

TOWARDS AUTONOMOUS SPORTS PRODUCTION: A MULTIMODAL PIPELINE FOR MULTI-ANGLE BASKETBALL HIGHLIGHT GENERATION

YUHE BIAN, CHUN ZHOU

Master's thesis
2026:E31



LUND UNIVERSITY

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Abstract

In multi-view sports broadcasting, extracting high-quality highlights and selecting the optimal camera angle from concurrent video streams is traditionally a labor-intensive process with significant computational overhead.

This thesis explores the feasibility of automating the production of multi-view sports highlights using Multimodal Large Language Models (MLLMs). To manage the high computational demands of multi-camera environments, we investigate a hierarchical data-reduction strategy that progressively filters temporal and spatial redundancies.

The research focuses on evaluating the effectiveness of vision-language alignment for precise action localization, comparing traditional video understanding architectures against emergent foundation models. Furthermore, we assess the capability of MLLMs to perform high-level decision-making in optimal view selection. By conducting a performance analysis on real-world multi-view datasets, this study validates the potential for multimodal architectures to achieve human-like directing logic with significantly reduced manual intervention and computational cost.

Keywords: Temporal Sentence Grounding, Video Foundation Models, Vision-Language Models, Cross-Modal Retrieval, Multi-View Video Understanding

Acknowledgments

We would like to express our sincere gratitude to our company supervisor at Sony, Jimmy Engström. His constant guidance, technical expertise, and encouraging feedback made this project a highly rewarding experience.

We sincerely thank Sony and IK EOS Lund, especially the sports director Torbjörn Lyrström, for their invaluable support in collecting and compiling the EOS dataset, which made the evaluation of this multi-view system possible.

We also wish to thank the Vision & AI System Team at Sony, as well as our colleagues Roberto Franceschi & Olivier Moliner. We deeply appreciate their warm welcome, their active engagement with our project, and the valuable time they generously offered during our discussions.

Finally, we extend our thanks to our academic supervisor, Viktor Larsson and Gustav Hanning at Centre for Mathematical Sciences at Lund University. Their academic oversight and insightful feedback were instrumental in shaping this report and keeping our research focused and on track.

Contents

1. Introduction	9
1.1 Sony	9
1.2 Problem Statement	9
1.3 Objectives	9
1.4 Research Questions	10
1.5 Division of Work	11
2. Background and Related Work	12
2.1 Background on Sports Pipeline Construction	12
2.2 Real-Time Object Detection and Spatial Localization	15
2.3 From Action Recognition to Temporal Action Grounding	16
2.4 Multi-Angle Sports and Best-Camera Selection	20
3. Dataset	22
3.1 APIDIS Dataset	22
3.2 IK EOS Dataset	24
4. Methodology & System Architecture	27
4.1 Data Acquisition Module	27
4.2 Trigger Time Extraction	27
4.3 Active Half-Court Detection	28
4.4 Action Recognition with TSM	28
4.5 Internvideo2	29
4.6 Intelligent Best-Cam Selection via VLMs	35
4.7 Design Methodology for the Pipeline Visualization Interface	36
5. Implementation	38
5.1 System Overview	38
5.2 Dataset Processing	39
5.3 TSM(MMAction2)-based Grounding	40
5.4 InternVideo2-based Grounding	42
5.5 Best Camera Selection	44
5.6 Implementation of the Visualization System	44

6. Evaluation and Results	48
6.1 Results on TSM(MMAction2) Model with APIDIS Dataset . . .	48
6.2 Results on TSM(MMAction2) model with IK EOS Dataset . . .	50
6.3 Results on Internvideo2 Model with the APIDIS Dataset	51
6.4 Results on Internvideo2 Model with IK EOS Dataset	53
6.5 Failure Cases Analysis	56
6.6 Best-cam Evaluation	57
7. Discussion	61
7.1 Thematic Synthesis of Research Findings	61
7.2 Comparative Analysis: Discriminative vs. Generative Paradigms	63
7.3 Best-Cam Selection and Pragmatic Edge-Computing Viability . .	64
7.4 Systemic Limitations and Threats to Validity	65
7.5 Future Research Directions	65
8. Conclusion	67
9. Use of AI Tools	68
Bibliography	69

1

Introduction

1.1 Sony

Sony [1] is a creative entertainment company with a solid foundation in technology. In Lund Sony mainly work with Research, specializing in 5G, 6G & IoT; Software Development for Sony products such as Xperia smartphones and Bravia TVs; New Business, e.g., the Sony Startup Acceleration Platform Europe; and Design.

1.2 Problem Statement

Within contemporary sports media production, the process of extracting basketball shooting highlights and selecting the optimal viewing angle from synchronized eight-camera configurations depends predominantly on manual directing. To mitigate the inefficiencies associated with this conventional approach, this paper introduces a fully automated, end-to-end pipeline designed entirely from inception. However, the simultaneous processing of multiple video streams presents a formidable technical challenge, particularly regarding the substantial computational overhead that subsequently induces severe system latency. Consequently, the primary objective of this project is to construct a highly functional prototype with a profound engineering emphasis on regulating computational expenditures, thereby minimizing processing durations and ensuring real-time responsiveness.

1.3 Objectives

The main objective of this thesis is to design, implement, and evaluate an automated, end-to-end video processing pipeline capable of extracting basketball shooting highlights and selecting the optimal camera angle from 8 synchronized video streams, with a focus on minimizing computational latency.

The specific objectives are the following:

- **Pipeline Architecture:** Design and build a functional 0-to-1 pipeline capable of ingesting and syncing 8 simultaneous camera feeds.
- **Event Detection and Best Camera Selection :** Implement algorithms/logic to accurately identify shooting events and determine the most suitable viewing angle for the highlight clip.
- **Performance Optimization :** Identify computational bottlenecks within the multi-view processing logic and apply optimization strategies to reduce execution time and system latency.
- **Evaluation :** Evaluate the finalized pipeline based on two criteria: the accuracy of the highlight extraction/angle selection, and the processing speed compared to a defined baseline.

To provide a holistic perspective of the proposed autonomous sports production system, an overview of our end-to-end multi-modal pipeline is illustrated in Figure 1.1. As a precursor to the detailed methodologies discussed in subsequent chapters, the pipeline conceptually streamlines the flow of multi-view broadcast data through five core stages: heuristic scoreboard trigger extraction, spatial-temporal action grounding, synchronized multi-angle retrieval, VLM-driven camera selection, and automated post-production. With this overview as a shared frame of reference, the subsequent chapters explain each module in detail, covering its mathematical formulation, architecture, and experimental validation.

1.4 Research Questions

To systematically evaluate the feasibility, efficiency, and alignment of the proposed autonomous sports production pipeline, this study formulates and addresses three overarching research questions:

- **Pipeline Feasibility and System Integration**
How feasible is the construction of an end-to-end, automated production pipeline driven by multimodal frameworks for basketball highlight extraction and multi-angle selection?
- **System Architectural Role and Computational Efficiency**
How can a hierarchical data-reduction strategy be designed to optimize the trade-off between computational efficiency and the semantic reasoning capabilities of Vision-Language Models for automated “best-camera” selection?
- **Subjective Quality and Human Alignment**
To what degree do the zero-shot view selection decisions generated by Vision-Language Models align with human subjective aesthetic preferences?

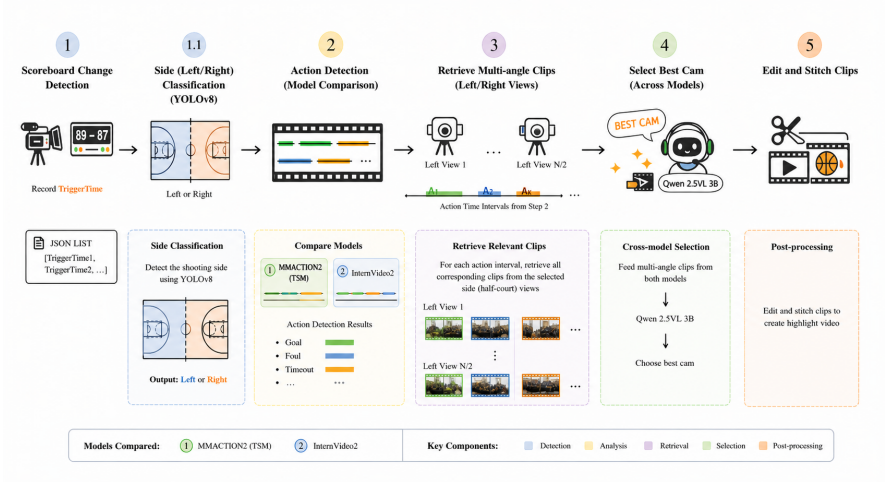


Figure 1.1 Overview of the proposed pipeline for automated basketball highlight generation. The system consists of five sequential stages: (1) scoreboard change detection with active half-court detection, (2) action detection via model comparison, (3) multi-angle clip retrieval, (4) cross-model best-camera selection using a vision-language model, and (5) final clip editing.

1.5 Division of Work

This thesis was carried out jointly by the two authors. For the action detection experiments, the TSM (MMACTION2) part was conducted by Chun Zhou, and the InternVideo2 part by Yuhe Bian. All other parts of the work were completed jointly by both authors.

2

Background and Related Work

2.1 Background on Sports Pipeline Construction

2.1.1 Industry Standards and Production Bottlenecks

In the current sports broadcasting landscape, the generation of highlight reels remains predominantly a manual endeavor[2, 3]. Professional production workflows typically involve a "logger-director" model, where specialized operators manually tag key events (e.g., goals, fouls, or significant shots) in real-time or during post-match review. For multi-camera environments, a director must concurrently monitor several feeds and make split-second decisions on which angle to broadcast or include in a summary[4]. This traditional approach faces two critical bottlenecks: first, the high operational cost and human error associated with manual logging; and second, the "production lag" between the occurrence of an event and the availability of edited content[5]. While some commercial systems utilize ball-tracking or basic rule-based triggers, a cohesive, autonomous pipeline capable of cinematic multi-angle directing and accurately grounding key actions within long untrimmed videos in near real time is still largely absent from the market[6, 7].

2.1.2 Automation and Latency: The E-sports Benchmark

In contemporary sports media, the speed of content delivery is as vital as its quality. A significant gap exists between traditional physical sports and competitive gaming (e-sports) regarding production latency. In e-sports, automated highlight systems can distribute key moments to live streams within seconds of their occurrence—often within the timeframe of a standard broadcast delay. This extreme efficiency is facilitated by direct access to game-engine telemetry, which provides instantaneous, structured game-state data. In contrast, physical sports provide no such structured signal: the only machine-readable record of the game is the raw video itself. In contrast, physical sports provide no such structured signal, which implies that the sole machine-readable record of the game is the raw video footage

itself. Key game developments, including made shots, fouls, and rebounds, exist only implicitly across sequential video frames. These moments must then be identified by computer-vision models through a deep semantic understanding of the visual content over time. This interpretation step is inherently slower and less certain than reading a telemetry stream, as its reliability is bounded by detection accuracy, camera viewpoint, and occlusion. This dependence on visual inference constitutes the principal source of production latency that distinguishes physical-sports broadcasting from e-sports. Translating the "sub-minute" production speed of e-sports to physical basketball requires a fully automated computer vision pipeline that bypasses manual logging while maintaining cross-camera temporal synchronization.

Recent advancements in computer vision, particularly the rigorous sub-second evaluation metrics established in the SoccerNet challenges [8], highlight the industry's shift towards addressing this production lag. Furthermore, novel self-supervised approaches for cross-view synchronization [9, 10] are paving the way for the fully automated, multi-camera pipelines required to bridge the efficiency gap between physical sports and e-sports[11].

