonstrating that the ability to extract fine-grained temporal cues is more impactful than model complexity.

Head modules such as transformer-based ActionFormer, and GRU-based E2E-Spot, generally outperform other methods. This advantage highlights their effectiveness in capturing long-term temporal dependencies through end-to-end training. Notably, E2E-Spot consistently outperforms ActionFormer across most settings, suggesting that GRU-based architectures may offer an advantageous trade-off between efficiency and representational power for certain types of temporal correlations.

Our proposed F^{3}ED model, leveraging the TSM video encoder, achieves the best performance among all granularity levels. This is attributable to two key design choices: the multi-label classifier and the contextual module. Detailed discussions of these design elements are presented in the next section.

Table 4: Ablation and analysis experiments. The default model takes stride size 2 and clip length 96.

	$\text{F}^3\text{Set}\,(G_{high})$			$\text{F}^3\text{Set}\,(G_{mid})$			$\text{F}^3\text{Set}\,(G_{low})$		
Experiment	F1_{evt}	F1_{elm}	Edit	F1_{evt}	F1_{elm}	Edit	F1_{evt}	F1_{elm}	Edit
TSM + E2E-Spot	31.4	71.4	68.7	39.5	72.3	77.9	60.6	78.4	82.1
(a) Feature extractor									
I3D [5] (clip-wise)	22.7	59.7	68.7	27.1	60.7	74.2	51.9	67.7	78.3
VTN [45] (video transformer)	14.8	58.3	56.7	20.0	59.4	68.2	39.7	63.1	73.1
ST-GCN++ [16] (skeleton-based)	25.4	62.1	56.1	32.4	63.9	63.5	55.1	69.4	73.2
PoseConv3D [17] ((skeleton-based))	20.1	54.5	53.2	26.0	55.4	61.9	48.8	63.0	69.7
(b) Stride size = 4	25.9	69.2	62.7	33.4	69.9	73.0	60.0	77.9	78.8
Stride size = 8	14.0	56.7	44.3	18.5	57.4	54.8	40.4	67.0	59.2
(c) without GRU	27.6	69.0	60.6	38.0	71.3	75.3	54.7	74.1	73.4
(d) Clip Length = 32	26.3	67.4	54.5	35.5	69.4	71.8	53.2	75.1	68.9
Clip Length = 64	30.7	71.2	67.4	38.6	72.4	77.5	58.4	77.9	81.1
Clip Length = 192	29.3	70.3	65.7	37.3	71.4	77.0	58.8	77.1	80.4
(e) Multi-label	37.9	74.3	71.7	45.9	75.6	80.1	66.6	80.1	85.1
(f) Multi-label + CTX (Transformer)	39.0	74.3	72.8	50.5	75.5	81.8	63.4	79.6	86.8
Multi-label + CTX (BiGRU)	40.3	75.2	74.0	48.0	76.5	82.4	68.4	80.0	87.2

5.2 Ablation Study

We selected the highest-performing baseline model (TSM + E2E-Spot) as our default configuration for the subsequent ablation studies. More ablation studies can be found in Appendix H.

Feature extractor. An effective feature extractor is crucial for accurate F^3 event detection. Below, we summarize some key findings (details in Appendix H). *(1) Frame-wise feature extraction outperforms clip-wise methods*, which divide inputs into non-overlapping segments. Experiments show clip-wise methods produce temporally coarse features and hinder precise event detection. *(2) Transformer-based video encoders* such as VTN [45] struggle on F^3Set due to high computational costs and limited ability to effectively capture short-term temporal correlations. *(3) In addition to RGB inputs, we also experimented with skeleton-based pose estimation methods*, including ST-GCN++ [16] and PoseConv3D [17] with human key points as input. While they excel in efficiency and interpretability, they lack critical details like shot direction, limiting performance on F^3Set.

Sparse sampling. Increasing the stride size allows for a broader temporal coverage within a fixed sequence length. This sparse sampling technique is prevalent in many video understanding tasks [38; 32], offering high efficiency and reasonable accuracy. However, this approach proves inadequate for our task, where events are characterized by their rapid occurrence, frequency, and fine granularity. As illustrated in Table 4(b), increasing the stride size to 4 and 8 leads to a marked decline in performance, underscoring the importance of dense sampling for detecting F^3.

Long-term temporal reasoning. The default model employs a spatio-temporal video encoder (TSM), complemented by a bidirectional Gated Recurrent Unit [14] (GRU) head for enhanced long-term temporal integration. To assess the necessity of long-term temporal reasoning, we replaced the GRU module with a fully connected layer. The results, presented in Table 4(c), indicate a significant performance decline relative to the original configuration. This finding highlights the essential role of long-term temporal reasoning in analyzing sub-classes such as shot direction, outcomes, and player movements that require information from subsequent frames.

Clip length. The sensitivity of sequence models to varying input clip lengths, which encapsulate different temporal contexts, is notable. In F^3Set, the incidence of F^3 events correlates directly with clip length. Table 4(d) shows that shorter clips result in fewer events per sequence, hindering the model's ability to leverage long-term dependencies among consecutive events effectively. Conversely, while longer clip lengths yield improved results, the marginal gains diminish with increasing length.

Multi-class versus multi-label classification. The challenge of modeling over 1,000 possible event type combinations as a multi-class classification problem is formidable. For example, consider two events, E_1 (far_ad_bh_stroke_DL_*slice*_apr_in) and E_2 (far_ad_bh_stroke_DL_*drop*_apr_in), which differ only in shot technique (*slice* vs. *drop*). Although similar, multi-class classification treats these as distinct classes, thus reducing training efficiency and exacerbating the long-tail distribution bias towards more frequent classes. A more natural approach is multi-label classification, where each event can belong to multiple sub-class elements (e.g., ['far', 'ad', 'serve', 'W', 'in']). Thus, $E1$ and

Table 5: Experimental results on other “semi-F^3” datasets.

	ShuttleSet [59]		FineDiving [62]		FineGym [51]		SoccerNetV2 [12]		CCTV-Pipe [40]	
Head arch.	F1$_{evt}$	Edit	F1$_{evt}$	Edit	F1$_{evt}$	Edit	F1$_{evt}$	Edit	F1$_{evt}$	Edit
MS-TCN [18]	70.3	74.4	65.7	92.2	57.6	65.3	43.4	74.5	25.8	31.3
ASformer [64]	55.9	70.6	49.9	87.6	53.6	66.3	46.3	76.1	15.4	33.4
G-TAD [63]	48.2	61.1	52.1	82.6	45.8	51.4	42.3	72.3	31.3	33.6
ActionFormer [65]	62.1	67.5	68.3	92.4	54.0	59.7	43.0	64.6	18.8	29.5
E2E-Spot [23]	70.2	75.0	75.8	93.7	62.1	65.4	46.2	72.9	27.2	35.2
F^{3}ED	70.7	77.1	77.6	95.1	70.9	70.7	48.1	76.6	37.0	39.5

$E2$ only differ in shot technique but are identical in other aspects. This adjustment facilitates more effective training and shows an increase in performance, as shown in Table 4(e).

