# Player-Centric Ball Action Spotting in Long-Form Soccer Broadcast Video: A Comparison of Vision-Language and Scene Graph Approaches

Wisoo Song
Graduate School of Engineering
Seoul National University
osiajod@snu.ac.kr

Jeonghoon Shim
Graduate School of Data Science
Seoul National University
jhshim98@snu.ac.kr

Myunggeum Jee
Graduate School of Data Science
Seoul National University
goldmango@snu.ac.kr

## Abstract

*Automated soccer video analysis holds significant value for the sports analytics industry. However, fine-grained annotation of player-attributed ball contact events across full-length broadcast matches remains a time-consuming manual process requiring domain expertise. We address the CVPR 2026 Player-Centric Ball Action Spotting Challenge, where the goal is to jointly identify 1. who performs each ball action, 2.which action it is, and 3.when it occurs across the entire broadcast videos. We compare two contending approaches. First, we propose* ***Soccer Action QWen****, a Qwen3.5-based vision language model that is augmented with video frame classifier heads and generates autoregressive natural language commentary as compact inter-segment memory. Second, we propose a* ***Video Scene Graph*** *approach that reconstructs spatio-temporal player interactions via a graph neural network, producing structured, video segment representations. Both approaches are evaluated on the FOOTPASS benchmark [18] and the CVPR 2026 challenge leaderboard.*

## 1. Introduction

The soccer industry's demand for fine-grained tactical intelligence is growing, yet extracting structured, player-centric match data from broadcast video remains largely manual [18]. Automating this pipeline would unlock scalable analytics for clubs, broadcasters, and performance platforms.

**Ball Action Spotting.** Ball Action Spotting (BAS) is the task of jointly predicting *who* performed each ball-contact action, *what* the action class is (pass, drive, cross, shot, etc.), and *when* it occurred across a full-length broadcast match [4, 11, 18]. Unlike sparse highlight detection, BAS operates at the dense level of on-ball events. It therefore serves as the foundation for player-level statistics, contribution attribution, and data-driven tactical modeling. The task combines action recognition, multi-object tracking, entity-relation analysis, and game-context understanding—making it one of the most challenging problems at the intersection of computer vision and sports analytics.

**Key Challenges.**
**(1) Long-horizon context.** A full broadcast spans over 100 minutes of video. Even at 1 fps, a 256K-token context window covers only ∼35 minutes [2]; methods must maintain a compact match-level representation without re-processing the entire video.
**(2) Frequent camera view changes.** Broadcast production routinely cuts between wide-angle live play, close-up tracking shots, and slow-motion replays from alternate angles. Methods that treat the broadcast as a continuous single-viewpoint stream will misinterpret these transitions (particularly replays) as new in-game events, degrading both spotting precision and player attribution.

**Our Approaches.** We examine two architecturally distinct methods addressing these challenges. A central requirement for understanding extremely long-form video is an efficient way to summarize and retain context from past segments. VLMs admit a natural solution to this: prior outputs and scene context can simply be fed back as text into the next inference step. Video processing networks, in contrast, retain context by representing on-field objects as

a graph. Because these correspond to two fundamentally different paradigms for compressing and maintaining context over video data, our motivation is to compare which paradigm performs better when applied to soccer broadcast video.

**SoccerAction-QWen (VLLM-based).** We leverage Qwen3.5 [2], pre-trained on multimodal corpora including sports content, making soccer fine-tuning a targeted subdomain specialization. The model autoregressively generates natural language commentary per segment. Feeding this textual summary provides prefix context for the next video segment. This reduces the problem of handling an intractable video context to a rolling textual digest. We extend the vocabulary with special tokens reserved for actions types, team distinction (left side, right side), replay flagging and jersey numbering. Furthermore, we inject event class token and jersey class token for aggregating video information from each video temporal unit. Event class token is responsible for player information summary and replay frame classification. For fine-tuning, we apply LoRA [13] to the language processing branch and train on the FOOTPASS dataset.

**Video Scene Graph Approach.** Motivated by FOOTPASS [18]'s multi-agent paradigm, we attach a Graph Neural Network to a video feature extractor to build a spatio-temporal scene graph per timestep. Nodes represent the players; edges encode proximity, possession likelihood, and interaction patterns from velocities and relative positions. Graph states propagate across segments for long-range temporal reasoning, and updates are suppressed during detected replay intervals.