2.1.3 From Discriminative Recognition to Semantic Grounding

The technical foundation of sports video analysis is evolving from simple classification to complex temporal grounding[12], where the model must identify not only what action occurs but also when it happens within the video timeline. Traditional discriminative frameworks, such as the Temporal Shift Module (TSM) [13], have established a robust baseline by shifting feature channels along the temporal dimension. These models focus on extracting spatio-temporal features to classify predefined actions into discrete categories.

While efficient, these task-specific models have two limitations relevant to highlight production. First, because they are optimized to assign the correct category label to a clip, they are rewarded for recognizing *what* action occurs rather than for delineating *when* it begins and ends; as a result, their predicted temporal boundaries tend to be coarse. Second, their output is confined to a fixed label set, so they cannot reason about open-ended, higher-level properties of a clip, such as which moment is the most salient or how compelling a given viewpoint is for a highlight. We refer to these judgments as *narrative quality*, and they require semantic reasoning that lies beyond closed-set classification.

Current research is transitioning toward Video Foundation Models (VFM), exemplified by InternVideo2 [14] and Video-LLaMA[15], which are pre-trained on massive-scale multimodal datasets. These models allow for "zero-shot" semantic reasoning through vision-language alignment, where visual and textual representations are mapped into a shared semantic space to enable cross-modal understanding without task-specific supervision. This thesis utilizes both paradigms: employing traditional discriminative models for their computational efficiency and precise boundary regression, while leveraging foundation models and Multimodal Large

Language Models (MLLMs) to perform high-level aesthetic and narrative evaluations for camera selection.

2.1.4 Computational Constraints and Cost-Efficiency

While the integration of foundation models and MLLMs represents a significant leap in semantic capability, it introduces substantial challenges regarding hardware expenditure and operational costs. Fully autonomous systems that rely exclusively on heavy MLLMs for real-time analysis of multiple concurrent high-resolution streams often require computationally expensive GPU clusters and incur massive inference latencies[16, 17]. Furthermore, the human resources required to maintain and fine-tune such large-scale architectures add a layer of financial burden that is often inaccessible for local or mid-tier sports productions[18]. A primary research objective of this thesis is to explore the possibility of a "low-cost" automated editing system[19, 20]. By implementing a hierarchical data-reduction strategy—where lightweight discriminative models handle high-throughput event localization and expensive multimodal models are invoked only for high-level decision-making—this study aims to achieve high-quality highlight generation while maintaining computational affordability .

2.1.5 Automated Directing and Subjective Evaluation

The task of "best-camera" selection mirrors the role of a human director in a broadcast truck. This study refers to the methodology in "Deep Directing: Deep Learning-Based Automated Video Directing for Live Sports Events" [6, 21, 11], which emphasizes that automated directing must align with human visual preferences. A core challenge is that directing quality is inherently subjective rather than binary[22]. Consequently, following the human-feedback paradigm used in subjective production research [23], this study evaluates the view-selection module against viewer preference. For each event, participants are shown the synchronized clips from all candidate camera angles and asked to indicate the most suitable view; the majority choice across participants defines the human-preferred view, and the module is scored by how often its top-ranked camera matches this consensus. Participants score synchronized multi-angle clips based on clarity and appeal, establishing a "gold-standard" baseline for the automated system to emulate.

2.1.6 The Landscape of Multi-View Sports Datasets

A major bottleneck in autonomous sports production is the acute scarcity of annotated multi-view datasets[24]. While projects like SoccerNet [2, 4, 25], have provided extensive footage for football, the data is primarily single-view or lacks fine-grained synchronization between cameras. For basketball, existing public datasets such as APIDIS are often outdated, characterized by low resolution and significant frame-synchronization errors. The near-total absence of publicly available basketball datasets providing both synchronized concurrent angles and sub-second event

annotations necessitates the construction of specialized internal datasets, such as the IK EOS dataset from Sony used in this research.

2.2 Real-Time Object Detection and Spatial Localization

Object detection serves as a critical prerequisite in automated sports video analysis, enabling task-specific applications such as player tracking, ball trajectory estimation, and spatial layout inference. Within this domain, the You Only Look Once (YOLO) family of single-stage detectors has established a standard for balancing inference velocity and localization accuracy. In particular, the YOLOv8[26] architecture introduced by Ultralytics incorporates key structural innovations over its predecessors to achieve state-of-the-art performance in highly dynamic environments.

Architecturally, YOLOv8 is composed of a modified CSPDarknet backbone, a Path Aggregation Network (PANet) neck, and an anchor-free decoupled detection head. The backbone replaces the traditional C3 modules with Cross-Stage Partial Bottlenecks with two convolutions (C2f). This C2f design enhances gradient flow and rich feature representation by stacking parallel bottleneck flows, thereby allowing the network to capture complex, multi-scale visual features with a reduced parameter count. The multi-scale features are subsequently fused in the PANet neck, which establishes bidirectional feature propagation paths to preserve precise spatial and semantic information.

Crucially, YOLOv8 transitions from an anchor-based to an anchor-free detection paradigm. Rather than relying on predefined bounding box anchors, the network directly predicts object centers and bounding box offsets from feature maps. This is achieved via a decoupled head structure, where classification and regression tasks are separated into parallel convolutional branches. This decoupling prevents task interference, accelerating convergence during training and improving regression accuracy. Furthermore, the regression task is supervised utilizing a combination of Complete Intersection over Union (CIoU) loss [27] and Distribution Focal Loss (DFL)[28]. While CIoU optimizes the geometric aspects of bounding box regression (including overlap area, aspect ratio, and center distance), DFL addresses boundary uncertainty by representing box locations as general probability distributions. This fine-grained boundary estimation is highly effective on dynamic sports courts characterized by frequent player occlusions and varying camera perspectives.

In the context of multi-view sports production, such lightweight and fast spatial detectors are instrumental for establishing low-latency heuristics. By localizing player distributions across the playing field, spatial regions of interest can be dynamically determined, enabling early-stage data reduction prior to the invocation of computationally expensive multimodal reasoning models.

2.3 From Action Recognition to Temporal Action Grounding

This section provides an overview of the key technical background and recent developments relevant to this research. As the objective of this study is to leverage multimodal foundation models for temporal action grounding in basketball videos, the discussion focuses on the evolution of video action understanding, temporal event localization, and multimodal video-language modeling. These research areas form the foundation of the proposed approach and provide the necessary context for understanding the methodology and design choices presented in this thesis.

2.3.1 Video Action Recognition

Video Action Recognition is a fundamental task in video understanding. Its primary goal is to identify the category of human actions occurring within a given video clip. For instance, in classic benchmark tasks like those based on the Kinetics dataset [29], models are tasked with assigning a single label—such as "playing guitar" or "running"—to a short video clip.

Early approaches to video action recognition typically followed a two-stage pipeline consisting of feature representation and classifier learning. In the feature representation stage, a variety of hand-crafted descriptors were designed to capture spatial and temporal information in videos. For example, space-time interest points (STIP) [30] extend 2D interest point detection into the temporal domain to capture motion patterns, while dense trajectory-based methods, particularly Improved Dense Trajectories (iDT) [31], explicitly track densely sampled points to model long-term motion dynamics. In addition, motion-aware descriptors such as HOG/HOF and their variants were widely used to encode appearance and motion information [32]. Given these feature representations, classification was typically performed using traditional supervised learning frameworks such as Support Vector Machines (SVMs) [31] or Hidden Markov Models (HMMs) [33]. Some works further explored unsupervised or weakly-supervised exploration techniques to discover structure in feature space [34].

Although these methods were effective on early benchmark datasets, they generally suffer from limited representational capacity, poor robustness to complex motion patterns, and weak generalization ability across diverse video scenarios [35]. Their core focus is solving the "What" problem, identifying the action category within a well-defined, trimmed clip. However, in the real world, videos are typically untrimmed. They contain extensive background noise, irrelevant frames, and multiple continuous actions. Traditional action recognition models lack the ability to answer "When", meaning they cannot precisely localize the start and end times of specific actions in long, complex video streams [36]. This lack of sensitivity to temporal boundaries has prompted researchers to shift their focus to the more chal-

lenging task of Temporal Action Detection.

2.3.2 Temporal Action Detection

To address the temporal insensitivity of traditional action recognition, Temporal Action Detection (or Temporal Action Grounding) was introduced. The primary objective of action recognition is to classify actions within a video without the need to detect their duration. In contrast, Temporal Action Detection requires precisely localizing the start and end time points of action instances within complex and long videos alongside identifying their respective categories. Furthermore, the two tasks differ significantly in their processing objects. Action recognition generally utilizes artificially trimmed videos that typically contain only a single action. However, real-world videos are mostly untrimmed and often contain multiple action instances of varying categories with the boundaries and duration of each action being completely unknown. The technical evolution of solving this complex localization problem can be broadly divided into two main stages: CNN-based methods and Transformer-based frameworks.

With the widespread application of deep learning, the design of detection models diversified and predominantly split into two mainstream directions. The first direction involves anchor-based methods, which evolved directly from the sliding window concept. Inspired by region-based Convolutional Neural Networks in object detection, models like SCNN [37] pioneered this direction by applying multi-stage networks to predefined multi-scale anchor boxes for proposal generation, classification, and precise boundary prediction. Subsequently, TURN [38] incorporated unit-level coordinate regression to enhance efficiency, while R-C3D [39] integrated three-dimensional convolutional networks to directly extract continuous video features for precise detection. The second direction comprises boundary-based or bottom-up methods, which were proposed to address the imprecise boundaries inherent in rigid anchor boxes. Models such as BSN [40], BMN [41], and TCA-Net [42] adopted a strategy of predicting the probability of each frame acting as a starting or ending point. High-probability points are then paired to generate highly flexible and precise action boundaries.

More recently, transformer architectures have significantly advanced video understanding and temporal action detection. On the representation side, self-attention mechanisms enable Transformers [43] to capture long-range spatio-temporal dependencies more effectively than conventional convolution-based architectures. Models such as TimeSformer [44] and Video Swin Transformer [45] demonstrate strong capability in extracting discriminative video features through patch-based attention and hierarchical feature modeling.

Beyond feature extraction, the Transformer encoder-decoder framework further

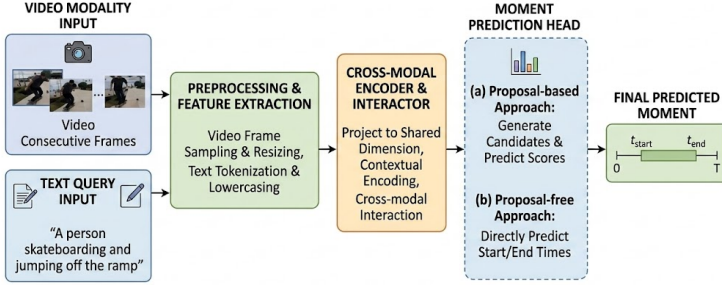


Figure 2.1 TSGV Pipeline

inspired a new generation of query-based temporal action detectors. Inspired by DETR [46]-style object detection, these methods formulate action instances as a set of learnable query embeddings that interact directly with the global video sequence through attention mechanisms. Compared with traditional anchor-based pipelines, query-based approaches largely alleviate the reliance on hand-crafted anchors and heuristic boundary matching strategies, leading to more unified end-to-end frameworks for temporal detection. Within this paradigm, various models introduce different improvements. TadTR [47] adopts temporal deformable attention to dynamically focus on informative temporal regions, while ReAct [48] enhances action classification through improved query interaction mechanisms. Collectively, these Transformer-based query methods have substantially improved temporal localization performance and enabled more unified end-to-end detection frameworks for untrimmed videos.