Contextual knowledge. Beyond the statistical results in Table 3, analysis of predicted event sequences reveals that current baselines may produce invalid sequences due to logical errors or uncommon practices. For instance, a right-handed player cannot logically direct a forehand shot from the deuce court as “II” or “IO”. Similarly, an event ending in a winner or error should logically conclude the sequence. Additionally, it is uncommon for a player to hit with backhand when the ball is played to their forehand side. Further examples are detailed in Appendix I. These observations indicate that existing baselines fail to effectively capture event-wise contextual correlations. By adding the CTX module, the performance further increases as shown in Table 4(f). We also compared BiGRU and Transformer Encoder for the CTX module. BiGRU performed slightly better, likely due to its efficiency in modeling short *event sequences* (usually <20 per clip) with fewer parameters.

5.3 Generalizability to “semi-F^3” data

F^3 task possesses broad applicability across numerous real-world domains, such as sports, autonomous driving, surveillance, and production line inspection. Nevertheless, creating such a F^3 dataset necessitates substantial expertise and extensive labeling efforts. We have found that existing video datasets often fail to fully address all three dimensions of the F^3 task—“fast”, “frequent”, and “fine-grained”. In this section, we conducted experiments on several “semi-F^3” datasets that partially meet these criteria, including Shuttleset [59] for badminton (racket sport), FineDiving [62] for diving (individual sports), FineGym [51] for gymnastics (individual sports), SoccerNetV2 [43] (team sports), and CCTV-Pipe [40] for pipe defect detection (industrial application). We report only the F1$_{evt}$ and Edit score, as not all datasets necessitate multi-label classification given their limited event types. For the video encoder, we chose TSM, which consistently outperforms the others on average.

Performance across different domains can vary significantly depending on the difficulty of tasks and the scale of datasets. For instance, the CCTV-Pipe dataset, targeting temporal defect localization in urban pipe systems, shows suboptimal performance due to factors such as ambiguous single-frame annotations for each defect, multiple defects at the same time, long-tailed distribution of defect types, and limited dataset size. Our performance is better than the results reported in [40]. Generally, methods that effectively handle F^{3}Set tend to perform well across other applications, as indicated in Table 5. Our F^{3}ED outperforms existing baselines in all datasets, demonstrating its robust generalizability for detecting “semi-F^3” events across various domains. While F^3 event detection benefits from accurate event localization, a high-performing LCL module is not a hard prerequisite (see Appendix J). Therefore, our method can be generalized and benefit broader applications.

6 Conclusion and Future Work

In this study, we addressed the challenge of analyzing fast, frequent, and fine-grained (F^3) events from videos by introducing F^{3}Set, a benchmark for precise temporal F^3 event detection. F^{3}Set datasets usually feature detailed event types (approximately 1,000), annotated with precise timestamps, and provide multi-level granularity. We have also developed a general annotation toolchain that enables domain experts to create F^3 datasets, thereby facilitating further research in this field. Moreover, we proposed F^{3}ED, an end-to-end model that effectively detects complex event sequences from videos, using a combination of visual features and contextual sequence refinement. Our comprehensive evaluations and ablation studies of leading methods in temporal action understanding on F^{3}Set highlighted their performance and provided critical insights into their capabilities and limitations. Moving forward, we aim to extend the scope of F^3 task to more real-world scenarios and advance the development of F^3 video understanding.

REFERENCES

[1] Pyscenedetect. https://www.scenedetect.com/.

[2] Nadine Behrmann, S Alireza Golestaneh, Zico Kolter, Juergen Gall, and Mehdi Noroozi. Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In *European Conference on Computer Vision*, pp. 52–68. Springer, 2022.

[3] Jiang Bian, Xuhong Li, Tao Wang, Qingzhong Wang, Jun Huang, Chen Liu, Jun Zhao, Feixiang Lu, Dejing Dou, and Haoyi Xiong. P2anet: A large-scale benchmark for dense action detection from table tennis match broadcasting videos. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(4):1–23, 2024.

[4] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 961–970, 2015.

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

[6] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1130–1139, 2018.

[7] Minghao Chen, Fangyun Wei, Chong Li, and Deng Cai. Frame-wise action representations for long videos via sequence contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13801–13810, 2022.

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 720–736, 2018.

[10] Tom Decroos, Jan Van Haaren, and Jesse Davis. Automatic discovery of tactics in spatio-temporal soccer match data. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, pp. 223–232, 2018.

[11] Tom Decroos, Lotte Bransen, Jan Van Haaren, and Jesse Davis. Actions speak louder than goals: Valuing player actions in soccer. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1851–1861, 2019.

[12] Adrien Deliege, Anthony Cioppa, Silvio Giancola, Meisam J Seikavandi, Jacob V Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B Moeslund, and Marc Van Droogenbroeck. Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *The IEEE/CVF conference on computer vision and pattern recognition*, pp. 4508–4519, 2021.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

[14] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pp. 1597–1600. IEEE, 2017.

[15] Guodong Ding, Fadime Sener, and Angela Yao. Temporal action segmentation: An analysis of modern techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[16] Haodong Duan, Jiaqi Wang, Kai Chen, and Dahua Lin. Pyskl: Towards good practices for skeleton action recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 7351–7354, 2022.

[17] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2969–2978, 2022.

[18] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *The IEEE/CVF conference on computer vision and pattern recognition*, pp. 3575–3584, 2019.

[19] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *The IEEE/CVF international conference on computer vision*, pp. 6202–6211, 2019.

[20] Panna Felsen, Pulkit Agrawal, and Jitendra Malik. What will happen next? forecasting player moves in sports videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 3342–3351, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[22] Yuchen He, Zeqing Yuan, Yihong Wu, Liqi Cheng, Dazhen Deng, and Yingcai Wu. Vistec: Video modeling for sports technique recognition and tactical analysis. *arXiv preprint arXiv:2402.15952*, 2024.

[23] James Hong, Haotian Zhang, Michaël Gharbi, Matthew Fisher, and Kayvon Fatahalian. Spotting temporally precise, fine-grained events in video. In *European Conference on Computer Vision*, pp. 33–51. Springer, 2022.

[24] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 954–960, 2018.

[25] Mostafa S Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. In *The IEEE conference on computer vision and pattern recognition*, pp. 1971–1980, 2016.

[26] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos “in the wild”. *Computer Vision and Image Understanding*, 155:1–23, 2017.

[27] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[28] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. In *Proc. IEEE Winter Applications of Computer Vision Conference (WACV 16)*, Lake Placid, Mar 2016.

[29] Colin Lea, René Vidal, and Gregory D Hager. Learning convolutional action primitives for fine-grained action recognition. In *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 1642–1649. IEEE, 2016.

[30] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 156–165, 2017.

[31] Zhe Li, Yazan Abu Farha, and Jurgen Gall. Temporal action segmentation from timestamp supervision. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8365–8374, 2021.

[32] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3320–3329, 2021.

[33] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *The IEEE/CVF international conference on computer vision*, pp. 7083–7093, 2019.

[34] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17949–17958, 2022.

[35] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3889–3898, 2019.

[36] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[37] Meng Liu, Liqiang Nie, Yunxiao Wang, Meng Wang, and Yong Rui. A survey on video moment localization. *ACM Computing Surveys*, 55(9):1–37, 2023.