We evaluate both approaches on the FOOTPASS benchmark and the CVPR 2026 leaderboard, analyzing their complementary strengths, failure modes, and computational trade-offs under real-world broadcast conditions.

## 2. Related Works

### 2.1. Player-Centric Ball Action Spotting

Action spotting in soccer broadcast video has been pioneered by the SoccerNet benchmark family. SoccerNet [10] introduced 500 complete matches annotated with temporal anchors for three coarse event classes, while SoccerNet-v2 [4] expanded coverage to 17 action classes with roughly 300,000 annotations, spurring a rich body of work on temporal pooling and end-to-end spotting architectures [10]. However, both benchmarks annotate events at the match level without identifying the acting player, which is insufficient for fine-grained sports analytics. MultiSports [16] formalized the problem of spatiotemporal action detection with per-actor bounding box annotations across multiple sports, but soccer-specific player-centric spotting remains underexplored. FOOTPASS [18] directly addresses this gap as the first benchmark for play-by-play action spotting in full-length soccer broadcasts, providing rich on-ball event annotations with per-player team identity, jersey number, and tactical role labels.

### 2.2. Spatio-Temporal Scene graphs

Recent studies have explored graph-based representations and spatio-temporal modeling techniques for Spatio-temporal action detection(STAD). Graph-based approaches represent the game context through a spatio-temporal graph constructed at the frame- or clip-level scene. Specifically, a scene graph $G = (V, E)$ is defined where each node corresponds to an individual player, and node embeddings encode various attributes, including player position, team affiliation, motion velocity, and visual features extracted from visual encoders. Edges are established between players based on distance-based thresholds or connecting K-nearest neighbors, enabling the graph to capture local player interactions within the game [5, 8, 12]. The resulting graph is processed using Graph Neural Networks or Transformer-based architectures to learn relational dynamics among players. Through these node interactions, the model can generate graph-level embeddings that effectively represent the contextual state of the scene. Previous studies have demonstrated that relational reasoning over scene graphs is highly effective for modeling both spatial interactions and long-range temporal dependencies in dynamic video understanding tasks [8, 22, 24]. For instance, in a passing scenario, the interactions among the ball carrier, nearby defenders, and the intended receiver provide critical contextual cues for accurate action recognition and temporal localization [5].

### 2.3. Vison Language Models

By synergizing pretrained vision encoders with large language models, vision-language models (VLMs) facilitate complex cross-modal reasoning. Initial breakthroughs in modality alignment were driven by CLIP's [19] contrastive pretraining, followed by BLIP-2 [14], which effectively bridged frozen encoders and LLMs using computationally light query adapters. LLaVA [17] subsequently catalyzed the widespread adoption of visual instruction tuning, a framework later expanded by Qwen-VL [1] through improved architectural backbones and high-resolution processing capabilities. Expanding VLMs to process video inherently requires modeling frame-to-frame temporal dynamics. Video-LLaMA [25] and VideoChat [15] spearheaded video-specific instruction tuning by utilizing visual Q-Formers to encode temporal frame samples. More recently, Qwen3-VL [2] has demonstrated remarkable efficacy in capturing nuanced temporal events by employing dynamic resolution and dense frame sampling, achieving strong performance across video QA and temporal ground-

ing benchmarks.

### 2.4. Video Processing Networks

In many modern action detection frameworks, the video encoders transform sequential video frames into high-dimensional feature tensors that encapsulate motion, appearance, and temporal context. These encoded representations are utilized by downstream detection modules to localize action instances both temporally and spatially within untrimmed videos [7, 20]. Most existing approaches employ 3D Convolutional Neural Networks (3D CNNs) as the primary video encoder due to their ability to jointly model spatial and temporal dynamics across consecutive frames. Among these, Inflated 3D Networks (I3D) [3] extend conventional 2D convolutional kernels into the temporal dimension and are commonly pretrained on large-scale datasets to improve spatio-temporal feature learning. More recent studies, particularly those focused on specialized domains like sports analytics [4, 11], have increasingly adopted lightweight backbones such as X3D [6]. X3D delivers competitive performance while significantly reducing the number of parameters and computational complexity compared to heavier architectures. This efficiency makes X3D particularly suitable for processing fine-grained sports actions and enables end-to-end training under limited computational resources.