2.3.3 Temporal Sentence Grounding in Videos

Temporal Sentence Grounding in Videos (TSGV), also known as natural language video localization, aims to identify a temporal segment within an untrimmed video that semantically corresponds to a given natural language query [49]. Compared with temporal action detection, which focuses on recognizing and localizing actions from a predefined set of categories, TSGV extends the task to open-vocabulary language descriptions. This removes the restriction of fixed label spaces and enables more flexible querying of complex and fine-grained human activities in videos. As a result, TSGV requires not only accurate temporal localization but also fine-grained semantic alignment between visual content and natural language [50]. This shift moves the problem from closed-set action recognition towards open-vocabulary cross-modal understanding.

To address this cross-modal grounding problem, existing TSGV methods generally follow a unified framework, as illustrated in Figure 2.1 [50]. Given a video-query pair, where the video is represented as a sequence of frames and the query is a natural language sentence, the model first performs preprocessing, including frame sampling for the video and tokenization for the text query. The processed visual and textual features are then encoded and projected into a shared semantic space, where cross-modal interactions are modeled to capture fine-grained correspondences between video content and language expressions. Based on these fused representations, a temporal localization module is used to predict the start and end boundaries of the target moment. Depending on the design of the temporal prediction strategy, existing approaches can be broadly categorized into two main paradigms: proposal-based and proposal-free methods. Proposal-based methods first generate a set of candidate temporal segments (proposals) from the video and then rank or refine them according to their relevance to the language query. In contrast, proposal-free methods directly regress the start and end timestamps without relying on explicit proposal generation.

Despite the progress made by these traditional TSGV architectures, they often encounter a significant bottleneck regarding feature representation. Most methods rely on off-the-shelf feature extractors trained primarily on closed-set action recognition datasets. These pre-extracted, isolated visual features lack rich semantic language context and struggle to capture the complex, fine-grained spatio-temporal dynamics required for open-vocabulary grounding. Consequently, the performance of the downstream temporal localization module is fundamentally constrained by the semantic gap between these rigid visual features and the highly flexible natural language queries.

To overcome this limitation and achieve deeper cross-modal alignment from the ground up, recent research has shifted towards leveraging large-scale Multimodal Video Foundation Models. By pre-training on massive datasets of video-text pairs, these foundation models learn unified and highly generalized representations that inherently bridge the gap between vision and language. Among them, InternVideo [14] has emerged as a highly advanced architecture. By seamlessly integrating masked video modeling and cross-modal contrastive learning, InternVideo demonstrates exceptional capability in robust video-language understanding and fine-grained spatio-temporal reasoning. This makes it an ideal foundational architecture for tackling challenging TSGV tasks, particularly in highly dynamic and visually complex scenarios such as basketball game analysis, where precise semantic alignment of fast-paced multi-person interactions is crucial.

2.3.4 Multimodal Video Foundation Models

Traditional TSGV methods typically rely on task-specific architectures and pre-extracted visual features, which often suffer from limited semantic understanding

and poor generalization across diverse video-language scenarios. Recent advances in large-scale multimodal pretraining and video foundation models have alleviated these limitations by learning unified video-text representations with stronger temporal reasoning and cross-modal alignment capabilities [51].

Following the success of large-scale multimodal pretraining, several video foundation models have been proposed for video-language understanding tasks. These models leverage Transformer-based architectures and large-scale video-text datasets to learn unified multimodal representations with strong temporal modeling capability. Representative approaches such as VideoMAE [52], InternVideo [53], and InternVideo2 [14] have demonstrated strong performance across various video understanding tasks, including action recognition, video retrieval, and temporal grounding.

In this work, we adopt the end-to-end temporal grounding pipeline provided by the official InternVideo2 framework. Specifically, InternVideo2 [14] is used as a frozen multimodal feature extractor to obtain semantically rich video and text representations. Video clips are sampled at regular intervals and encoded into high-dimensional feature sequences, while text queries are projected into the same embedding space to facilitate cross-modal alignment.

To enable precise temporal localization on top of these foundation representations, the pipeline incorporates CG-DETR [54] (Correlation-Guided DETR) as the grounding module. Different from traditional DETR-based methods that suffer from irrelevant visual noise, CG-DETR explicitly models the contextual relationships between the text query and video background. It leverages a context-guided scrubbing mechanism to suppress text-irrelevant video frames while highlighting moments highly salient to the target action. Building upon the DETR-style [46] formulation, CG-DETR represents action segments as a set of learnable query embeddings, which interact with video and text features through dedicated attention mechanisms. This design decouples multimodal representation learning from temporal boundary regression, allowing each component to focus on its respective role.

By leveraging this framework, our system combines the strong contextual representations of InternVideo2 with the precise localization capability of CG-DETR, providing a reliable foundation for analyzing fast-paced and complex basketball game videos.

2.4 Multi-Angle Sports and Best-Camera Selection

2.4.1 Evolution of Multi-Angle Action Grounding

The research trajectory of action recognition has transitioned from simple classification on single-view datasets to precise temporal grounding in multi-angle footage[8]. Traditional discriminative frameworks, such as the Temporal Shift Module (TSM) [13] (implemented in MMAAction2), provide a robust baseline for event detection but are inherently constrained by fixed perspectives. In a single broadcast

feed, critical shooting motions are frequently obscured by occlusions[55]. Multi-view systems overcome these physical limitations by aggregating complementary information across synchronized camera arrays[56]. Current research in this domain focuses on maintaining rigorous cross-camera synchronization—where minor temporal drifts can disrupt narrative continuity—and merging asynchronous probability streams into a unified temporal anchor for highlight generation[57].

2.4.2 Semantic Reasoning and Computational Affordability

Selecting the optimal camera angle, or the "best-camera," requires a paradigm shift from recognizing patterns to reasoning about visual narratives. Video understanding is evolving toward general-purpose semantic reasoning through emergent Video Foundation Models (VFM), such as InternVideo2 [14] and Multimodal Large Language Models (MLLMs). Unlike discriminative models that only identify action labels, MLLMs can act as "synthetic directors," evaluating the cinematic quality and spatial clarity of different viewpoints through zero-shot reasoning. However, the application of heavy MLLMs across multiple concurrent high-resolution streams presents a prohibitive cost barrier due to extreme GPU memory demands[58]. Consequently, this research investigates a "low-cost" hierarchical strategy, using lightweight spatial filters to prune irrelevant views before invoking expensive multimodal reasoning, thereby balancing directing intelligence with computational affordability[59].

3

Dataset

The development and evaluation of the pipeline rely on two basketball datasets that serve complementary roles: the publicly available APIDIS dataset [60], used in the early stage to establish the feasibility of the system, and an internal multi-camera dataset (IK EOS dataset) provided by Sony, used in the later stage for the main experimental evaluation. The two datasets differ substantially in age, camera coverage, and annotation quality, and each required dedicated preprocessing steps before integration into the pipeline.

3.1 APIDIS Dataset

The APIDIS dataset is a publicly available multi-camera basketball recording that can be used in early multi-view sports analysis research. It comprises footage from seven calibrated cameras placed around a single basketball court, as illustrated in 3.1. Despite its suitability for multi-view basketball analysis, the dataset exhibits two notable limitations:

- **Limited and uneven camera coverage.** Only seven viewpoints are available, and more than 50% of the recorded segments do not contain simultaneous footage from all seven cameras. Additionally, several segments suffer from frame loss and prolonged black frames, where one or more views are missing or remain unrecorded for extended periods.
- **Outdated storage format.** The original release stores each camera view as a sequence of individual image frames rather than as a continuous video stream. This format is not only incompatible with the video-based input expected by modern action detection and vision-language models but also leads to substantial, uncompressed data volumes, rendering it computationally inefficient for large-scale processing.



Figure 3.1 Overview of the APIDIS dataset multi-view framework. The dataset captures a full professional basketball game with a total duration of approximately 2 hours. The synchronized multi-camera setup consists of 7 calibrated viewpoints (including 5 lateral perspective color cameras and 2 overhead fisheye cameras) recording concurrently at an average frame rate of 22 fps and a native resolution of 1600×1200 pixels. The panel illustrates synchronized frames across all active cameras at a single temporal anchor.

Data Preprocessing and Synchronization To address these limitations, a preprocessing procedure was applied to convert the per-camera image sequences into MP4 video files at the original frame rate. Segments containing black or missing frames were either discarded or padded depending on their position relative to the trigger times (see Section 4.2). To address these limitations, a preprocessing procedure was applied to convert the per-camera image sequences into MP4 video files at the original frame rate. To maintain strict time synchronization across multiple views, we implemented a tiered recovery strategy for missing frames:

- **Short Gaps ($\leq 0.2s$):** Linear interpolation was employed to fill these gaps, effectively smoothing visual transitions and preserving optical flow consistency.
- **Long Gaps ($> 0.2s$):** Black frame padding was applied to keep the global timeline intact and ensure temporal alignment across all camera streams.

Segments that remained irrecoverable relative to the trigger times were discarded.

Given the partial multi-view coverage, APIDIS was treated as an early-stage development dataset: it was used to validate the feasibility of each pipeline stage and

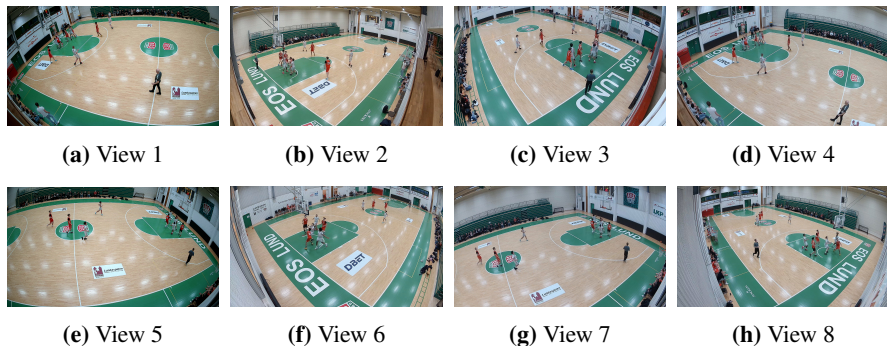


Figure 3.2 Example frames from eight camera views in the IK EOS dataset.

to demonstrate the potential of the proposed approach, rather than as a benchmark for rigorous quantitative evaluation.

3.2 IK EOS Dataset

The Sony internal dataset, made available during the mid-to-late phase of the project, constitutes the primary dataset for system evaluation. It consists of full broadcast footage of a single basketball game (*IK Eos Lund all-footage*) recorded across four periods (Period 1–4) by eight synchronized cameras. The cameras are evenly partitioned into four left-half-court views and four right-half-court views, providing dense multi-angle coverage of each play.

To better illustrate the spatial coverage and view differences in the dataset, Figure 3.2 shows example frames from eight different camera views in the IK EOS dataset. It can be observed that different views vary significantly in camera angle, occlusion, and object scale. These differences make multi-view modeling more challenging and also highlight the importance of using information from multiple cameras for complementary reasoning.