[38] Xiaolong Liu, Song Bai, and Xiang Bai. An empirical study of end-to-end temporal action detection. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20010–20019, 2022.

[39] Yi Liu, Limin Wang, Yali Wang, Xiao Ma, and Yu Qiao. Fineaction: A fine-grained video dataset for temporal action localization. *IEEE transactions on image processing*, 31:6937–6950, 2022.

[40] Yi Liu, Xuan Zhang, Ying Li, Guixin Liang, Yabing Jiang, Lixia Qiu, Haiping Tang, Fei Xie, Wei Yao, Yi Dai, et al. Videopipe 2022 challenge: Real-world video understanding for urban pipe inspection. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 4967–4973. IEEE, 2022.

[41] Zhaoyu Liu, Kan Jiang, Zhe Hou, Yun Lin, and Jin Song Dong. Insight analysis for tennis strategy and tactics. In *2023 IEEE International Conference on Data Mining (ICDM)*, pp. 1175–1180. IEEE, 2023.

[42] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

[43] Hassan Mkhallati, Anthony Cioppa, Silvio Giancola, Bernard Ghanem, and Marc Van Droogenbroeck. Soccernet-caption: Dense video captioning for soccer broadcasts commentaries. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5073–5084, 2023.

[44] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 122–132, 2020.

[45] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3163–3172, 2021.

[46] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, pp. 3153–3160. IEEE, 2011.

[47] OpenAI. Gpt-4 technical report, 2023.

[48] Luca Pappalardo, Paolo Cintia, Paolo Ferragina, Emanuele Massucco, Dino Pedreschi, and Fosca Giannotti. Playerank: data-driven performance evaluation and player ranking in soccer via a machine learning approach. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(5):1–27, 2019.

[49] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8347–8356, 2019.

[50] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

[51] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *The IEEE/CVF conference on computer vision and pattern recognition*, pp. 2616–2625, 2020.

[52] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. Tridet: Temporal action detection with relative boundary modeling. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18857–18866, 2023.

[53] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1049–1058, 2016.

[54] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[55] Roman Voeikov, Nikolay Falaleev, and Ruslan Baikulov. Ttnet: Real-time temporal and spatial video analysis of table tennis. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 884–885, 2020.

[56] Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. Reconstruction network for video captioning. In *The IEEE conference on computer vision and pattern recognition*, pp. 7622–7631, 2018.

[57] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018.

[58] Wei-Yao Wang, Hong-Han Shuai, Kai-Shiang Chang, and Wen-Chih Peng. Shuttlenet: Position-aware fusion of rally progress and player styles for stroke forecasting in badminton. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 4219–4227, 2022.

[59] 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. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5126–5136, 2023.

[60] Zhenzhi Wang, Ziteng Gao, Limin Wang, Zhifeng Li, and Gangshan Wu. Boundary-aware cascade networks for temporal action segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pp. 34–51. Springer, 2020.

[61] Dekun Wu, He Zhao, Xingce Bao, and Richard P Wildes. Sports video analysis on large-scale data. In *European Conference on Computer Vision*, pp. 19–36. Springer, 2022.

[62] Jinglin Xu, Yongming Rao, Xumin Yu, Guangyi Chen, Jie Zhou, and Jiwen Lu. Finediving: A fine-grained dataset for procedure-aware action quality assessment. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2949–2958, 2022.

[63] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10156–10165, 2020.

[64] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. *arXiv preprint arXiv:2110.08568*, 2021.

[65] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision*, pp. 492–510. Springer, 2022.

A Preliminary Results for Multi-Modal LLMs

We conducted experiments using GPT4 (47), specifically employing the GPT4-vision-preview feature. As depicted in Figure 5, we input a sequence of 12 consecutive frames from a tennis rally into the system. This sequence encompasses a total of 6 shots:

- wide serve by the near player,
- forehand return down the middle by the far player
- backhand stroke cross-court by the near player
- backhand stroke cross-court by the far player
- forehand stroke inside-in by the near player
- forehand stroke cross-court by the far player (last shot).

Our initial tests are aimed at assessing GPT4-vision-preview's ability to decipher basic contextual information. As demonstrated in Table 6, GPT4 demonstrates proficiency in recognizing the type of sport, deciphering information from the scoreboard, and identifying the tournament category. However, it exhibits some inaccuracies in determining the exact game points.

Table 6: User Prompt and GPT4's Response for Question 1. The text highlighted in bold represents correct responses, underlined text indicates incorrect responses, and text in red within brackets is the correction.

User These are images from a video. What information can you get?
GPT4 The images you provided are **sequential frames from a tennis match**. Based on the information visible on the scoreboard in the images, it seems to be an intense moment with the following details: • The **score is at deuce**, indicating that both players have won the same number of points after each having scored **at least three points in this particular game**. • The player labeled **'Djokovic' is leading the match**, having won the first set 7-6 (7-5) and is ahead in the second set with a score of 4-2 (6-2). • The location is marked as **Melbourne**, which suggests that this match might be **taking place during the Australian Open**, a major tennis tournament that is held annually in Melbourne. • The players are **engaged in a rally, with one player serving and the other returning the ball**. The images show a snapshot of the action, **capturing the movement and intensity of professional tennis play**.

However, when we shift our focus to more complex queries requiring deeper insight, as shown in Table 7, we observe a notable disparity. Although GPT4 generates coherent responses, its precision in recognizing and interpreting the fine-grained sequence of actions within the rally is limited. As there are too many incorrect responses, we do not use GPT4 in the experiment for comparison.

B Tennis Lexicon.

In this section, we will provide an overview of technical tennis terms used in our model based on official terminology from the USTA[1]. A standard tennis court is depicted in Fig. 6. Each side of the court can be divided into three regions: the deuce court (red), the middle court (green), and the ad court (blue). Additionally, the service box is defined by the boundaries of the net, service line, center service line, and single sideline. It includes three sub-regions, the *T*, *B*, and *W* areas.

In tennis, the initial shot of a point is called a *serve*. The ball must land in the service box and is usually hit overhead. On odd points, the server serves to the receiver's right-hand side, named *deuce court serve*; while on even points, the server serves to the receiver's left-hand side, named *ad court serve*. The server can aim to hit the ball in the *T*, *B*, or *W* area of the service box.

[1] https://www.usta.com/en/home/improve/tips-and-instruction.html

Table 7: User Prompt and GPT4's Response for Question 2. The text highlighted in bold represents correct responses, underlined text indicates incorrect responses, and text in red within brackets is the correction.

User
These images show the video of a whole tennis rally between two right-hand players. Describe what's likely going on in each frame. Try to find connections across frames and describe the shot type (serve, forehand, backhand) and direction (T, wide, body, cross-court, down the line, down the middle, inside-in, inside-out).