### 2.5. Dataset

**FOOTPASS.** FOOTPASS [18] benchmarks player-centric Ball Action Spotting in full-length soccer broadcasts, featuring 102,992 on-ball event annotations across 54 matches (48 train / 3 val / 3 challenge). Annotations specify the exact frame, team, jersey number, and one of 8 action classes (Drive, Pass, Cross, Shot, Header, Throw-in, Tackle, Block). Per-frame tracking data provides structured tactical context, including player jersey numbers, roles, normalized global pitch positions, velocities, and bounding boxes. Evaluation computes F1-score at confidence threshold $\tau$=0.15 with a $\pm$12-frame temporal tolerance.

**Data Processing for Soccer Action QWen.** Our `SoccerEventDataset` segments half matches into non-overlapping, 150-frame clips ($\sim$6 s at 25 fps), sampled at 3–5 fps and resized to $\leq$ 416 $\times$ 768 pixels. A SigLIP 2 [21] processor packs frame pairs into temporal units.

One training sample consists of: (i) a system prompt establishing the analyst role alongside a tactical context string (roles, jersey numbers, and coordinates for a representative frame); (ii) the video frames; and (iii) assistant-side natural language commentary describing events from the past $K$ clips. Tactical context string precedes the video block, enabling injected <|event_cls|> tokens to leverage the full tactical context via causal self-attention.

To prolong context memory, a sliding window prepends commentary output from the preceding $K$ clips as clip history. During training, groundtruth history is prepended, while for inference, the model autoregressively inputs and generates commentary. Cross-match generalization is encouraged by sampling from different matches simultaneously in each batch. For each match progression in the batch, the `GameOrderedSampler` enforces strict chronological ordering within matches during training to ensure that we prepend valid history records for respective clips.

Multi-task supervision applies jointly across all temporal units. By prepending <|event_cls|> tokens to each group, the model predicts four attributes per unit (action class, replay flag, team, jersey number). This jointly optimizes the visual classification heads alongside the autoregressive language generation objective.

## 3. Experiments

We evaluate four methods on the FOOTPASS benchmark [18] for player-centric ball action spotting in full-length soccer broadcast videos.

### 3.1. X3D + TAAD

**Settings** We examine the X3D-based TAAD baseline for player-centric action spotting as proposed by [18]. The model follows a fully visual spatio-temporal action detection pipeline that combines a lightweight 3D CNN backbone with ROI-based player feature extraction and temporal action classification.

As the visual backbone, we adopt a pretrained X3D-S network [6] initialized from Kinetics pretraining. The X3D backbone extracts hierarchical spatio-temporal feature maps from the input video stream. ROIAlign is applied to the feature maps using the provided player tracking annotations. For each player, cropped region features are extracted across the temporal dimension and pooled into fixed-dimensional embeddings. The resulting player feature sequences are further processed using temporal convolution layers to model short-term motion dynamics and temporal context. Finally, a fully connected classification head predicts frame-wise action probabilities over predefined action categories.

**Training Details.** The model is trained end-to-end using the Adam optimizer for 10 epochs. Batch normalization and GELU activation functions are applied throughout the architecture to stabilize training and improve feature representation learning. The X3D backbone remains initialized from pretrained weights while the temporal aggregation and prediction layers are jointly optimized on the FOOTPASS training split.

**Results.** Experimental results are reported in Table 3.

### 3.2. Vanilla VLM

**Settings.** We use Qwen3.5-4B as our base model. A full 90-minute match is split into fixed-length clips, and action spotting is performed independently on each clip to produce predictions over the entire match. We evaluate two inference strategies: (1) Basic: the VLM is given a video clip and directly performs action spotting; (2) Two-Stage: the VLM first generates a natural language description of the clip, which is then used as additional context to perform action spotting.

**Configuration.** Each match half is segmented into non-overlapping 10 frame clips and we vary the sampling rate across the 1 sec, 5 sec, 10 sec clip coverage length. We fix the resolution at 640 × 352.