Manual event annotation. Although raw scoreboard and timing information accompanies the footage, fine-grained event annotations are largely absent in the original data. To enable supervised evaluation of action detection and best-camera selection, all successful scoring-related in-game events—specifically including 1-point free throws, 2-point field goals, and 3-point field goals—were manually annotated by the authors. Non-scoring events (such as fouls, turnovers, or timeouts) were excluded from this annotation phase as they do not directly trigger scoreboard updates. Each annotated event is represented as a structured record containing a unique identifier, start and end timestamps, the team involved, an event type description, the period number, and the score state at the time of the event. The annotation schema is illustrated below:

```
{
  "id": 13318658,
  "startedAt": "2026-03-08T18:10:05.000000Z",
  "teamName": "IK Eos Lund P11 Vit",
  "description": "3 points",
  "eventTypeId": 103,
  "period": 4,
  "currentScore": {
    "home": 100,
    "away": 56
  },
  "endedAt": "2026-03-08T18:10:07.700000Z",
  "scoreboardchanged": "2026-03-08T18:10:08.014000Z"
}
```

The `startedAt` and `endedAt` fields define the temporal extent of the event in the broadcast feed, while the `scoreboardchanged` timestamp marks the moment at which the score display is updated. The latter is used as the reference `TriggerTime` in the pipeline, since the scoreboard change is the signal that the data acquisition module observes in practice (Section 4.1).

Event filtering for highlight generation. Since the goal of the pipeline is to produce highlight videos rather than exhaustive game logs, the annotated events were filtered according to standard practice in basketball highlight production. Target events for evaluation are defined exclusively as successful two-point and three-point field goals. One-point free-throw events are excluded from the analysis, as such events typically involve static player positioning and limited visual intensity, which are inconsistent with the high-energy, dynamic nature of highlight reels in sports broadcasting[61]. This exclusion ensures that the generated highlights prioritize events with significant game-changing impact and visual appeal. After filtering, the dataset contains 43 two-point events and 12 three-point events, yielding a total of 55 scoring events distributed across the four periods. The event distribution is summarized in Figure 3.3.

The IK EOS dataset thus provides the dense multi-camera coverage and reliable event annotations required for meaningful quantitative evaluation of the action detection, retrieval, and best-camera selection stages, complementing the qualitative validation obtained on APIDIS.

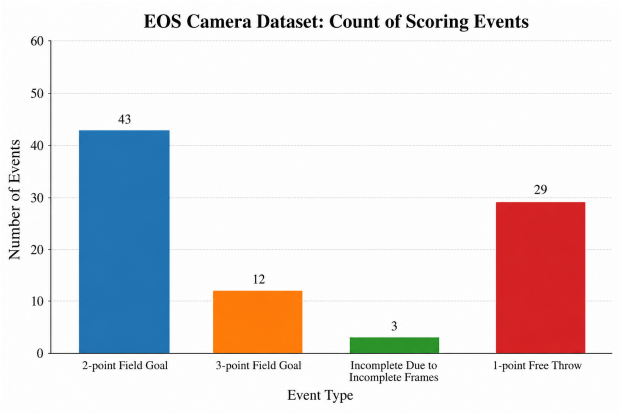


Figure 3.3 Distribution of annotated scoring events in the IK EOS dataset after filtering. Two-point and three-point successful field goals are retained, while one-point free-throw events are excluded from highlight generation.

4

Methodology & System Architecture

4.1 Data Acquisition Module

The data acquisition module prepares the inputs required by the downstream stages of the pipeline. Two pieces of information are extracted for every event of interest: a temporal anchor that marks when the event occurred, and a spatial label that identifies the half of the court where the play took place.

4.2 Trigger Time Extraction

For the IK EOS dataset, each annotated event contains a `scoreboardchanged` field. These raw timestamps were originally captured by Sony’s proprietary Optical Character Recognition (OCR) data acquisition system, and we subsequently extracted them from the system logs during the manual event annotation process (detailed in Section 3.2). Consequently, in the current implementation of our prototype pipeline, the trigger times adopted as the temporal index, represented as

$$\mathcal{T} = \{t_1, t_2, \dots, t_M\}, \quad (4.1)$$

are technically sourced from these manually curated event metadata. This design choice isolates the downstream action detection and view selection modules from potential OCR detection noise, allowing for a rigorous and independent evaluation of the core multimodal reasoning models.

To transition this research prototype into a fully autonomous, production-ready system in practical applications, a real-time automated OCR system (such as the one utilized in Sony’s acquisition logs) can be directly integrated into the video ingestion pipeline. This would allow the system to dynamically detect scoreboard changes on the fly and trigger the processing modules automatically, thereby achieving complete end-to-end automation.

4.3 Active Half-Court Detection

To identify the active side of the court for each extracted trigger event, a spatial heuristic based on player distribution is implemented. As reviewed in Section 2.2, the real-time YOLOv8 object detection model is deployed to localize player bounding boxes within the main broadcast view. For each trigger time t_i , the spatial coordinates of detected players are evaluated against the center line of the court. The half-court containing the majority of player detections is identified as the active side, generating a binary spatial label:

$$s_i \in \{\text{Left}, \text{Right}\}. \quad (4.2)$$

This lightweight spatial distribution heuristic effectively constrains the subsequent retrieval of candidate camera views to the relevant half of the court, avoiding the high computational complexity associated with explicit ball tracking.

4.4 Action Recognition with TSM

The action detection module adopts the Temporal Shift Module (TSM) [13] as its discriminative backbone. TSM augments a standard 2D convolutional network with temporal reasoning by shifting a fraction of the feature channels forward and backward along the temporal axis prior to each convolution, so that information is exchanged between neighboring frames with no additional parameters and negligible computational overhead. This approximates the spatio-temporal receptive field of a 3D convolution at the cost of a 2D network, yielding the high-throughput inference required in our multi-view setting, where the detector must scan synchronized streams as the lightweight first stage of the hierarchical data-reduction strategy. TSM is therefore employed as an efficient discriminative baseline whose temporal modeling is repurposed for action localization around each scoreboard trigger, in contrast to the heavier foundation-model branch reserved for fine-grained grounding.

The model is implemented using MMAction2 [62], an open-source video understanding toolkit based on PyTorch. The toolkit provides a unified framework for tasks such as action recognition, spatio-temporal action detection, and temporal action localization. MMAction2 was adopted for its following characteristics:

Reference Implementations: A range of state-of-the-art video understanding models, including TSM, is provided with reproducible training and evaluation configurations, which reduces implementation overhead.

Modular Architecture: Components such as backbones, recognizers, and data pipelines are decoupled, allowing individual modules to be customized and reused and thereby facilitating rapid prototyping.

Extensive Tooling: A comprehensive set of analysis utilities, including visualizers and evaluators, is included to streamline model troubleshooting and performance comparison.

OpenMMLab Integration: As part of the OpenMMLab ecosystem, the toolkit follows standardized interface conventions, enabling interoperability with other libraries and lowering the overhead of cross-module integration.

4.5 Internvideo2

4.5.1 Task Definition

The core of our automated sports highlight generation pipeline is a Moment Retrieval task. In this practical context, the objective of the MR task is to accurately localize the exact temporal boundaries of specific sporting events within an untrimmed video.

Formally, given an untrimmed video sequence V from our dataset, which consists of T frames or video segments denoted as $V = \{v_1, v_2, \dots, v_T\}$, and a specific target event Q (e.g., a predefined highlight label such as a shot), the system aims to predict the temporal window of this event. The output is defined by a start timestamp and an end timestamp, denoted as $(\tau_{\text{start}}, \tau_{\text{end}})$, where $0 \leq \tau_{\text{start}} < \tau_{\text{end}} \leq T$.

Since the pipeline processes synchronized multi-angle camera feeds, the primary goal of this task is to analyze these synchronized visual inputs and accurately output the $(\tau_{\text{start}}, \tau_{\text{end}})$ boundaries. These predicted boundaries are then directly used to clip the continuous video streams into short, distinct highlight segments.

4.5.2 Overall Framework

The proposed automated highlight generation pipeline is designed as a three-stage architecture: independent feature extraction, cross-modal fusion, and temporal boundary prediction.

First, to ensure computational efficiency, the system extracts modality-specific features offline. The raw video stream and the natural language query are processed independently using a pre-trained InternVideo2 encoder. This yields separate spatio-temporal video features and a distinct semantic text vector, with no early interaction between the two modalities.

Second, these independent features are fed into the CG-DETR network for cross-modal fusion. To retain absolute temporal context, the video sequence is augmented with temporal positional embeddings. Specifically, a set of sinusoidal (sine/cosine) positional vectors is generated based on the time index of each video clip. These position vectors are then directly added (element-wise) to the corresponding multimodal video features extracted by InternVideo2. By doing so, the absolute chronological order is explicitly injected into the video sequence, enabling

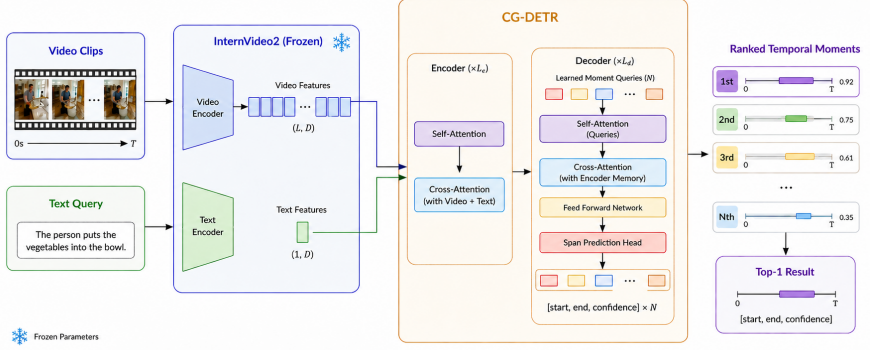


Figure 4.1 The proposed framework for the video grounding task

the subsequent CG-DETR layers to perceive the continuous flow of time and effectively capture long-range temporal dependencies in fast-paced basketball games. A text-to-video cross-attention mechanism is then applied, where the text feature serves as the Query (Q) and the video features act as the Key (K) and Value (V). This allows the network to dynamically extract a text-aware video representation.

Finally, the fused context is processed by an anchor-free temporal grounding head. Using a small set of learnable queries, the decoder directly regresses the temporal boundaries alongside an associated confidence score. The model outputs a center-width pair that is mapped to the final $(\tau_{start}, \tau_{end})$ timestamps, and the window with the highest confidence is selected to execute the actual video clipping.

4.5.3 Visual Feature Extraction

To preserve the temporal structure of long untrimmed videos while maintaining computational efficiency, we adopt a clip-based video representation strategy. Given an input video window, the video is uniformly divided into L non-overlapping clips, where each clip corresponds to a short temporal segment of the original video.

For each clip, a fixed number of frames are uniformly sampled along the temporal dimension and resized to a unified spatial resolution before being fed into the video encoder. The number of sampled frames is selected as a trade-off between temporal modeling capability and computational cost.

We employ the video encoder from InternVideo2[14], which is built upon a Video Transformer architecture for joint spatio-temporal representation learning. Spatially, each frame is decomposed into patch embeddings to capture local visual patterns. Temporally, self-attention mechanisms are utilized to model motion dynamics and long-range dependencies across frames. The encoder aggregates the

spatio-temporal information within each clip into a compact D -dimensional feature vector.

After feature extraction, the representations of all clips are stacked sequentially to form a video feature matrix:

$$V = [v_1, v_2, \dots, v_L] \in \mathbb{R}^{L \times D},$$

where each feature vector v_i corresponds to a specific temporal position in the video. This sequential representation preserves the temporal structure of the original video and serves as the visual input for the subsequent CG-DETR grounding module.