GPT4
1. One player has just **served** the ball. The serve appears to be going towards the T (center) (wide) of the service box on the opponent's side.
2. The receiving player is getting ready to return the serve with a backhand stroke (forehand), suggesting the serve was directed to his backhand side.
3. The receiver has hit the ball back, likely aiming cross-court (down the middle) based on their body position and the angle of the racket at impact.
4. The server has **moved towards the center** of the baseline in preparation to return the shot, suggesting they **anticipated a cross-court return**.
5. The server hits the ball, potentially with a forehand (backhand) stroke, aiming the shot either cross-court or down the line (cross-court), as they are positioned on the deuce side of the court.
6. The **ball is in transit**, and the players are positioning themselves for the next shot. The **receiver is moving to his left**, probably anticipating a ball to his **forehand** side.
7. The receiver is setting up for a **forehand return**, indicating the previous shot was indeed to his forehand.
8. The ball is struck, with the player's body and racket angle indicating a possible **cross-court** shot.
9. The original server is on the run to the opposite side, preparing for a possible backhand return (forehand), which suggests the previous shot was a cross-court forehand (backhand).
10. The player makes contact with a backhand stroke (forehand), likely sending the ball either down the line or cross-court (cross-court) given their body orientation.
11. The players reposition again for the next shot, with the server **transitioning toward the center**, preparing for either a forehand or backhand.
12. The server prepares for a backhand return (forehand), suggesting the ball was returned to his backhand side.

The shot taken by the receiver after a serve is called a *return*. If it lands in bounds after crossing the net. Subsequent shots are referred to as *strokes*. Players can hit the ball cross-court, down the line, down the middle, inside-in or inside-out using either their *forehand* (fh) or *backhand* (bh). *Cross-court* (CC) means a shot that travels diagonally from the player's position (e.g., a right-hand player's forehand shot from his deuce/middle court to the opponent's deuce court). *Down the line* (DL) refers to a straight shot from their position (e.g., a right-hand player's backhand shot from his ad court to the opponent's deuce court). *Down the middle* (DM) means a shot toward the opponent's middle court. *Inside-out* (IO) / *inside-in* (II) refers to a player changing the shot from backhand to forehand or vice versa. For example, if a right-hand player hits a forehand shot from his ad/middle court to the opponent's ad court, this is called a forehand inside-out. Similarly, if a right-hand player hits a forehand shot from his ad court to the opponent's deuce court, this is called a forehand inside-in.

Generally, a player can approach the net on a ball that lands around the service line or shorter, or if they recognize that their opponent is out of position and is likely to provide a weak ball. This is called an *approach* (apr) shot, which is defined as an offensive shot that allows a tennis player to transition from the baseline to the net, hitting a forehand or a backhand. If a player choose to stay at the baseline or already positioned at the net, we call it - (n-apr). Players can also apply different shot techniques under certain conditions. A *ground stroke* (gs) is a basic tennis shot executed after the ball bounces once on the court, typically used from the baseline. A *slice* is a shot where the player imparts a backspin on the ball, causing it to travel slowly and with a lower trajectory, which can disrupt an opponent's timing. A *volley* is hit before the ball bounces, usually performed near the net to shorten the point. A *lob* is a shot that sends the ball high and deep into the opponent's court, often used to counter opponents who are close to the net.

Each shot has four possible outcomes: *in-bound (in)*, where the ball lands within the opponent's court boundaries; *winner (win)*, a shot that is successfully placed where the opponent cannot return it, directly winning the point; *forced error (fe)*, where the shot is so challenging that the opponent makes

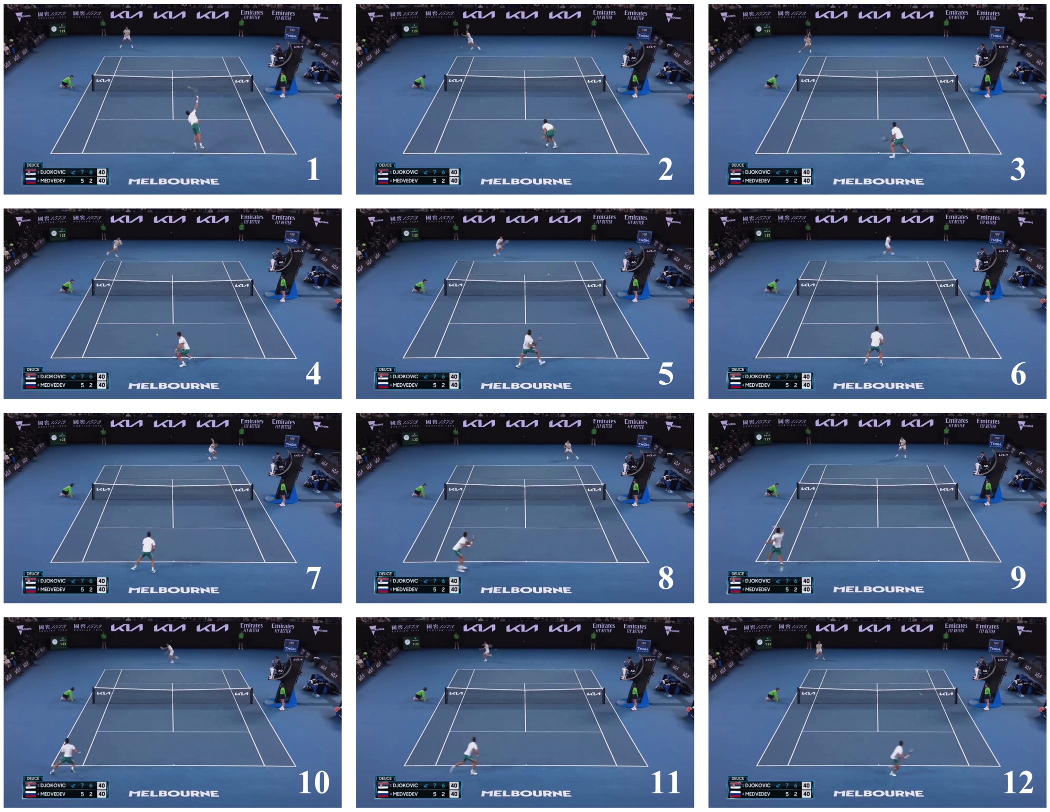


Figure 5: Video frames from a tennis rally.

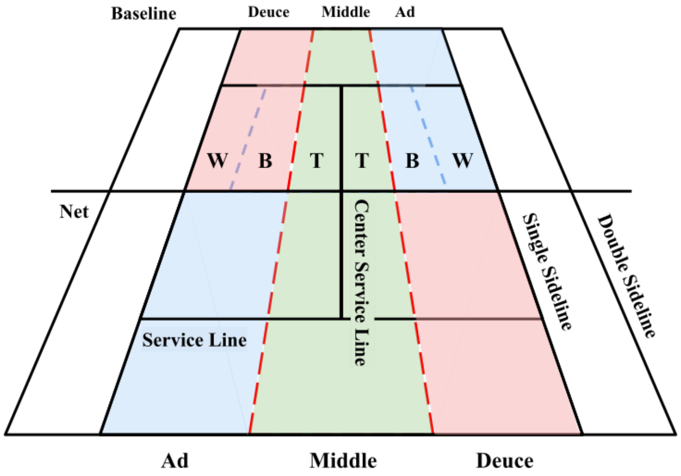


Figure 6: Tennis court and related terminologies.

an error trying to return it, often due to the pressure exerted by the aggressive play; and *unforced error (ufe)*, where the player fails to return the ball in the court without external pressure, typically due to a mistake in execution.