**Results.** Table 1 shows the results. All configurations yield F-1 score below 3%, indicating severely limited action spotting performance. The Two-Stage mode shows a marginal improvement over the Basic baseline, yet remains far below practical utility. We further investigate whether the clip length constrains the amount of information available to the model by evaluating 3 settings: (1) 5-second, (2) 10-second, (3) 1-second clips. All three settings show low performance, however, the 1-second window yields the highest scores. This can be interpreted as reflecting the nature of soccer, where actions occur in very quick successions.

Table 1. Vanilla VLM F-1 score

| | Basic | | | Two-Stage | | |
|---|---|---|---|---|---|---|
| | 1 sec | 5 sec | 10 sec | 1 sec | 5 sec | 10 sec |
| Qwen3.5-4B | 0.019 | 0.008 | 0.008 | 0.027 | 0.013 | 0.009 |
| Qwen3.5-2B | 0.010 | 0.003 | 0.003 | 0.013 | 0.006 | 0.004 |

### 3.3. SoccerAction-Qwen3.5

We implement and evaluate two model variants: one with the Qwen3.5-4B base and the other with 2B-parameter base (Qwen-3.5-2B), both fine-tuned with LoRA ($r = 24$) on the FOOTPASS training split. Refer to Figure1. for our architectural adaptations.

**4B configuration.** Each match half is segmented into non-overlapping 100-frame clips (∼4 s coverage at 25 fps), downsampled at 3 fps and resized to at most $352 \times 640$ pixels (max_pixels = 220k). The vision encoder is kept frozen throughout fine-tuning. Training runs for 5 epochs with a learning rate of $1 \times 10^{-5}$ and a decay factor of 5, using a batch size of 2 with no gradient accumulation.

**2B configuration.** Each clip spans 150 frames (∼6 s), downsampled at 5 fps and resized to at most $416 \times 768$ pixels (max_pixels = 320k). Unlike the 4B variant, the vision encoder is jointly fine-tuned with the language branch. Training runs for 5 epochs under the same learning rate schedule.

**Shared settings.** Both variants use a context window of $K = 1$ preceding clip as rolling match history, the `GameOrderedSampler` with 4 consecutive clips per game window and 5 concurrent game streams. Off-screen player information is excluded from the tactical context string to reduce input token count. To address class imbalance, background-only clips are skipped with probability 0.8 during sampling.

**Training objective.** The base LM weights are frozen; only LoRA adapters ($r = 24$, targets: `q_proj`, `v_proj`) and the classification heads are updated. The total loss is:

$$\mathcal{L} = \lambda_{\text{lm}}\,\mathcal{L}_{\text{LoRA-lm}} + \lambda_{\text{aux}}\,\mathcal{L}_{\text{aux}} \quad (1)$$

$$\mathcal{L}_{\text{aux}} = \mathcal{L}_{\text{action}} + \mathcal{L}_{\text{replay}} + \mathcal{L}_{\text{team}} + w_{\text{jsy}}\,\mathcal{L}_{\text{jersey}} \quad (2)$$

$\mathcal{L}_{\text{LoRA-lm}}$ is loss the term that flows through the LoRA adapters: the teacher-forced autoregressive cross-entropy on commentary tokens, creating a gradient path from the classification heads back into the LoRA-adapted attention layers. The four classification losses are: a 9-class cross-entropy for action class ($\mathcal{L}_{\text{action}}$), binary cross-entropy for replay flag ($\mathcal{L}_{\text{replay}}$) and team direction ($\mathcal{L}_{\text{team}}$), and a 99-class cross-entropy for jersey number ($\mathcal{L}_{\text{jersey}}$). The weights are $\lambda_{\text{lm}} = 1.0$, $\lambda_{\text{aux}} = 0.1$, and $w_{\text{jsy}} = 3.0$, reflecting the greater difficulty of jersey number recognition in broadcast footage.