4.5.4 Textual Feature Encoding

To maintain semantic consistency during training and inference, we adopt a closed-vocabulary query design. Specifically, textual queries are constructed from a predefined template set describing basketball-related actions and scoring events. During dataset preparation, each annotated event is assigned to one corresponding query template. The same fixed query set is used throughout both training and inference.

The textual encoder of InternVideo2 is employed to extract semantic representations from input queries. The encoder is built upon a Transformer architecture, where the input sentence is first tokenized and then processed through multiple layers of self-attention for contextual semantic modeling.

After encoding, the token-level features are aggregated through a pooling operation to obtain a compact D -dimensional sentence representation:

$$q \in \mathbb{R}^{1 \times D}.$$

The dimensionality of the textual representation is identical to that of the visual features, as InternVideo2 aligns video and text representations into a shared embedding space through contrastive video-text pretraining.

To improve computational efficiency, textual features are extracted and cached offline before training begins. During both training and inference, the precomputed query embeddings are directly loaded without repeated encoding.

Unlike token-level sequence representations commonly used in large language models, the textual encoder outputs a single pooled semantic vector instead of a token sequence. This compact query representation serves as the query embedding for the subsequent cross-modal grounding module in CG-DETR.

4.5.5 Encoder-Decoder Architecture

The extracted visual and textual feature are processed by the encoder-decoder architecture of CG-DETR for multimodal temporal reasoning and moment localization. Given the video feature sequence

$$V = [v_1, v_2, \dots, v_L] \in \mathbb{R}^{L \times D},$$

and the textual representation

$$q \in \mathbb{R}^{1 \times D},$$

the encoder first performs temporal context modeling and text-guided feature fusion, followed by moment prediction in the decoder.

Encoder. The encoder receives the video feature sequence extracted by Intern-Video2 together with the textual query embedding. The video representations first undergo self-attention-based temporal modeling, where each clip attends to all other clips in the sequence to capture long-range temporal dependencies and contextual relationships across the video.

To explicitly encode temporal positional information, Temporal Endpoint Features (TEF) are incorporated into the video representation. For a video segmented into a sequence of T clips, the TEF for the i -th clip is defined as a two-dimensional vector $\mathbf{t}_i = [s_i, e_i] \in \mathbb{R}^2$, where s_i and e_i denote the normalized start and end time ratios within the range $[0, 1]$, respectively. These positional features are projected via a linear layer and then concatenated to the respective visual feature $\mathbf{v}_i \in \mathbb{R}^D$ extracted by the backbone network. The enhanced clip representation $\tilde{\mathbf{v}}_i$ is formulated as:

$$\tilde{\mathbf{v}}_i = [\mathbf{v}_i; \text{Linear}(\mathbf{t}_i)] \quad (4.3)$$

By providing the exact normalized boundaries along the video timeline, TEF effectively compensates for the limited sensitivity of vanilla Transformer architectures to absolute temporal locations, thereby facilitating more precise temporal boundary regression in downstream tasks.

After temporal modeling, cross-attention is applied for text-guided feature interaction. Specifically, the textual representation serves as the query, while the video feature sequence acts as the key and value:

$$Q = \tilde{V} \in \mathbb{R}^{L \times D}, \quad (4.4)$$

$$K = q \in \mathbb{R}^{1 \times D}, \quad (4.5)$$

$$V_A = q \in \mathbb{R}^{1 \times D}, \quad (4.6)$$

where V_A is introduced to distinguish the attention value from the baseline video representations.

Through this operation, the textual query performs semantic retrieval over the video sequence, assigning higher attention weights to clips that are more relevant to the query semantics. The encoder finally outputs text-aware video representations:

$$F \in \mathbb{R}^{L \times D}.$$

Each position in the fused sequence simultaneously encodes temporal visual dynamics and query-guided semantic information, which are then passed to the decoder for temporal localization.

Decoder. The decoder takes the encoder output together with N learned moment queries for temporal localization. These moment queries are randomly initialized learnable embeddings that are progressively optimized during training.

Within each decoder layer, self-attention is first applied among the moment queries, enabling different queries to interact with each other and distribute themselves across diverse temporal candidates. This mechanism helps reduce redundant predictions and encourages coverage of different potential moments.

Subsequently, cross-attention is performed between the moment queries and the encoder output representations. Each query retrieves the most relevant temporal context from the text-aware video features, enabling query-guided temporal localization.

After attention interaction, a feed-forward network (FFN) is applied to each query for nonlinear feature transformation and refinement.

After multiple decoding layers, each moment query predicts one candidate temporal segment together with its confidence score:

$$\hat{y}_i = [t_s, t_e, c],$$

where t_s and t_e denote the predicted start and end timestamps, and c represents the confidence score.

Finally, all candidate predictions are ranked according to their confidence scores, and the top-ranked prediction is selected as the final grounding result.

4.5.6 Loss

To optimize temporal localization and enhance cross-modal representation learning, the framework is trained end-to-end using a joint objective function. The overall training loss $\mathcal{L}_{\text{total}}$ is formulated as a weighted combination of four distinct components: Span Regression loss, Generalized IoU loss, Classification loss, and Orthogonality regularization:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{span}} \mathcal{L}_{\text{span}} + \lambda_{\text{giou}} \mathcal{L}_{\text{giou}} + \lambda_{\text{label}} \mathcal{L}_{\text{label}} + \lambda_{\text{orth}} \mathcal{L}_{\text{orthogonal_dummy}} \quad (4.7)$$

where λ_{span} , λ_{giou} , λ_{label} , and λ_{orth} are the scaling coefficients for their respective loss terms and matching costs [54]. The detailed formulation and intuition of each component are described below:

- **Span Regression Loss ($\mathcal{L}_{\text{span}}$).** This loss is implemented as an L_1 regression loss on the predicted temporal span parameters, including the center coordinate and duration of the target moment. It encourages the model to accurately regress the temporal boundaries of the action segment:

$$\mathcal{L}_{\text{span}} = \|\hat{c} - c\|_1 + \|\hat{w} - w\|_1 \quad (4.8)$$

where (c, w) and $(\hat{c}, \hat{w})$ denote the ground-truth and predicted center-width representations, respectively.

- **Generalized IoU Loss ($\mathcal{L}_{\text{giou}}$).** To further improve temporal boundary alignment, the model adopts a Generalized Intersection-over-Union (GIoU) loss [63] between the predicted and ground-truth temporal spans. Compared with pure regression losses, GIoU directly optimizes the quality of temporal overlap and effectively alleviates boundary ambiguity. Specifically, for a predicted temporal span $[\hat{s}, \hat{e}]$ and the corresponding ground-truth span $[s, e]$, the loss is formulated as:

$$I = \max(0, \min(\hat{e}, e) - \max(\hat{s}, s)) \quad (4.9)$$

$$U = (\hat{e} - \hat{s}) + (e - s) - I \quad (4.10)$$

$$\text{IoU} = I/U \quad (4.11)$$

$$C = \max(\hat{e}, e) - \min(\hat{s}, s) \quad (4.12)$$

$$\text{GIoU} = \text{IoU} - (C - U)/C \quad (4.13)$$

$$\mathcal{L}_{\text{giou}} = 1 - \text{GIoU} \quad (4.14)$$

where I and U represent the intersection and union of the intervals, respectively, and C denotes the length of the smallest enclosing interval.

- **Classification Loss ($\mathcal{L}_{\text{label}}$).** A cross-entropy classification loss is applied to distinguish foreground moments from background segments. This loss supervises the model to correctly identify whether a temporal proposal corresponds to the target action described in the query. Since the basketball videos are captured from relatively fixed viewpoints with limited background variation, this loss can effectively enhance foreground-background discrimination. The binary cross-entropy loss over N total predicted queries (including positive and negative samples after matching) is defined as:

$$\mathcal{L}_{\text{label}} = -\frac{1}{N} \sum_{i=1}^N [y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)] \quad (4.15)$$

where $\hat{p}_i$ is the predicted foreground probability for the i -th query, and $y_i \in \{0, 1\}$ represents the ground-truth label. For an ambiguous prediction slot that spans both action and background, its target assignment y_i is strictly binarized to either 0 or 1. The classification loss heavily penalizes the resulting underconfident or false-positive probability $\hat{p}_i$, thereby pushing the network to separate the mixed features and improve classification accuracy.

- **Orthogonality Loss ($\mathcal{L}_{\text{orthogonal_dummy}}$).** Following the CG-DETR design, the textual query is augmented with a set of learnable *dummy tokens* alongside the *content tokens* that carry the actual query semantics. The dummy tokens act as attention absorbers: video clips with no relevant textual correspondence can attend to them instead of being forced to align with the content tokens, which suppresses spurious query-frame matches on irrelevant background.

An orthogonality regularization term is then imposed between the dummy tokens and content tokens. This constraint encourages the dummy tokens to capture complementary information while minimizing feature redundancy in the learned multi-modal representations. Formally, the text query is decomposed into K sub-query vectors $\mathbf{q}_1, \dots, \mathbf{q}_K$, forming a matrix $\mathbf{Q} \in \mathbb{R}^{K \times d}$ where each row is L_2 normalized. To prevent information redundancy among sub-queries, the loss is computed as:

$$\mathcal{L}_{\text{orthogonal_dummy}} = \|\hat{\mathbf{Q}}\hat{\mathbf{Q}}^\top - \mathbf{I}\|_F^2 \quad (4.16)$$

where $\hat{\mathbf{Q}}$ denotes the L_2 -normalized sub-query matrix, $\mathbf{I}$ is the identity matrix, and $\|\cdot\|_F$ represents the Frobenius norm. This loss penalizes non-orthogonal sub-queries and reaches 0 when they are perfectly mutually orthogonal.

4.6 Intelligent Best-Cam Selection via VLMs

4.6.1 Task Formulation

Given a set of multi-camera video streams $\{C_1, C_2, \dots, C_N\}$ and a target basketball event defined by a temporal interval (T_s, T_e) , the goal is to automatically select the most informative camera view that best captures the event. We formulate this task as a view ranking problem, where each camera provides a candidate visual representation of the same event. The optimal view is selected based on both semantic relevance and visual quality.

4.6.2 Candidate View Construction

For each camera, we construct a compact representation of the event by sampling a small number of representative frames within the temporal interval (T_s, T_e) .

These sampled frames serve as a summary of the visual content observed from each viewpoint during the event. Instead of processing full video sequences, we use these frames to efficiently represent each camera view while preserving key semantic information such as player position, ball location, and action context.

Each camera is thus treated as a candidate view for subsequent ranking.

We further organize all sampled frames from different cameras into a single multimodal input, and jointly provide them to a Vision-Language Model (VLM), enabling unified cross-view reasoning within a single inference pass.

4.6.3 Vision-Language Model based View Ranking

Once the relevant half-court has been identified by the side-classification stage (Section 4.3), the set of candidate cameras is restricted to those covering that half-court. A Vision-Language Model (VLM), namely Qwen2.5-VL, is then employed to perform semantic ranking over these candidate views.

For each candidate camera, a single representative frame is sampled at the temporal midpoint of the target event window. To guarantee that all cameras are sampled at the same real-world instant, the frame index of each camera is derived from the period start time together with its leading-frame offset, thereby compensating for the differing recording start points across devices. The sampled frames are down-scaled with their aspect ratio preserved and JPEG-compressed to limit the input payload.

The resulting frames are assembled into a single multi-image prompt and submitted to the model in one request, with each image labelled by its camera index. The model is instructed to rank the cameras by their suitability for capturing the complete shooting action, according to the following criteria:

- Visibility and scale of the players and ball, such that action details remain discernible;
- Spatial proximity to the half-court where the shot occurs, penalising distant views from the opposite half;
- Clarity of the action, free from motion blur, under-exposure, occlusion, or signal loss;
- Absence of severe geometric distortion, such as overhead fisheye perspectives that obscure player movement.