C F^3 Events in Other Domains

For the badminton dataset, each event consists of 6 *sub-classes*, denoted as $sc_1, sc_2, ..., sc_6$:

- sc_1 – *hit by which player*: (1) near- or (2) far-end player;
- sc_2 – *hit from which court location*: (3) left, (4) middle, or (5) right court;
- sc_3 – *hit at which side of the body*: (6) forehand or (7) backhand;
- sc_4 – *shot type*: (8) serve-short, (9) serve-long, (10) net, (11) smash, (12) lob, (13) clear, (14) drive, (15) drop, (16) push or (17) rush;
- sc_5 – *shot direction*: (18) T, (19) B, (20) W, (21) CC, (22) DL, or (23) DM;
- sc_6 – *shot outcome*: (24) in, (25) winner, or (26) error.

Altogether, there are 26 *elements* and 1,008 *event types* based on various combinations.

For the table tennis dataset, each event consists of 7 *sub-classes*, denoted as $sc_1, sc_2, ..., sc_7$:

- sc_1 – *hit by which player*: (1) near- or (2) far-end player;
- sc_2 – *hit from which court location*: (3) left, (4) middle, or (5) right court;
- sc_3 – *hit at which side of the body*: (6) forehand or (7) backhand;
- sc_4 – *shot spin*: (8) top, (9) bottom, or (10) side;
- sc_5 – *shot type*: (11) serve, (12) push, (13) chop, (14) drive, (15) block, or (16) smash;
- sc_6 – *shot direction*: (17) straight-long, (18) straight-short, (19) diagonal-long, or (20) diagonal-short;
- sc_7 – *shot outcome*: (21) in, (22) winner, or (23) error.

Altogether, there are 23 *elements* and 1,296 *event types* based on various combinations.

For the tennis double dataset, each event consists of 7 *sub-classes*, denoted as $sc_1, sc_2, ..., sc_6$:

- sc_1 – *hit by which player*: (1) near- or (2) far-end player;
- sc_2 – *hit from which court location*: (3) deuce or (4) ad court;
- sc_3 – *hit at which side of the body*: (5) forehand or (6) backhand;
- sc_4 – *shot type*: (7) serve, (8) return, (9) volley, (10) lob, (11) smash, or (12) swing;
- sc_6 – *shot direction*: (13) T, (14) B, (15) W, (16) CC, (17) DL, (18) II, or (19) IO;
- sc_7 – *serving formation*: (20) conventional, (21) I-formation, (22) Australian, or (23) non-serve.
- sc_7 – *shot outcome*: (24) in, (25) winner, or (26) error.

Altogether, there are 26 *elements* and 744 *event types* based on various combinations.

For the full elements and event types, please refer to `https://github.com/F3SET/F3Set/blob/main/data/`.

D Data Statistics

We recognize the importance of testing generalizability under diverse real-world conditions, including variations in camera angles, court types, weather, and illumination. While professional competition videos are filmed under relatively standardized conditions, we have ensured that F^3Set captures a significant level of diversity in these factors:

- Camera Angles: The dataset includes videos from 114 broadcast matches across various tournaments, each exhibiting different camera angles. Additionally, individual matches often feature multiple perspectives: standard bird's-eye view and low-angle view shots from behind one player, where the foreground player appears significantly larger than the background player.
- Court Types: The dataset covers all three tennis court surfaces—hard court, clay court, and grass court—with diverse color schemes (e.g. blue, green, red, black, green, purple, etc).
- Weather Conditions and Illumination: The videos in F^3Set reflect diverse weather and lighting scenarios, including day and night matches, cloudy weather, indoor and outdoor games, and challenging conditions such as partial sunlight casting shadows on the court. For outdoor matches, some videos feature strong sunlight on parts of the court, making it harder to track the ball or

Table 8: Distribution of elements across sub-classes in the F^{3}Set.

Sub-Class	Element	Count	Proportion (%) in Sub-Class
sc_1	*near*	21,467	50.1%
	far	21,362	49.9%
sc_2	*deuce*	14,474	33.8%
	ad	16,310	38.1%
	middle	12,045	28.1%
sc_3	*forehand*	27,802	64.9%
	backhand	15,027	35.1%
sc_4	*serve*	11,584	27.0%
	return	8,216	19.2%
	stroke	23,029	53.8%
sc_5	*T*	4,428	10.3%
	Body	2,241	5.2%
	Wide	4,915	11.5%
	cross-court	11,933	27.9%
	down the line	3,521	8.2%
	down the middle	11,040	25.8%
	inside-in	608	1.4%
	inside-out	4,143	9.7%
sc_6	*ground stroke*	38,287	89.4%
	slice	3,358	7.8%
	volley	497	1.2%
	lob	334	0.8%
	drop	236	0.5%
	smash	117	0.3%
sc_7	*approach*	964	2.3%
	non-approach	41,865	97.7%
sc_8	*in-bound*	31,245	73.0%
	winner	3,734	8.7%
	forced error	2,808	6.5%
	unforced error	5,042	11.8%

Table 9: Summary of F^{3}Set badminton dataset statistics.

Category	Details
Matches	10 broadcast matches
Players	16 (16 men)
Handedness	13 right-handed, 3 left-handed
Frame Rate	25–30 FPS
Clips	112 rallies
Average Clip Duration	13.5 seconds
Total Shots	1692
Shots Per Rally	1 to 62

players in those areas. Indoor matches also vary in brightness, with some tournaments having brighter lighting setups and others being relatively dimmer.

To further enhance the diversity, we have collected videos from college-level matches for tennis doubles. We are also actively expanding F^{3}Set by incorporating more matches from platforms like UTR[2] and junior-level competitions using our annotation toolchain.

[2] https://www.utrsports.net/

Table 10: Summary of F^{3}Set table tennis dataset statistics.

Category	Details
Matches	5 broadcast matches
Players	9 (4 men, 5 women)
Handedness	9 right-handed
Frame Rate	25–30 FPS
Clips	42 rallies
Average Clip Duration	5.9 seconds
Total Shots	361
Shots Per Rally	2 to 36

Table 11: Summary of F^{3}Set tennis doubles dataset statistics.

Category	Details
Matches	8 broadcast matches
Players	24 (24 men)
Handedness	24 right-handed
Frame Rate	25–30 FPS
Clips	78 rallies
Average Clip Duration	6.0 seconds
Total Shots	645
Shots Per Rally	2 to 21

D.1 ADDITIONAL F^{3}ED STATISTICS

Key statistics for F^{3}Set badminton dataset are summarized in Table 9; for F^{3}Set table tennis dataset are summarized in Table 10; for F^{3}Set tennis doubles dataset are summarized in Table 11.

Furthermore, we present additional statistics related to the F^{3}Set (tennis single) datasets. Table 8 details the frequency and proportional occurrence of elements within each sub-class. For the full elements and event types, please refer to `https://github.com/F3SET/F3Set/blob/main/data/f3set-tennis/`.

E ETHICAL CONSIDERATIONS FOR F^{3}SET DATASETS

The F^{3}Set is constructed from publicly available video data sourced from YouTube, specifically from officially broadcasted tennis tournaments featuring professional players. This document outlines the ethical considerations related to data collection, copyright compliance, privacy concerns, and bias mitigation.