**Results.** Table 2 reports overall F-1 scores on the test set for the 2B and 4B configuration. We note that SoccerAction-QWen 2B model improved over the vanilla QWen baseline. Although counterintuitive, the 2B model performed better than the 4B model. We believe this is attributable to the fact that in training the two models in DGX SPARK (120GB VRAM) we could use higher resolution frames and a larger context window of 12 seconds for the 2B model, while for the 4B model we had to downsample the frame rates and resolution more aggresively. In prolonged wide-angle shots, jersey numbers often become illegible and high resolution may be a necessary condition for optical character recognition. Furthermore, the video processing branch(SigLIP2) was left fully trainable for 2B while it was left frozen for the 4B model due to OOM issues.

Table 2. SoccerAction-QWen 2B and 4B F-1 score

| | **4B, 640x352** | **2B, 768x416** |
|---|---|---|
| F-1 | 0.04 | 0.121 |

### 3.4. Video Scene Graph

The overall pipeline consists of four stages: (1) player-centric node feature construction, (2) scene graph construction and graph encoding, (3) dual-head action and actor pre-

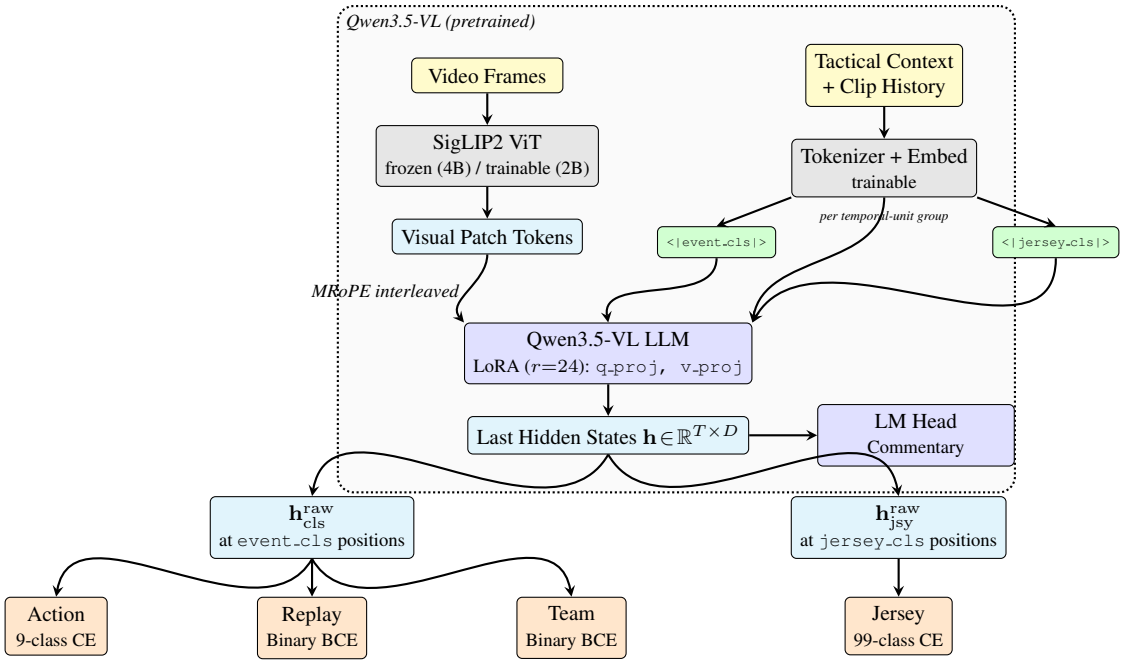


Figure 1. **SoccerAction-QWen architecture.** Video frames are encoded by a SigLIP2 ViT [21] (frozen for 4B, jointly trained for 2B) into visual patch tokens, which are interleaved with the text token sequence inside the LLM via multimodal RoPE (MRoPE). A learnable <|event_cls|> and <|jersey_cls|> token pair is added to the vocabulary and embedded through the language model's *Tokenizer + Embed* layer; one of each special token type is inserted per temporal-unit group. All tokens are processed by the Qwen3.5-VL language model, whose base weights are frozen and adapted via LoRA ($r$=24, targets q_proj and v_proj). After the forward pass, hidden states at the <|event_cls|> positions ($\mathbf{h}^{\text{raw}}_{\text{cls}}$) feed three linear heads predicting action class (9-way CE), replay flag (binary BCE), and team direction (binary BCE). Hidden states at the <|jersey_cls|> positions ($\mathbf{h}^{\text{raw}}_{\text{jsy}}$) feed a jersey-number head (99-way CE). The LM head produces autoregressive natural language commentary, trained jointly with all four heads.