Rather than relying on handcrafted heuristics or explicit geometric rules, the VLM performs high-level semantic reasoning jointly over all candidate views and outputs an ordered ranking of camera indices from the most to the least informative.

4.6.4 Best View Selection

The ranking returned by the VLM is parsed into a per-camera ordering, and any candidate camera omitted from the model output is assigned the lowest priority. The top-ranked camera is then selected as the primary view for the event. This selected view is used as the principal perspective for downstream tasks such as event understanding, temporal grounding, and highlight generation.

4.7 Design Methodology for the Pipeline Visualization Interface

To provide a comprehensive environment for validating the autonomous sports production pipeline, a specialized visualization interface was conceptualized. The primary goal of this interface is to facilitate **system observability**, allowing the researcher to monitor how the multimodal data-reduction strategy and the InternVideo2-driven grounding engine operate in a real-world multi-view context.

4.7.1 Information Architecture and Layout Logic

The interface is designed as a centralized "Control Tower," structured to provide a multi-dimensional view of the pipeline's internal state. The information architecture is partitioned into four primary zones:

1. **System Control and Telemetry Zone:** This area provides high-level administrative controls and real-time hardware monitoring. By visualizing GPU utilization and VRAM occupancy, the interface allows for the empirical assessment of the pipeline's computational cost—a core research objective.
2. **Multi-View Matrix:** The core visual component is a synchronized grid capable of displaying up to eight concurrent camera feeds. This layout is critical for verifying spatial consistency across the court and analyzing how the system handles occlusions through viewpoint complementarity.
3. **Decision and Heuristic Overlay:** This zone visualizes the output of the hierarchical reduction logic. Specifically, it indicates which half of the court is identified as "active" via the YOLO-based side gate and highlights the "Best Camera" candidate proposed by the Vision-Language Model.
4. **Temporal Grounding and Log Stream:** A synchronized command console and timeline provide a detailed audit trail of every system event. This includes trigger timestamps, InternVideo2 saliency scores, and the final predicted grounding intervals.

4.7.2 Synchronization Paradigm for Multi-Angle Verification

A key methodological requirement of the interface is the implementation of a **Unified Temporal Anchor**. To accurately evaluate temporal grounding, the interface must ensure that all video feeds and algorithmic outputs (saliency curves and bounding boxes) are perfectly aligned. This design allows for "Synchronized Scrubbing," where an evaluator can move through the match history and see the simultaneous state of all cameras alongside the model's confidence scores at any given frame.

4.7.3 Human-Centric Evaluation Framework

The interface serves as the primary tool for the "human-in-the-loop" validation phase. It is architected to support subjective scoring protocols, where expert viewers can compare the VLM-selected "Best-Cam" against alternative views. By providing a standardized environment for this qualitative assessment, the interface transforms subjective aesthetic judgment into quantifiable data that can be used to benchmark the system's directing logic.

5

Implementation

The practical implementation details of our proposed basketball highlight generation pipeline are elaborated in this chapter. Rather than presenting isolated algorithms, each module is implemented to seamlessly fit into the end-to-end synchronized workflow conceptually outlined in Figure 1.1 (see Section 1.3). The following subsections specify the software architecture, the specific execution frameworks for TSM and InternVideo2, and the deep-learning-based decision-making process for camera selection.

5.1 System Overview

The system is designed as a modular pipeline that transforms multi-camera broadcast footage of basketball games into a concise highlight reel without manual intervention. As illustrated in Figure 1.1, the pipeline is composed of five sequential stages, each addressing a well-defined sub-problem.

The pipeline begins with **scoreboard change detection**, which monitors the score display in the broadcast feed and records a list of `TriggerTime` events whenever the score changes. Each trigger time marks a candidate moment of interest (e.g., a successful shot) and serves as a temporal anchor for subsequent processing. A complementary **active half-court detection** step, based on YOLOv8, determines whether the scoring action occurred on the left or right half of the court, which constrains the camera views considered in later stages.

In the second stage, **action detection** is performed around each trigger time using two state-of-the-art video understanding models, TSM(MMAAction2) [13, 62] and InternVideo2 [14], executed in parallel for comparison. Each model produces a set of action time intervals labelled with categories such as *goal*.

The third stage, **multi-angle clip retrieval**, uses the detected action intervals together with the active half-court detection result to extract synchronized clips from all relevant half-court camera views. For each action, a collection of candidate clips covering the event from multiple perspectives is produced.

In the fourth stage, **cross-model best-camera selection** feeds the retrieved multi-angle clips into the Qwen 2.5-VL 7B vision-language model [64], which evaluates the clips and selects the most informative camera angle for each action. This stage exploits the complementary outputs of the two action detectors and resolves disagreements by deferring to the visual judgement of the VLM.

Finally, the **post-processing** stage extracts the selected best-camera clip for each event, cut directly to the temporal boundaries predicted by the grounding stage.

5.2 Dataset Processing

5.2.1 Missing Frame Issue in APIDIS Dataset

During the utilization of the APIDIS dataset, we accounted for the frame-drop issues documented in the initial video acquisition. According to the official report regarding the initial video acquisition of the APIDIS dataset, the criteria, prevalence, and underlying causes of the frame-drop issues are summarized as follows:

- **Detection Criterion:** The baseline frame rate of the videos ranges between 22 and 23 frames per second (FPS), yielding a normal inter-frame delay of approximately 0.045 seconds. A frame-drop event is formally identified when the delay between two consecutive frames exceeds three times this normal threshold.
- **Prevalence and Severity:** Frame dropping is a ubiquitous phenomenon rather than an isolated incident, persistently occurring across all cameras throughout the four quarters of the match. However, the severity varies significantly among devices. Cameras 2, 3, and 4 exhibit the poorest reliability, frequently suffering from severe delays exceeding 20 seconds. In contrast, Camera 5 demonstrates the highest stability with the lowest frame-drop frequency.
- **Causal Analysis:** The pervasive nature of this issue across all time periods (Quarters 1–4) indicates a systemic flaw inherent to the recording equipment or the capture environment. Furthermore, the distinct discrepancy in frame-drop frequencies among different cameras points to device-specific hardware instability, particularly pronounced in Cameras 2, 3, and 4.

The distribution of black frames across the 7 cameras is shown in the Figure 5.1, red part represents the black frames part.

5.2.2 Processing Strategy

As previously established in Section 3.1, the source videos suffer from frame-drop anomalies and severe flickering. To maintain strict temporal synchronization across

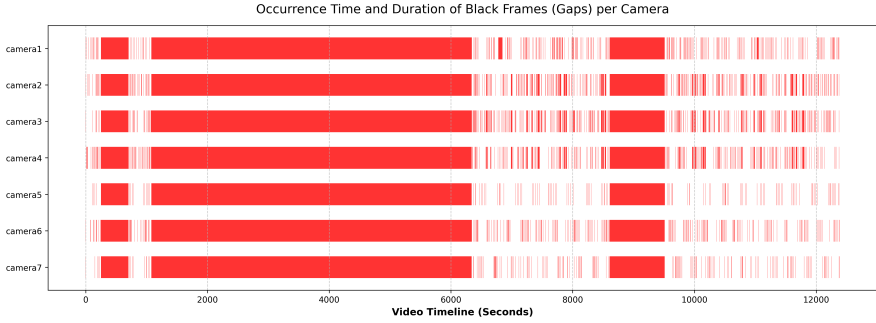


Figure 5.1 Black Frames Distribution

all camera views, we apply the identical dual-tier frame restoration strategy (comprising linear interpolation and black frame padding) detailed in the prior section.

While this processing protocol effectively alleviates temporary data loss, the extended informational voids caused by long gaps remain fundamentally irrecoverable. Therefore, to ensure a fair and rigorous evaluation, all video segments containing long gaps are excluded from the final testing dataset. This filtering step guarantees that the model performance is benchmarked exclusively on temporally continuous and reliable video sequences.

5.3 TSM(MMAction2)-based Grounding

The Temporal Shift Module (TSM) [13] with a ResNet-50 backbone serves as the baseline discriminative model for temporal action grounding. To implement a zero-shot, training-free comparison framework, an off-the-shelf TSM model pre-trained on the Kinetics-400 (K400) dataset [29] is utilized. Rather than fine-tuning a dedicated binary classifier, the pipeline repurposes the pre-trained action recognition capabilities of the network by extracting frame-level probabilities for basketball-related classes within the K400 label space, specifically targeting actions such as *shooting basketball*, *dunking basketball*, *playing basketball*, and *dribbling basketball*.

Notably, this baseline model is employed solely to demonstrate the feasibility of the proposed pipeline, and further specific performance optimization is beyond the scope of this work.

5.3.1 Temporal Synchronization and Event Triggering

The temporal grounding pipeline operates strictly on localized search windows initiated by scoreboard trigger events, avoiding continuous dense inference over the entire broadcast stream. Upon receiving an absolute UTC timestamp of a scoreboard change, $t_{\text{scoreboard}}$, the corresponding relative trigger time in seconds for the

selected camera stream, $t_{\text{trigger, cam}}$, is computed as:

$$t_{\text{trigger, cam}} = t_{\text{scoreboard}} - t_{\text{start, cam}} \quad (5.1)$$

where $t_{\text{start, cam}}$ denotes the absolute UTC start time of the selected camera video stream.

The temporal search range is defined as a fixed 10-second window preceding the trigger point:

$$T_{\text{search}} = [t_{\text{trigger, cam}} - \Delta t, t_{\text{trigger, cam}}] \quad (5.2)$$

where the prediction window Δt is set to 10.0 seconds.

5.3.2 Per-window scoring and peak-based segment selection

Within the search window $T_{\text{search}} = [t_{\text{trigger}} - \Delta t, t_{\text{trigger}}]$ ($\Delta t = 10$ s), the off-the-shelf TSM (ResNet-50, Kinetics-400) is applied in a sliding-window manner: each window spans 2.0 s and is advanced by a stride of 0.1 s, and 8 frames are uniformly sampled per window. The 400-dimensional class logits are converted to probabilities by a softmax, and the per-window basketball score $p(t)$ is the sum of the probabilities of the basketball-related Kinetics-400 classes *{shooting basketball (296), dribbling basketball (99), dunking basketball (107), throwing (357)}*.

The score sequence is Gaussian-smoothed ($\sigma = 1.5$) and an adaptive level $\theta = \mu + k\sigma$ ($k = 0.3$) is computed, where μ, σ are the mean and standard deviation of the smoothed scores in the window. Instead of thresholding the curve into contiguous proposals, a single segment is selected: (i) local maxima are detected with a minimum separation of 2 s; (ii) each peak is ranked by $p(t)/(1 + |t - t_{\text{trigger}}|/3)$, i.e. its score discounted by temporal distance to the trigger, and the top-ranked peak t^* is kept; (iii) the segment is grown outward from t^* while the score exceeds $\max(\theta, 0.70p(t^*))$, with the window-centre boundaries extended by 0.75 s on each side; and (iv) the duration is hard-capped at 5 s, re-centring on t^* if exceeded. This produces exactly one predicted segment ($\tau_{\text{start}}, \tau_{\text{end}}$) per event. A trigger-anchored window of length Δt serves only as a safety fallback when no window can be scored; in practice the peak selector almost always returns a segment, so this fallback is rarely exercised on the test split.