The dataset consists of publicly available sports broadcasts, ensuring compliance with ethical and legal standards. We do not store or distribute local copies of the videos unless explicitly permitted under Creative Commons or similar licenses. If a video is removed or becomes unavailable, we update our dataset accordingly while ensuring adherence to copyright policies. To comply with YouTube's Terms of Service, we provide only video URLs, ensuring that the content remains under the control of rights holders.

Our dataset exclusively features professional players in widely broadcasted tournaments. As such, the dataset includes individuals who are already in the public domain through official broadcasts. No private or off-court data is collected, and annotations focus solely on event-based information. The dataset is strictly intended for research purposes, and users must ensure ethical compliance in their applications. A disclaimer is provided, explicitly stating that the dataset should not be used beyond academic research.

We take steps to mitigate potential biases in the dataset. The dataset does not incorporate or filter data based on nationality or ethnicity, ensuring a broad and representative scope. We encourage users to evaluate and report any potential biases that may emerge in model training. Additionally, the dataset will be periodically reviewed and updated based on community feedback to ensure fairness.

The F^{3}Set dataset has been designed to align with ethical standards, ensuring responsible use of publicly available content. We have addressed concerns regarding copyright, privacy, and bias. By emphasizing academic-only use and compliance with licensing terms, we provide a valuable resource for sports analytics research while respecting the rights and privacy of all involved stakeholders.

F Implementation Details

F.1 Baseline models

- *TSN* [57] utilizes a purely 2D CNN architecture. Each frame is processed independently with RGB images as inputs, employing the RegNet-Y 200MF architecture [50] as the backbone.
- *TSM* [33] incorporates a temporal shift mechanism within the 2D convolutional process of video encoders. This shift along the temporal axis mimics a cost-free 1D convolution, enabling efficient extraction of spatiotemporal features through subsequent convolutions on shifted inputs. The RegNet-Y 200MF backbone is augmented with Temporal Shift Modules (TSM) [33], integrated at strategic points within each residual block, specifically targeting a quarter of the channels, adjusted to the nearest multiple of four.
- *SlowFast* [19] features dual pathways: the slow pathway processes frames at a sparse rate to capture high-level information, and the fast pathway processes at a higher frame rate with fewer channels to capture detailed motion information efficiently. These pathways are integrated at various stages to enhance the assimilation of motion information. We utilize the SlowFast network as our video encoder, specifically the "SlowFast 4×16, R50" variant. This model samples N and $N/8$ frames in their fast and slow pathways, respectively, which are then resized to length N and concatenated.

The selection of RegNet-Y 200MF over traditional ResNet models [21] is based on its recent advancements, lower parameter count (3.2M compared to ResNet-18's 11.7M), and superior performance in image classification benchmarks [13]. This architecture allows flexibility for integrating alternative 2D CNN designs. Beginning with pretrained ImageNet-1K weights [13], the encoder is meticulously fine-tuned to our specific dataset needs.

Let J denotes the number of event types.

- *MS-TCN* [18] employs successive layers of dilated convolutions to capture long-range temporal dependencies in sequence modeling tasks. We adapt the code from [18], using dilated temporal convolution networks. We use 3 TCN stages for our MS-TCN baselines and a depth of 5 layers for each stage. Each layer has dimension of 256. Per-frame predictions are made with a fully connected layer that maps from 256 to $J + 1$, where each frame is classified as either background or one of the event types.
- *ASFormer* [64] leverages a transformer-based architecture with segment embeddings to enhance temporal action segmentation. We use code and settings from the implementation by [64].
- *G-TAD* [63] employs a graph convolutional network to model complex temporal relationships between video segments, enhancing the accuracy of action detection. We use the GCNeXt block architecture proposed by [63], which produces a dimensional feature encoding H of 384, 384, and 768 for TSN, TSM, and SlowFast, respectively, for each frame. Per-frame predictions are made with a fully connected layer mapping from H to $J + 1$.
- *ActionFormer* [65] employs transformer networks for efficient single-shot temporal action localization, using multiscale features and local self-attention. We employ the architecture proposed by [65] and produce a dimensional feature encoding H of 384, 384, and 768 for TSN, TSM, and SlowFast, respectively. Per-frame predictions are made with a fully connected layer mapping from H to $J + 1$.
- *E2E-Spot* [23] utilizes a bidirectional GRU layer to facilitate long-term temporal reasoning for precise action spotting. We use a 1-layer bidirectional GRU [14] with dimensions H of 384, 384, and 768 for TSN, TSM, and SlowFast, respectively. Per-frame predictions are made with a fully connected layer, from H to $J + 1$.

F.2 F^{3}ED

Training. We train all components of F^{3}ED in an end-to-end manner. The video encoder, equipped with a 2D CNN (i.e., RegNet-Y 200MF) and TSM, is initialized using pre-trained ImageNet-1K weights [13] and subsequently fine-tuned the targeted dataset. The LCL processes frame-wise spatio-temporal features to perform dense predictions, distinguishing event instances from the background. The MLC receives ground truth event instances to concentrate on the classification task, assuming accurate localization. Classification is executed densely, yet losses are computed only on frames that contain event instances. For the CTX, we input the predicted event sequences. The sequence is obtained by combining the outputs from LCL, MLC, and ground truth event locations. We then feed the predicted sequence to CTX to obtain a refined one. Overall, our composite loss function is defined as $L = L_{LCL} + L_{MLC} + L_{CTX}$.

Inference. During the inference phase, the MLC uses localization results from the LCL to predict corresponding event types. The CTX processes the event sequence generated from the LCL and MLC outputs, producing a new sequence of the same length that incorporates both visual predictions and contextual correlation across events.

F.3 Model Implementation Details

For both baseline models and the proposed F^{3}ED, the training protocol processes sequences of 96 frames with a stride of 2. Batch size is set to 4. Standard data augmentation techniques, including cropping and color adjustments, are applied during training to enhance data diversity and improve model robustness; these augmentations are omitted during testing. Input frames are resized to a height of 224 pixels, followed by a random crop to a 224 × 224 square, ensuring preservation of essential visual information by selectively adjusting the width. Techniques such as cropping and color jittering are employed to further augment the dataset and fortify the models against overfitting.

Each model performs dense, per-frame classification to identify event types and their precise timestamps. Given the imbalance in event distribution, where less than 3% of frames correspond to specific event instances, the loss weight for foreground classes is increased fivefold relative to background classes to address this disparity.

The models are optimized using the AdamW optimizer, with a learning rate schedule controlled via cosine annealing. Training is conducted over 50 epochs, with each epoch taking approximately 10 minutes on an RTX 4090 GPU. The initial learning rate is set to 0.001, with three linear warm-up steps before transitioning to a cosine decay schedule. For computationally intensive video encoders, such as SlowFast and VTN, a smaller initial learning rate of 0.0001 is used to ensure stable convergence.

G Additional Experimental Results

In this section, we provide the full experimental result table as shown in Table 12.

H Additional Ablation Studies

In this section, we provide additional ablation studies as well as more implementation details.

H.1 Frame-wise versus Clip-wise Feature Extractor

We would like to clarify the advantages of frame-wise over clip-wise feature extraction.