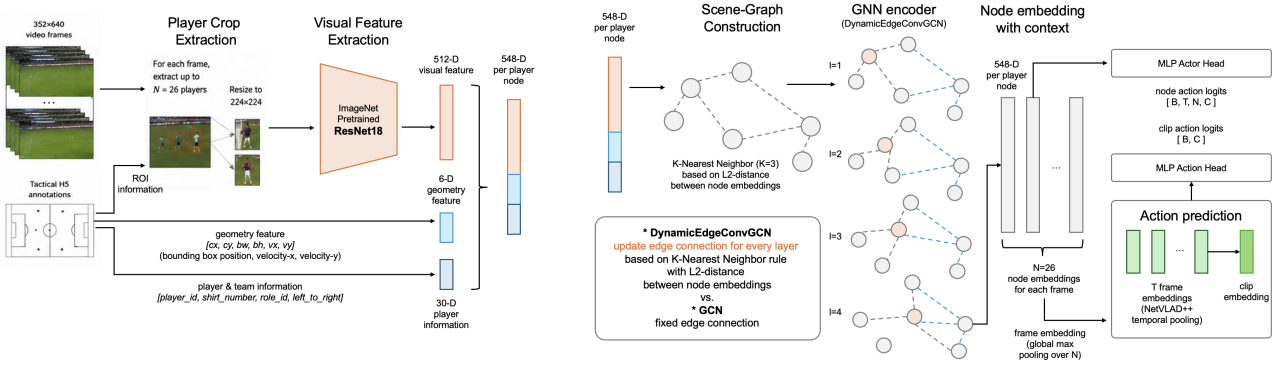


Figure 2. **Video Scene Graph architecture.** Player crops, player information, and geometry features are combined to construct player-centric node representations. DynamicEdgeConvGCN models player interactions through frame-level scene graphs, producing context-aware node embeddings. The resulting representations are used for both actor prediction and clip-level action prediction.

diction, and (4) temporal aggregation for clip-level action understanding.

**Player-Centric Node Feature Construction.** Each player is represented as a graph node whose feature is composed of three complementary components: visual appearance, player identity information, and geometric context. For every frame, player regions are extracted using the ROI coordinates provided in the tactical annotations and resized to 224 × 224. A ResNet18 backbone is used to extract a 512-dimensional visual feature from each player crop. In addition, player-specific metadata including player ID, jersey number, role ID, and attacking direction are converted into learnable embeddings and concatenated into a 30-dimensional player-information vector. Geometric in-

formation is represented by a 6-dimensional vector containing the bounding-box center coordinates, width, height, and velocity components. The final node representation is obtained by concatenating the visual feature, player-information embedding, and geometry feature, resulting in a 548-dimensional player embedding.

Each input clip spans 20 seconds and is sampled at 2 FPS, producing a sequence of $T = 40$ frames. Up to $N = 26$ player nodes are extracted from each frame.

**Scene Graph Construction and Encoding.** For each frame, a player-centric scene graph is constructed where nodes correspond to players and edges represent player interactions. Graph connectivity is established using a dynamic $k$-nearest-neighbor strategy in the node embedding space with $k = 3$. The graph sequence is processed by a 4-layer DynamicEdgeConvGCN encoder [23]. Unlike conventional GCNs with fixed graph structures, DynamicEdgeConv recomputes neighborhood relationships at every layer based on the updated node embeddings, enabling adaptive interaction modeling between players. The graph encoder produces context-aware node embeddings for every player while simultaneously generating frame-level graph embeddings through global max pooling over player nodes.

**Dual-Head Prediction Architecture.** To jointly predict actions and their corresponding actors, two prediction branches are employed.

The first branch is a node-level actor prediction head. The context-aware node embeddings generated by the graph encoder are directly passed to a multi-layer perceptron (MLP), which outputs actor-aware action scores for every player node. This branch produces node-level logits of shape $[B, T, N, C]$, where $C$ denotes the number of action classes.

