

A PRELIMINARY RESULTS FOR MULTI-MODAL LLMs

We conducted experiments using GPT4 (54), 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:
<ul style="list-style-type: none"> • 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 <u>won the first set 7-6</u> (7-5) and is ahead in the second set with a <u>score of 4-2</u> (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 USTA¹. 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
<ol style="list-style-type: none"> 1. One player has just served the ball. The serve appears to be going towards the <u>T (center)</u> (wide) of the service box on the opponent’s side. 2. The receiving player is getting ready to <u>return the serve with a backhand stroke</u> (forehand), suggesting the serve was directed to his backhand side. 3. The receiver has hit the ball back, likely aiming <u>cross-court</u> (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 <u>forehand</u> (backhand) stroke, aiming the shot either <u>cross-court or down the line</u> (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 <u>backhand return</u> (forehand), which suggests the previous shot was a cross-court <u>forehand</u> (backhand). 10. The player makes contact with a backhand stroke (forehand), likely sending the ball either <u>down the line or cross-court</u> (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 <u>server prepares for a backhand return</u> (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.

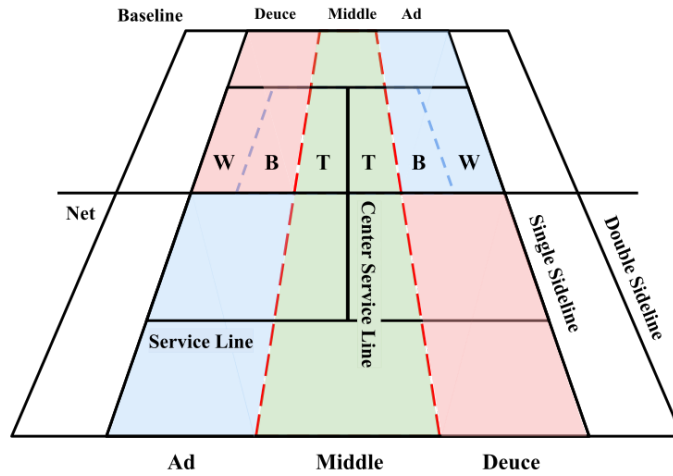


Figure 6: Tennis court and related terminologies.

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 F³ EVENTS IN OTHER DOMAINS

For the badminton dataset, each event consists of 6 *sub-classes*, denoted as sc_1, sc_2, \dots, 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, \dots, 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, \dots, 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) IL, 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 F³Set 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 F³Set 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 F³Set.

Sub-Class	Element	Count	Proportion (%) in Sub-Class
<i>sc</i> ₁	<i>near</i>	21,467	50.1%
	<i>far</i>	21,362	49.9%
<i>sc</i> ₂	<i>deuce</i>	14,474	33.8%
	<i>ad</i>	16,310	38.1%
	<i>middle</i>	12,045	28.1%
<i>sc</i> ₃	<i>forehand</i>	27,802	64.9%
	<i>backhand</i>	15,027	35.1%
<i>sc</i> ₄	<i>serve</i>	11,584	27.0%
	<i>return</i>	8,216	19.2%
	<i>stroke</i>	23,029	53.8%
<i>sc</i> ₅	<i>T</i>	4,428	10.3%
	<i>Body</i>	2,241	5.2%
	<i>Wide</i>	4,915	11.5%
	<i>cross-court</i>	11,933	27.9%
	<i>down the line</i>	3,521	8.2%
	<i>down the middle</i>	11,040	25.8%
	<i>inside-in</i>	608	1.4%
<i>sc</i> ₆	<i>inside-out</i>	4,143	9.7%
	<i>ground stroke</i>	38,287	89.4%
	<i>slice</i>	3,358	7.8%
	<i>volley</i>	497	1.2%
	<i>lob</i>	334	0.8%
	<i>drop</i>	236	0.5%
<i>sc</i> ₇	<i>smash</i>	117	0.3%
	<i>approach</i>	964	2.3%
<i>sc</i> ₈	<i>non-approach</i>	41,865	97.7%
	<i>in-bound</i>	31,245	73.0%
	<i>winner</i>	3,734	8.7%
	<i>forced error</i>	2,808	6.5%
	<i>unforced error</i>	5,042	11.8%

Table 9: Summary of F³Set 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 F³Set by incorporating more matches from platforms like UTR² and junior-level competitions using our annotation toolchain.

²<https://www.utrsports.net/>

Table 10: Summary of F³Set 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 F³Set 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 F³ED STATISTICS

Key statistics for F³Set badminton dataset are summarized in Table 9; for F³Set table tennis dataset are summarized in Table 10; for F³Set tennis doubles dataset are summarized in Table 11.

Furthermore, we present additional statistics related to the F³Set (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 F³SET DATASETS

The F³Set 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 F³Set 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* [64] utilizes a purely 2D CNN architecture. Each frame is processed independently with RGB images as inputs, employing the RegNet-Y 200MF architecture [57] as the backbone.
- *TSM* [35] 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) [35], integrated at strategic points within each residual block, specifically targeting a quarter of the channels, adjusted to the nearest multiple of four.
- *SlowFast* [20] 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 [22] 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 [12]. This architecture allows flexibility for integrating alternative 2D CNN designs. Beginning with pretrained ImageNet-1K weights [12], the encoder is meticulously fine-tuned to our specific dataset needs.

Let J denotes the number of event types.