5.3.3 Hardware and Experimental Setup

All experiments in this Section were conducted on an HP ZBook Power 15.6 inch G9 Mobile Workstation PC. The system configuration comprises a 12th Gen Intel Core i7-12700H processor, 32 GB of system memory, and an Intel Iris Xe Graphics (ADL GT2) processing unit, running on Ubuntu 24.04.4 LTS under a 64-bit Wayland windowing environment.

5.4 InternVideo2-based Grounding

In our pipeline, InternVideo2 [14] is adopted as the foundational vision-language backbone, whose core MLLM architecture and pre-training capabilities have been detailed in Section 2.3.3. Here, we leverage its powerful video-text alignment and fine-grained spatiotemporal perception to extract rich multi-modal feature representations from the multi-angle basketball videos. These high-fidelity features serve as the direct input for the subsequent temporal action localization and correlation-guided detection stages.

5.4.1 Data Preparation

Since the original competition videos are extremely long, directly feeding them into the model is impractical. We utilize the precise event trigger timestamps provided by the dataset to extract local video segments using a sliding window approach.

The video extraction and segmentation process is structured as follows:

- **Window Construction:** Considering that basketball actions (e.g., jumping for a shot, ball in flight, and making the basket) typically occur before the scoreboard updates, we use the trigger time as a reference point. We extract 10 seconds before the trigger to construct a time window with a total length of 10 seconds:

$$Window = [t_{trigger} - 10, t_{trigger}]$$

- **Clip Segmentation:** Subsequently, we uniformly divide this 10-second window into 33 video clips, each 0.3 seconds long.
- **Supervision Signal Generation:** We then record the specific clips containing the actual actions based on the annotation information in the file, which serves as the supervision signal for subsequent training.

5.4.2 Extraction based on InternVideo2

To mitigate the GPU memory overhead of the 1B-parameter model, we freeze the weights of InternVideo2 and utilize it solely as an offline feature extractor to pre-extract multimodal features.

- **Video Feature Extraction:** For each 0.3-second clip generated from the previous step, we uniformly sample 4 frames and feed them into the visual encoder of InternVideo2. Each clip is encoded into a 512-dimensional vector. Consequently, a complete 10-second video window is represented by a feature matrix of shape (33, 512).
- **Text Feature Extraction:** Simultaneously, we design specific natural language queries for different action categories to provide precise semantic guidance. The exact text queries are defined as follows:

Action Category	Text Query
Standard scoring (e.g., two-point shot)	“basketball player shooting and scoring”
Three-point shot	“three point shot basketball”

Table 5.1 Natural language queries for different action categories.

These class-specific text descriptions are processed by the InternVideo2 text encoder to output a semantic feature vector of shape (1,512).

During its pre-training phase, InternVideo2 has already aligned these two modalities into a unified semantic space. Finally, we serialize and save these extracted features as PyTorch tensor files (.pt) to decouple the feature extraction stage from the downstream model training process.

5.4.3 Downstream Temporal Grounding Network: CG-DETR

After extracting the features, rather than simply relying on feature similarity ranking, we introduce CG-DETR, a Transformer-based regression model, to serve as the downstream temporal grounding network. This piece of code is also included in the open-source code of InternVideo2.

Unlike the InternVideo2 feature extractor whose weights are completely frozen, this downstream CG-DETR network is specifically trained in our study using the dataset from Sony.

- **Cross-Modal Encoder:** We concatenate the (33,512) video features (which include temporal positional encodings) with the (1,512) text features, and feed them into a 2-layer Transformer encoder. The self-attention mechanism facilitates deep contextual interaction between the visual and linguistic features across the global receptive field.
- **Query Decoder and Boundary Regression:** We utilize a 2-layer Transformer decoder initialized with 5 learnable queries, which act as “candidate detectors”. During the decoding process, each detector uses cross-attention to search for the target within the cross-modal features. Ultimately, it outputs a pair of normalized values: the temporal center coordinate (*center*) and the temporal duration (*width*). Through this regression mechanism, the model directly predicts continuous time boundaries.

Loss Functions. Regarding the loss scaling factors in Eq. 4.7, we retained the default configuration provided in the official InternVideo CG-DETR implementation to ensure stable convergence. Specifically, the loss weights were set to $\lambda_{span} = 10$ and $\lambda_{label} = 4$ to emphasize precise boundary regression and robust foreground discrimination, while the weights for the remaining components (λ_{giou} and λ_{orth}) were kept at their default value of 1.

5.4.4 Implementation Details

All model training and inference in Section 5.4 were conducted on a machine equipped with an NVIDIA GeForce RTX 5060 (8GB VRAM).

5.5 Best Camera Selection

The system takes synchronized multi-camera video streams and event timestamps as input, and outputs the selected best camera view for each event.

Frame Sampling and Preprocessing. For each camera, we extract frames within the event interval and uniformly sample a fixed number of representative frames. In all experiments, we set the number of sampled frames per camera to $N = 1$ to balance efficiency and performance.

To reduce communication cost with the Vision-Language Model, all frames are resized to 480×270 resolution and compressed using JPEG encoding with quality factor 50.

Multi-camera Inference. All sampled frames from different cameras are aggregated into a single multimodal input and sent to the Vision-Language Model in a single inference pass. Instead of relying on tedious pairwise scoring, this design forces the model to perform a joint comparison across all available camera views simultaneously, effectively compelling it to select the single best perspective.

Vision-Language Model Configuration The Qwen2.5-VL-7B architecture is adopted as the primary backbone Vision-Language Model for the automated view selection module. Renowned for native spatiotemporal understanding and the ability to process dynamic high-resolution visual sequences, Qwen2.5-VL-7B serves as the decision-making core of the pipeline. In the practical implementation, the VLM is configured with a temperature of 0.001 and a top-p value of 0.9 to suppress hallucinations and ensure stable, deterministic ranking decisions. To optimize hardware efficiency and comply with local workstation constraints, the model is deployed using INT4 quantization. This setup significantly minimizes the VRAM footprint while preserving rich spatial-semantic reasoning capabilities.

Output Generation The final selected view is the top-ranked camera predicted by the VLM. The system outputs both the corresponding video file and the camera identifier, which are used for downstream event analysis.

5.6 Implementation of the Visualization System

The visualization system is realized through a decoupled full-stack architecture, integrating a high-performance Python backend with a reactive web frontend. This implementation serves as the operational core of the research, bridging the gap between heavy AI inference and real-time human observation.

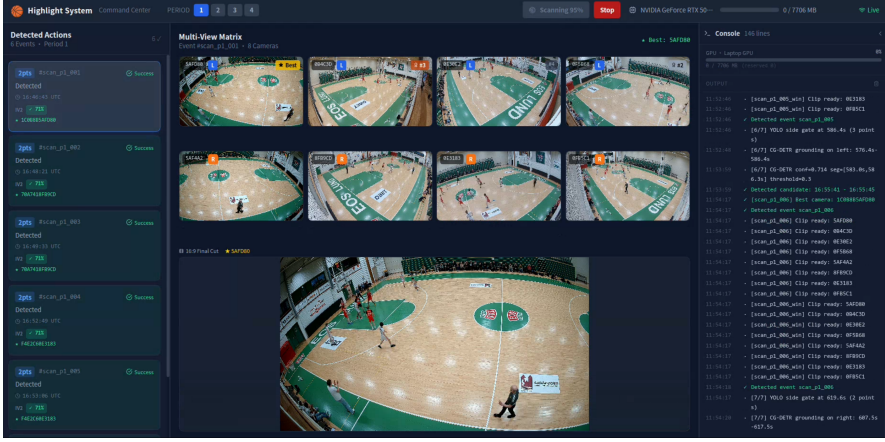


Figure 5.2 The user interface (UI) dashboard of our automated highlight clipping system. The interface is optimized for real-time multi-angle streaming, featuring an execution console (right), an indexed event archive (left), and a multi-view synchronization player (center) that emphasizes the optimal camera view.

5.6.1 User Interface and Workflow Demonstration

To facilitate real-time monitoring and interactive verification, we developed a comprehensive control center interface for the highlight clipping system. As illustrated in Fig. 5.2, the UI is split into three functional areas that intuitively visualize the operational pipeline:

- **Console Log (Right Panel):** When the automated clipping process initiates, this panel displays real-time execution logs, showcasing pipeline activities such as YOLO side-gate temporal localization, CG-DETR grounding log, and video frame extraction timestamps.
- **Detected Actions (Left Panel):** Once an event detection and clipping sequence completes, the successfully processed highlights are archived chronologically as actionable list items on the left. Each entry provides atomic meta-data including event type (e.g., 2-point shot), detection confidence, and a unique event hash.
- **Multi-View Video Player (Center Panel):** Clicking on any recorded event from the left panel dynamically populates the central viewing matrix with the corresponding synchronized footage. This player is subdivided into two tiers:
 - *Upper Tier (Multi-View Matrix):* Simultaneously streams synchronized clips across all 8 peripheral camera angles, labeled with distinct device IDs to ensure full-court coverage verification.

- *Lower Tier (Final Cut Preview)*: Automatically highlights and enlarges the dynamic stream from the designated "Best Camera" (e.g., Camera 5AFD80), offering an optimized, broadcast-ready viewpoint of the specific moment.

5.6.2 Asynchronous Backend Orchestration

The backend is implemented as an asynchronous service designed to manage high-latency AI tasks without compromising system responsiveness. To ensure fluid operation, the system employs a **non-blocking orchestration strategy**:

- **Decoupled Execution**: Complex pipeline operations, including foundational model reasoning and video clipping, are offloaded to a dedicated sequential execution pool. This prevents the inference heavy-lift from interrupting the delivery of real-time telemetry data.
- **Dynamic Temporal Mapping**: The system implements a frame-accurate retrieval mechanism that maps absolute UTC trigger points to camera-relative timestamps. By performing random-access sampling directly on multi-angle raw footage, the backend generates synchronized frame previews in memory, minimizing I/O overhead while ensuring spatial consistency across the multi-view matrix.

5.6.3 Real-time Telemetry and WebSocket Integration

To provide the system observability required for cost-efficiency analysis, a **bidirectional communication channel** is established between the server and the interface.

- **Live Hardware Monitoring**: The system utilizes a background telemetry loop that queries hardware-level device properties, including VRAM occupancy and GPU utilization. This data is broadcasted to the interface via a persistent socket connection, allowing for the empirical validation of the pipeline's computational footprint in real-time.
- **Asynchronous Audit Trail**: Every internal state transition—from initial event detection to successful grounding—is serialized and pushed through the data stream. This creates a live console that provides the researcher with immediate feedback on pipeline progress and potential grounding drift.

5.6.4 Reactive Frontend Architecture and Synchronization

The frontend interface is architected using a modern reactive paradigm, focusing on maintaining a single source of truth for the multi-camera environment.

- **Centralized State Orchestration**: A global state provider manages the complex metadata associated with each sporting event. This includes temporal

boundaries, grounding confidence scores, and camera configurations. By centralizing these parameters, the system ensures that any update—such as a user-triggered temporal seek—is propagated to all active camera components simultaneously.

- **Hierarchical Visualization and Selective Rendering:** To optimize cognitive ergonomics, the interface dynamically adjusts the visual priority of the camera grid. Views are programmatically styled based on heuristic labels (e.g., active vs. inactive half-court), while the primary viewing area is reserved for the model-selected optimal angle, facilitating rapid expert verification.

5.6.5 Synchronized Interactivity for Expert Validation

The implementation supports high-precision manual auditing through a synchronized scrubbing mechanism. Interacting with the global controller triggers a unified temporal update across the entire multi-camera array. This ensures that the human evaluator can verify the presence of an action or the occurrence of an occlusion across all 8 perspectives at the exact same millisecond, fulfilling the methodological requirement for a robust human-in-the-loop evaluation.