The second branch is a clip-level action prediction head. Frame-level graph embeddings are aggregated across time using NetVLAD++ temporal pooling [9] to obtain a compact clip representation. A fully connected classifier then predicts clip-level action logits of shape $[B, C]$, indicating which action classes occur within the input clip.

**Inference Strategy.** During inference, the clip-level action head first identifies candidate action classes occurring in the input clip. For each detected action class, the actor prediction head selects the player node with the highest corresponding actor score. The final prediction is represented as a structured event tuple consisting of frame index, player ID, jersey number, action class, and confidence score.

**Training Details.** The model is trained end-to-end using a joint optimization objective consisting of an action classification loss and an actor prediction loss. The action branch is supervised using clip-level action labels, while the actor branch is supervised using actor-aware node-level annotations. The total loss is defined as the weighted sum of the two objectives. The model is trained using the Adam optimizer with a batch size of 64 and an initial learning rate of $1 \times 10^{-3}$. The learning rate is gradually decayed during training, reaching a final learning rate of $1 \times 10^{-8}$ after 10 epochs.

Table 3. F-1 score of Scene-Graph Video Processing Network and the X3D + TAAD baseline on the FOOTPASS validation split.

| | **Scene Graph** | **X3D + TAAD** |
|---|---|---|
| F-1 | 0.189 | 0.410 |

**Results.** Table 3 presents the experimental results of the proposed Scene-Graph framework compared to the TAAD baseline on the FOOTPASS validation split.

### 3.5. VLM Qualitative Analysis

In our previous experiments, we observed that VLMs exhibit notably poor action spotting capabilities in soccer broadcast videos. Through fine-grained qualitative analysis, we identified two recurring failure modes, illustrated in Figure 3.

(1) Repetitive response tendency: As shown in (a) and (b), while player 17 does perform an action in the scene, the ground-truth label is *cross* (Action Code: 3) rather than *drive* (Action Code: 1). Nevertheless, the VLM confidently predicts *drive* with high confidence (0.95) and persistently repeats this erroneous response across subsequent frames.

(2) Hallucination propagation: As shown in (a) and (c), player 26 is the most clearly visible jersey number in the scene. When the VLM fails to clearly identify player 17's jersey number, it incorrectly assumes the number to be 26. We attribute this to error propagation: the preceding misclassification of player 26 as performing a dribble contaminates the model's subsequent reasoning, leading it to confuse player identity based on an already-incorrect action spotting result.

## 4. Conclusion

SoccerAction-QWen's task-specific adaptations, jersey/team/replay auxiliary supervision, LoRA fine-tuning, and tactical context conditioning raised vanilla VLM's F-1 from below 3% to 12.1% (Table 2).

Despite these gains, our Video Scene Graph approach, pairing a lightweight visual feature extractor with a graph neural network over player nodes, outperforms on the FOOTPASS benchmark (Table 3).

We attribute this gap to how each paradigm confronts the two key challenges identified in Section 1.

**Long-horizon context.** The scene graph maintains a match-level representation through an active update on a compact, fixed set of player nodes: features are propagated and refreshed clip-to-clip across the graph. This is a far more memory-efficient mechanism than the rolling text summary that SoccerAction-QWen depends on as its only