- Temporal Precision: To use a clip-wise feature extractor, we can divide the input video into non-overlapping segments and extract one feature vector per segment, which is a common technique in many TAL and TAS tasks. To investigate this, we conducted an experiment where 96-frame video clips were divided into 6-frame segments, with features extracted using I3D [5] for each segment. The resulting 16 feature vectors were interpolated back to 96 frames using PyTorch's F.interpolate function. As shown in Table 4(a), this approach produces temporally coarse features, leading to inadequate performance in precise event detection tasks.
- Efficiency and Scalability: An alternative approach is to stride a clip-wise feature extractor to obtain per-frame feature densely. However, this approach introduces significant computational overhead as each frame is processed multiple times in overlapping windows. This overhead makes end-to-end feature learning or fine-tuning impractical. In contrast, our frame-wise approach processes each

Table 12: Full experimental results on F^{3}Set (tennis) with 3 levels of granularity.

		F^{3}Set (G_{high})			F^{3}Set (G_{mid})			F^{3}Set (G_{low})		
Video encoder	Head arch.	F1$_{evt}$	F1$_{elm}$	Edit	F1$_{evt}$	F1$_{elm}$	Edit	F1$_{evt}$	F1$_{elm}$	Edit
TSN [57]	MS-TCN [18]	15.9	59.8	53.5	23.2	60.9	65.8	45.7	70.4	72.8
	ASformer [64]	11.9	54.3	49.8	17.3	56.1	62.5	40.3	67.3	70.3
	G-TAD [63]	6.0	47.5	24.7	14.1	52.1	48.6	19.9	57.4	44.7
	ActionFormer [65]	18.4	60.6	55.2	24.8	61.9	67.3	48.7	70.6	72.2
	E2E-Spot [23]	24.7	65.3	60.1	31.5	66.2	71.0	53.5	73.6	75.0
SlowFast [19]	MS-TCN [18]	17.2	63.1	56.2	24.3	65.5	70.3	47.4	73.1	73.5
	ASformer [64]	14.1	60.8	55.3	20.3	62.8	69.4	44.8	72.9	71.9
	G-TAD [63]	23.0	66.1	64.0	29.6	66.5	74.2	53.3	76.0	77.9
	ActionFormer [65]	28.7	70.0	67.6	35.5	70.9	76.4	59.3	77.1	81.5
	E2E-Spot [23]	25.9	69.4	65.7	33.8	70.4	75.4	55.5	76.5	79.5
I3D [5]	E2E-Spot [23]	22.7	59.7	68.7	27.1	60.7	74.2	51.9	67.7	78.3
VTN [45]	E2E-Spot [23]	14.8	58.3	56.7	20.0	59.4	68.2	39.7	63.1	73.1
TSM [33]	MS-TCN [18]	21.7	67.3	58.6	30.4	69.5	73.0	50.2	74.0	75.3
	ASformer [64]	17.6	61.9	57.5	25.5	64.0	74.2	46.0	72.9	74.0
	G-TAD [63]	16.9	62.5	55.2	29.8	66.9	74.8	39.8	70.1	67.2
	ActionFormer [65]	22.4	65.7	60.3	31.0	68.2	74.7	52.4	73.8	74.9
	E2E-Spot [23]	31.4	71.4	68.7	39.5	72.3	77.9	60.6	78.4	82.1
TSM[33]	F^{3}ED	**40.3**	**75.2**	**74.0**	**48.0**	**76.5**	**82.4**	**68.4**	**80.0**	**87.2**

frame only once, enabling the training of much longer sequences (hundreds of frames) in an end-to-end manner on a single GPU.

H.2 Skeleton-based Feature Extractor

We recognize the potential of using human pose estimation for representation learning and its ability to generalize to other domains. To explore this, we conducted additional experiments leveraging skeleton-based representations for F^3 event detection in F^{3}Set. We used MMPose[3] to extract player key points from original 1280x720 resolution images, generating skeleton data. Two advanced skeleton feature extractors ST-GCN++ [16] (GCN-based) and PoseConv3D [17] (CNN-based) were evaluated. The extracted skeleton features were processed with the F^{3}ED head architecture for classification and localization. The results are summarized in Table 13.

Key findings include:

- Among the two skeleton-based methods, ST-GCN++ demonstrated better overall performance.
- Visual features extracted from RGB images using TSM consistently outperformed skeleton-based methods in all three granularities. This is likely because many event types in F^{3}Set include information such as shot direction and shot outcomes, which skeletal data cannot capture.
- Skeleton-based methods excel in computational efficiency and interpretability, requiring fewer parameters and offering faster inference, while directly highlighting player movements and poses.

While skeleton-based approaches may not fully match the performance of RGB-based models for F^{3}Set, they offer unique benefits, particularly in terms of speed and transparency. We plan to further investigate skeleton-based methods and their integration with visual features in future work.

H.3 Input image resolution

We conducted additional experiments to analyze the effects of using different resolution inputs on model performance. The results are summarized in Table 14.

First, we evaluated F^{3}Set using TSM as the video encoder with input resolutions of 224×224, 336×336, and 448×448. The results show a consistent improvement in performance as resolution increases, though the gains diminish at higher resolutions (e.g., 448×448). This suggests that while

[3] https://mmpose.readthedocs.io/en/latest/

Table 13: Skeleton-based method compared with TSM + F^3ED. "Params(M)" refers to the number of model parameters, and "Inference time (ms)" refers to the per-frame inference time on a Nvidia RTX 4090 GPU.

			F^3Set (G_{high})			F^3Set (G_{mid})			F^3Set (G_{low})		
Experiment	Params (M)	Inference time (ms)	$F1_{evt}$	$F1_{elm}$	Edit	$F1_{evt}$	$F1_{elm}$	Edit	$F1_{evt}$	$F1_{elm}$	Edit
TSM + F^3ED	5.6	10.6	40.3	75.2	74.0	48.0	76.5	82.4	68.4	80.0	87.2
ST-GCN++ [16]	2.3	4.0	25.4	62.1	56.1	32.4	63.9	63.5	55.1	69.4	73.2
PoseConv3D [17]	6.8	6.4	20.1	54.5	53.2	26.0	55.4	61.9	48.8	63.0	69.7

Table 14: Ablation study on input image resolution. "Params(M)" refers to the number of model parameters. "FLOPs" refers to floating-point operations per second.

					F^3Set (G_{high})		
Video encoder	Head arch.	Resolution	Params (M)	FLOPs	$F1_{evt}$	$F1_{elm}$	Edit
TSM	F^3ED	224×224	5.6	77.3	40.3	75.2	74.0
TSM	F^3ED	336×336	5.6	77.3	43.2	77.1	74.8
TSM	F^3ED	448×448	5.6	77.3	44.4	78.1	74.5
SlowFast 4×16	F^3ED	336×336	52.7	494.2	37.4	73.7	71.3
SlowFast 8×8	F^3ED	336×336	52.8	903.7	41.0	74.6	75.0

higher resolutions can provide additional visual details, the marginal benefits decrease beyond a certain point.