- *MS-TCN* [19] employs successive layers of dilated convolutions to capture long-range temporal dependencies in sequence modeling tasks. We adapt the code from [19], 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* [71] leverages a transformer-based architecture with segment embeddings to enhance temporal action segmentation. We use code and settings from the implementation by [71].
- *G-TAD* [70] 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 [70], 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* [72] employs transformer networks for efficient single-shot temporal action localization, using multiscale features and local self-attention. We employ the architecture proposed by [72] 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* [24] utilizes a bidirectional GRU layer to facilitate long-term temporal reasoning for precise action spotting. We use a 1-layer bidirectional GRU [13] 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 F³ED

Training. We train all components of F³ED 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 [12] 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 F³ED, 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 F³Set (tennis) with 3 levels of granularity.

Video encoder	Head arch.	F ³ Set (G_{high})			F ³ Set (G_{mid})			F ³ Set (G_{low})		
		F1 _{evt}	F1 _{elm}	Edit	F1 _{evt}	F1 _{elm}	Edit	F1 _{evt}	F1 _{elm}	Edit
TSN [64]	MS-TCN [19]	15.9	59.8	53.5	23.2	60.9	65.8	45.7	70.4	72.8
	ASformer [71]	11.9	54.3	49.8	17.3	56.1	62.5	40.3	67.3	70.3
	G-TAD [70]	6.0	47.5	24.7	14.1	52.1	48.6	19.9	57.4	44.7
	ActionFormer [72]	18.4	60.6	55.2	24.8	61.9	67.3	48.7	70.6	72.2
	E2E-Spot [24]	24.7	65.3	60.1	31.5	66.2	71.0	53.5	73.6	75.0
SlowFast [20]	MS-TCN [19]	17.2	63.1	56.2	24.3	65.5	70.3	47.4	73.1	73.5
	ASformer [71]	14.1	60.8	55.3	20.3	62.8	69.4	44.8	72.9	71.9
	G-TAD [70]	23.0	66.1	64.0	29.6	66.5	74.2	53.3	76.0	77.9
	ActionFormer [72]	28.7	70.0	67.6	35.5	70.9	76.4	59.3	77.1	81.5
	E2E-Spot [24]	25.9	69.4	65.7	33.8	70.4	75.4	55.5	76.5	79.5
I3D [5]	E2E-Spot [24]	22.7	59.7	68.7	27.1	60.7	74.2	51.9	67.7	78.3
VTN [52]	E2E-Spot [24]	14.8	58.3	56.7	20.0	59.4	68.2	39.7	63.1	73.1
TSM [35]	MS-TCN [19]	21.7	67.3	58.6	30.4	69.5	73.0	50.2	74.0	75.3
	ASformer [71]	17.6	61.9	57.5	25.5	64.0	74.2	46.0	72.9	74.0
	G-TAD [70]	16.9	62.5	55.2	29.8	66.9	74.8	39.8	70.1	67.2
	ActionFormer [72]	22.4	65.7	60.3	31.0	68.2	74.7	52.4	73.8	74.9
	E2E-Spot [24]	31.4	71.4	68.7	39.5	72.3	77.9	60.6	78.4	82.1
TSM[35]	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 F³ event detection in F³Set. We used MMPose³ to extract player key points from original 1280x720 resolution images, generating skeleton data. Two advanced skeleton feature extractors ST-GCN++ [17] (GCN-based) and PoseConv3D [18] (CNN-based) were evaluated. The extracted skeleton features were processed with the F³ED 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 F³Set 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 F³Set, 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 F³Set using TSM as the video encoder with input resolutions of 224x224, 336x336, and 448x448. The results show a consistent improvement in performance as resolution increases, though the gains diminish at higher resolutions (e.g., 448x448). This suggests that while

³<https://mmpose.readthedocs.io/en/latest/>

Table 13: Skeleton-based method compared with TSM + F³ED. “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.

Experiment	Params (M)	Inference time (ms)	F ³ Set (G_{high})			F ³ Set (G_{mid})			F ³ Set (G_{low})		
			F1 _{evt}	F1 _{elm}	Edit	F1 _{evt}	F1 _{elm}	Edit	F1 _{evt}	F1 _{elm}	Edit
TSM + F ³ ED	5.6	10.6	40.3	75.2	74.0	48.0	76.5	82.4	68.4	80.0	87.2
ST-GCN++ [17]	2.3	4.0	25.4	62.1	56.1	32.4	63.9	63.5	55.1	69.4	73.2
PoseConv3D [18]	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.

Video encoder	Head arch.	Resolution	Params (M)	FLOPs	F ³ Set (G_{high})		
					F1 _{evt}	F1 _{elm}	Edit
TSM	F ³ ED	224×224	5.6	77.3	40.3	75.2	74.0
TSM	F ³ ED	336×336	5.6	77.3	43.2	77.1	74.8
TSM	F ³ ED	448×448	5.6	77.3	44.4	78.1	74.5
SlowFast 4×16	F ³ ED	336×336	52.7	494.2	37.4	73.7	71.3
SlowFast 8×8	F ³ ED	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 F³ED 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 F³ED. 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 F³ED 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 ³ Set (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 F³ED SYSTEM

To understand the impact of the Event Localizer (LCL) on the performance of the overall F³ED 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 F³ED as the head architecture.

We observe that the performance of F³ED 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, F³ED’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.