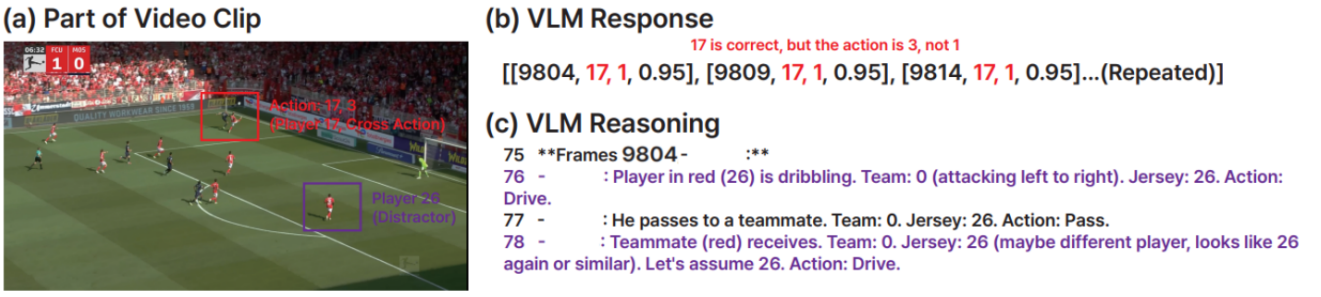


Figure 3. Examples of VLM failure modes in soccer broadcast action spotting. (a) A broadcast scene in which player #17 performs a cross (Action Code: 3). (b) The VLM's action spotting response, which incorrectly predicts *drive* (Action Code: 1) with high confidence (0.95) and repeats this error across frames. (c) The VLM's reasoning, illustrating hallucination propagation: after misidentifying player #26 as dribbling.

cross-clip channel. By contrast, VLM's fixed-length textual prefix increasingly discards detail.

**Frequent camera view changes.** By modeling player dynamics explicitly through inter-node relationships rather than reading identity off raw pixels, the video scene graph is robust to the OCR failures that prolonged wide-angle shots and abrupt camera-view changes induce. Player attribution rests on geometric and motion cues that survive resolution loss and viewpoint transitions. By contrast, SoccerAction-QWen must re-read jersey numbers that are frequently illegible under exactly these conditions, as our qualitative analysis confirms.

**Future Work.** We identify two directions for follow-up work. First, SoccerAction-QWen could be further improved by modifying SigLIP2 to better ingest player velocity information; we currently exclude this portion of the tactical data because SigLIP2 does not support encoding differential quantities, and naively injecting them as text has proven to be ineffective. One possibility is exploiting the fact that two video frames are grouped into a temporal group. We may employ a separate training scheme by setting the optical flow as a target proxy and passing the token that encodes consecutive frame information to regress the velocity. Second, the commentary audio track is a promising additional signal for both SoccerAction-QWen and Video Scenegraph. This is especially promising for the QWen3.5-omni which supports audio understanding.

## References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. 2

[2] Shuai Bai and Yuxuan Cai et al. Qwen3-vl technical report, 2025. 1, 2

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017. 3

[4] Adrien Deliege, Anthony Cioppa, and Silvio et al. Giancola. Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4508–4519, 2021. 1, 2, 3

[5] Haytham Fayek, Mahmoud Ali, and Neil M. Robertson. A graph-based method for soccer action spotting using unsupervised player classification. In *Proceedings of the 2022 International Conference on Multimodal Interaction*, pages 542–546, 2022. 2

[6] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 203–213, 2020. 3

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6202–6211, 2019. 3

[8] Kirill Gavrilyuk, Amir Ghodrati, Zhenzhong Li, and Cees G. M. Snoek. Actor-transformer: A novel architecture for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 349–359, 2020. 2

[9] Silvio Giancola and Bernard Ghanem. Temporally-aware feature pooling for action spotting in soccer broadcasts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4490–4499, 2021. 6

[10] Silvio Giancola and Bernard Ghanem. Temporally-aware feature pooling for action spotting in soccer broadcasts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4490–4499, 2021. 2

[11] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1711–1721, 2018. 1, 3

[12] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings*

*of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019. 2

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 2

[14] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 2

[15] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding, 2024. 2

[16] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. Multisports: A multi-person video dataset of spatio-temporally localized sports actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13536–13545, 2021. 2

[17] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 2

[18] Jeremie Ochin, Raphael Chekroun, Bogdan Stanciulescu, and Sotiris Manitsaris. Footpass: A multi-modal multi-agent tactical context dataset for play-by-play action spotting in soccer broadcast videos. *Computer Vision and Image Understanding*, 269:104790, 2026. 1, 2, 3

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2

[20] Gurkirt Singh, Suman Sharma, Suman Saha, Jeroen Cuyper, Michael Listztenberger, and Luc Van Gool. Spatio-temporal action detection in the wild: A large-scale dataset and baseline. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3110–3120, 2022. 3

[21] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 3, 5

[22] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 399–417, 2018. 2

[23] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):1–12, 2019. 6

[24] Jianchao Wu, Hao Wang, Yufei Wang, Yu Qiao, and Jian Tang. Group activity recognition with multi-granularity graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9610–9619, 2021. 2

[25] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding, 2023. 2