We also tested F^3ED with SlowFast as an encoder on higher-resolution inputs (336×336). We experimented with two variants: SlowFast 4×16 and SlowFast 8×8. Despite higher complexity, SlowFast 4×16 underperformed TSM, likely due to its lower temporal resolution, which limits its ability to capture subtle differences. SlowFast 8×8 achieved slightly better performance than SlowFast 4×16 and marginally outperformed TSM in Edit score (+0.2) but lagged in $F1_{evt}$ and $F1_{elm}$ metrics.

The 224×224 resolution remains a common choice in video analytics due to its efficiency and compatibility with pre-trained models. Balancing complexity, performance, and efficiency, we selected 224×224 and TSM as the default configuration for F^3ED. We will include the above analysis in the revised manuscript to clarify the trade-offs between resolution, complexity, and performance.

H.4 Choice of CTX module

For the CTX module, we acknowledge that transformer-based models have demonstrated superior efficiency in modeling long-range dependencies. To ensure our choice was justified, we conducted comparative experiments using a Bidirectional GRU (BiGRU) and a Transformer Encoder for the CTX stage. The results are summarized in Table 4(f). As the results indicate, the performance of the two module choices is comparable, with BiGRU slightly outperforming the Transformer Encoder in our F^3ED system. We attribute this to the relatively short event sequences passed to the CTX module, which typically contains fewer than 20 events per 96-frame input clip. Under these conditions, the BiGRU effectively models the necessary temporal context with fewer parameters and lower computational overhead compared to the Transformer Encoder.

I Examples of Error Event Sequences

In this section, we discuss examples of predicted event sequences alongside their corresponding ground truth sequences. These examples illustrate instances of logical errors or uncommon practices observed in predictions generated by TSM + E2E-Spot.

Example 1 (Logical Error):

```
Clip: 20190308-W-Indian_Wells-R64-Serena_Williams-Victoria_Azarenka_173593_173770
Far-end player: Victoria Azarenka (right-handed)
```

```
Near-end player: Serena Williams (right-handed)

Predicted sequence:
near_deuce_serve_-_-_W_-_in ->
far_deuce_return_fh_gs_CC_-_in ->
near_deuce_stroke_fh_gs_DL_-_winner ->
far_ad_stroke_bh_slice_DL_-_forced-err

Ground truth sequence:
near_deuce_serve_-_-_W_-_in ->
far_deuce_return_fh_gs_CC_-_in ->
near_deuce_stroke_fh_gs_DL_-_in ->
far_ad_stroke_bh_slice_DL_-_forced-err
```

In this rally between Victoria Azarenka and Serena Williams, the predicted event sequence contains a clear logical error: the sequence labels a shot as a “winner” but is followed by another shot labeled as “forced-err”, which contradicts the definition of a “winner”.

Example 2 (Logical Error):

```
Clip: 20190308-W-Indian_Wells-R64-Serena_Williams-Victoria_Azarenka_64883_65114
Far-end player: Serena Williams (right-handed)
Near-end player: Victoria Azarenka (right-handed)

Predicted sequence:
near_deuce_serve_-_-_T_-_in ->
far_middle_return_bh_gs_DM_-_in ->
near_middle_stroke_fh_gs_CC_-_in ->
far_deuce_stroke_fh_gs_DM_-_in ->
near_deuce_stroke_fh_gs_DL_-_winner

Ground truth sequence:
near_deuce_serve_-_-_T_-_in ->
far_middle_return_bh_gs_DM_-_in ->
near_middle_stroke_fh_gs_CC_-_in ->
far_deuce_stroke_fh_gs_DM_-_in ->
near_middle_stroke_fh_gs_IO_-_winner
```

The error in this prediction occurs in the final event, where the predicted “near_deuce_stroke_fh_gs_DL...” contradicts the ground truth “near_middle_stroke_fh_gs_IO...”. The prediction does not logically follow from the previous event where the far-end player directed the ball down the middle.

Example 3 (Uncommon Practice):

```
Clip: 20130607-M-Roland_Garros-SF-Novak_Djokovic-Rafael_Nadal_108769_108956
Far-end player: Rafael Nadal (left-handed)
Near-end player: Novak Djokovic (right-handed)

Predicted sequence:
near_deuce_serve_-_-_T_-_in ->
far_middle_return_bh_gs_CC_-_in->
near_ad_stroke_bh_gs_CC_-_winner

Ground truth sequence:
near_deuce_serve_-_-_T_-_in ->
far_middle_return_fh_gs_CC_-_in->
near_ad_stroke_bh_gs_CC_-_winner
```

In this rally between Novak Djokovic and Rafael Nadal, the predicted sequence suggests an uncommon practice: Nadal, a left-handed player, is unlikely to return a deuce court serve to T using his backhand. Typically, a left-hander would use a forehand for such a shot, indicating a likely error in the predicted event.

Table 15: Ablation study on input image resolution.

Dataset	F1_{lcl}	F1_{evt}	Edit
F^3Set (G_{high})	86.7	40.3	74.0
ShuttleSet	97.9	70.7	77.1
FineDiving	94.2	77.6	95.1
FineGym	84.0	70.9	70.7
CCTV-Pipe	71.9	37.0	39.5

J Impact of the Event Localizer to the Whole F^3ED System

To understand the impact of the Event Localizer (LCL) on the performance of the overall F^3ED system, we conducted additional analysis and included an "F1_{lcl}" column in Table 15, which evaluates the precision of the LCL module in identifying event moments with tight temporal tolerance. The table compares the F1_{lcl} metric with the overall system metrics (F1_{evt} and Edit) across various datasets. All use TSM as video encoder and F^3ED as the head architecture.

We observe that the performance of F^3ED is positively correlated with the quality of the LCL module. For example, datasets like FineDiving and ShuttleSet, which have high-performing LCL modules, result in superior downstream performance (F1_{evt} and Edit). Conversely, on datasets like CCTV-Pipe, where the LCL module performs less effectively, F^3ED's overall performance is less ideal.

However, it is important to highlight that even when the LCL module does not perform well, our method still outperforms other state-of-the-art methods (as shown in Table 5 in the paper). Therefore, a very good-performing LCL module is not a hard prerequisite.

K Limitations and Social Impact

In addressing the limitations of our current work, we acknowledge that the expansion of our dataset to include a broader range of videos from both professional and lower-tier matches is an essential yet exceedingly time-consuming and labor-intensive task. The enhancement of our dataset is imperative for providing a more comprehensive analysis that spans various levels of play.

Furthermore, the primary objective of this project is to extend the scope of our tennis analytics from exclusively focusing on elite professional athletes to encompassing a wider audience. This includes semi-professional players, collegiate athletes, junior competitors, and general tennis enthusiasts. By broadening our analytical reach, we aim to democratize access to advanced sports analytics, enabling players at all levels to refine their techniques and strategies.

Socially, the implications of our work are significant. By making sophisticated analytics available to a more diverse group of users, we can contribute positively to public health and fitness. Access to detailed performance data allows individuals to make informed decisions about their training regimes, thus enhancing their overall sports skills and encouraging a healthier lifestyle. This democratization not only fosters a greater appreciation and understanding of tennis but also motivates a broader spectrum of the population to engage actively in sports, thereby promoting physical well-being and health consciousness across communities.