6

Evaluation and Results

6.1 Results on TSM(MMAAction2) Model with APIDIS Dataset

To establish a structural baseline for the pipeline, initial experiments were conducted on the APIDIS basketball dataset. The primary objective of this phase was to verify the feasibility of the "trigger-to-grounding" workflow rather than optimizing for recognition accuracy on this specific domain.

For this evaluation, the Temporal Shift Module (TSM) implementation within the TSM(MMAAction2) framework was utilized. Crucially, the model used was the **original pre-trained version** without any domain-specific fine-tuning or training on basketball-specific data. This "off-the-shelf" approach was selected to identify the performance floor of traditional discriminative models when tasked with action spotting in a multi-view environment.

To assess the system's performance, we focused on three custom metrics in addition to the standard Recall ($R@0.3$ and $R@0.5$): Precision, Total Predictions (Preds), and F1-Score.

- **Recall@IoU ($R@θ$):** A standard metric in temporal action localization that evaluates the system's coverage of actual events. A ground truth segment is successfully recalled if there exists a predicted segment with a Temporal Intersection over Union (tIoU) greater than or equal to a designated threshold $θ$ (e.g., $θ = 0.3$). It balances the positioning accuracy of the temporal boundaries with the overall detection rate.
- **Precision:** Unlike standard IoU evaluations, we defined Precision based on a strict temporal inclusion check. A predicted segment $[p_{start}, p_{end}]$ is considered a valid hit only if it falls entirely within the ground truth interval $[gt_{start}, gt_{end}]$, allowing for a slight temporal tolerance of 0.1 seconds:

$$p_{start} \geq gt_{start} - 0.1 \quad \text{and} \quad p_{end} \leq gt_{end} + 0.1 \quad (6.1)$$

While Recall measures whether an event was found at all, this strict Precision metric evaluates whether the pipeline avoids false positives and generates clean, tightly bound clips without over-reporting.

- **Total Predictions (Preds):** This tracks the overall count of highlight segments generated by the system. In our evaluation code, a grid search strategy filters multiple overlapping candidates and retains only the latest valid segment per event. As a result, the total number of predictions roughly equals the total number of actual events, unless the model fails to detect an event entirely.
- **F1-Score:** Calculated as the harmonic mean of Recall@0.3 and Precision:

$$F1 = \frac{2 \cdot R@0.3 \cdot \text{Precision}}{R@0.3 + \text{Precision}} \quad (6.2)$$

We utilized the F1-Score as the primary metric to rank different parameter configurations on our leaderboard. It provides a balanced assessment of the system’s ability to find the highlights (Recall) against its ability to keep the extractions accurate (Precision).

The performance of the TSM-based grounding module was evaluated under various hyperparameter configurations. Table 6.1 summarizes the performance metrics on the APIDIS dataset.

Table 6.1 Grounding performance of the original TSM(MMAAction2) model on the APIDIS dataset across different parameter configurations.

Configuration	Recall@0.3	Recall@0.5	Precision	F1-Score	Preds
k-0.5 / stride 0.1	58.0%	20.0%	40.3%	47.5%	72
k-1.0 / stride 0.1	54.0%	18.0%	40.3%	46.2%	67
k-1.0 / stride 0.2	54.0%	26.0%	39.7%	45.8%	68
k-0.5 / stride 0.2	54.0%	16.0%	38.0%	44.6%	71
k-0.3 / stride 0.1	54.0%	16.0%	36.0%	43.2%	75
k-0.3 / stride 0.2	52.0%	18.0%	36.6%	43.0%	71
k-0.5 / stride 0.3	50.0%	22.0%	35.7%	41.7%	70

The baseline hyperparameter configurations and search grids results for the TSM(MMAAction2) model tested on APIDIS dataset are summarized in Table 6.1. To establish a rigorous mathematical foundation for the event boundary detection algorithm, two key parameters are defined: the dynamic threshold multiplier k and the normalized sliding window stride ratio $stride$.

The parameter k represents the scaling coefficient of the standard deviation in the dynamic thresholding formula. To filter high-confidence action proposals from

the background noise, an adaptive event-triggering threshold θ is formulated as:

$$\theta = \mu + k \cdot \sigma, \quad (6.3)$$

where μ and σ denote the temporal mean and the standard deviation of the smoothed classification score sequence within the 10-second detection window, respectively. A lower k value (e.g., 0.3) yields a highly permissive threshold that maximizes temporal recall, whereas a higher k value (e.g., 1.5) establishes a conservative boundary that minimizes false positives.

The parameter *stride* designates the temporal displacement of the sliding window, expressed as a normalized ratio of the total detection duration D (where $D = 10$ seconds). Specifically, the physical step size of the sliding window Δt is calculated as:

$$\Delta t = \text{stride} \cdot D. \quad (6.4)$$

For instance, configuring *stride* = 0.2 dictates a temporal step size of 2.0 seconds, guaranteeing an 80% temporal overlap between consecutive candidate proposals. This overlap ratio is instrumental in maintaining temporal continuity during sliding window inference.

Given the baseline nature of this test, further domain-specific optimization was omitted in favor of exploring the superior zero-shot capabilities of Video Foundation Models.

6.2 Results on TSM(MMAction2) model with IK EOS Dataset

This evaluation is compared, under the evaluation-protocol caveats discussed in Section 7.2, against the InternVideo2 results on the same IK EOS test split (Table 6.4).

The TSM (MMAction2) predictions on the IK EOS dataset are scored with a custom, recall-oriented evaluation script. Each event is associated with a single retained prediction: the pipeline keeps one segment per event (the detected segment nearest the trigger time when several survive the dynamic threshold), or a trigger-anchored fallback window when no segment fires. This single prediction is then scored against the annotated ground-truth interval $[t_{\text{start}}, t_{\text{end}}]$. All metrics are averaged over the $N = 16$ events of the Period-3 test split. This protocol differs from the one used for InternVideo2+CG-DETR, whose metrics are produced by the official CG-DETR evaluator; the consequences for cross-model comparison are discussed in Section 7.2.

- **Recall@1 at tIoU θ (R1@ θ).** As exactly one prediction is retained per event, the event is recalled at θ if the temporal IoU between this prediction and the ground truth is at least θ . R1@ θ is the fraction of recalled events

among the N events, matching the single-prediction (top-1) convention used for InternVideo2+CG-DETR.

- **Average Recall (AR).** The mean of $R1@θ$ over $θ ∈ \{0.1, 0.2, \dots, 0.9\}$. AR is a purely recall-based summary of localization tolerance and must *not* be confused with the precision-based mean Average Precision (mAP) reported for InternVideo2, which is computed by a different evaluator; the two are not directly comparable.
- **mIoU.** The mean, over the N events, of the temporal IoU between the retained prediction and the ground truth.
- **mTP (mean temporal precision).** The mean ratio of the prediction–ground-truth intersection to the *predicted* segment duration, indicating how tightly the predicted clip is bounded.
- **mTR (mean temporal recall).** The mean ratio of the same intersection to the *ground-truth* duration, indicating how much of each event is temporally covered.
- **Miss@ $θ$.** The number of events whose retained prediction falls below tIoU $θ$ (equivalently $N(1 - R1@θ)$). In earlier result tables this quantity was labelled the *false-positive count* (“FP-count”); strictly, however, it records under-localized or missed events rather than false positives, since under the single-prediction convention each prediction is matched to exactly one ground-truth event. It is therefore renamed Miss@ $θ$ here, and no false-positive (unmatched-prediction) count is defined.

The baseline yields a low mIoU of 0.1350 and an Average Recall (AR) of 13.19% over tIoU 0.1–0.9. Both the mean temporal recall (mTR, 31.25%) and the mean temporal precision (mTP, 13.50%) are low, indicating that the single peak-selected segment usually overlaps only a small fraction of the ground-truth event: 14 of the 16 events fall below tIoU = 0.5 and all 16 below tIoU = 0.7 (Miss@ $θ$). This is consistent with the selection design—the predicted segment is centred on the probability peak nearest the trigger and may be expanded up to a 5-second cap—so whenever that trigger-proximate peak is mislocated, the predicted window lands largely outside the short (2–3 s) scoring event. The zero-shot Kinetics-400 score, which has no basketball-specific notion of shot onset and offset, provides little signal to place the peak precisely.

6.3 Results on Internvideo2 Model with the APIDIS Dataset

In the early and middle phase of development, we used the public APIDIS dataset to set a baseline for the pipeline. The main goal here was simply to test the core system

Table 6.2 Performance of the off-the-shelf, zero-shot TSM (MMAAction2) baseline on the Period-3 test split of the IK EOS dataset (MR-full task, $N = 16$). Metrics are produced by a custom recall-oriented evaluator and are *not* directly comparable to the precision-based mAP of Table 6.4; see Section 7.2.

Metric	Value
R1@tIoU = 0.1	31.25%
R1@tIoU = 0.3	18.75%
R1@tIoU = 0.5	12.50%
R1@tIoU = 0.7	0.00%
AR (mean R1, tIoU 0.1–0.9)	13.19%
mIoU	0.1350
mTP	13.50%
mTR	31.25%
Miss@0.5	14 / 16
Miss@0.7	16 / 16
Evaluated events N	16

architecture and see how the models performed out-of-the-box, before introducing the company’s internal data.

Evaluation Setup. The evaluation on the APIDIS dataset was strictly direct, with no domain-specific training or fine-tuning applied to the models. To effectively extract sports highlights in a zero-shot setting, a dynamic thresholding strategy was implemented to handle variation in confidence scores across different video segments.

Let S_t denote the model’s confidence score at time step t , and let $\mathcal{S}$ represent the set of scores within a sliding temporal window. Instead of applying a rigid, globally fixed threshold, the dynamic threshold T for action segment activation is formulated as:

$$T = \mu_{\mathcal{S}} + k \cdot \sigma_{\mathcal{S}} \quad (6.5)$$

where $\mu_{\mathcal{S}}$ and $\sigma_{\mathcal{S}}$ represent the mean and standard deviation of the scores in $\mathcal{S}$, respectively, and k is a user-defined multiplier. A temporal frame at time t is selected as a highlight candidate only if its score satisfies $S_t > T$. The temporal filtering process is further governed by the sliding window’s *stride*.

Results and Analysis. Table 6.3 presents the top performance metrics obtained from the zero-shot evaluation across different parameter configurations, ranked by their F1-Score.

The empirical results confirm that the engineering infrastructure of the pipeline successfully ingests synchronized multi-angle video streams and generates structured highlight candidates. Under the optimal configuration ($k = -0.5$, $\text{stride} = 0.1$), the system achieves an F1-Score of 47.5% and a Recall@0.3 of 58.0%. This indi-

Table 6.3 Performance Metrics

Params (k,stride)	R@0.3	R@0.5	Precision	F1-Score	Preds
pred_k-0.3_stride0.5	60.0%	30.0%	35.8%	44.8%	81
pred_k0.3_stride0.1	50.0%	22.0%	38.6%	43.5%	83
pred_k-0.5_stride0.3	66.0%	36.0%	31.8%	42.9%	88
pred_k-0.3_stride0.1	70.0%	38.0%	29.8%	41.8%	94
pred_k0.7_stride0.1	38.0%	14.0%	38.6%	38.3%	70
pred_k-0.8_stride0.5	74.0%	42.0%	25.3%	37.7%	83

cates a reasonable baseline capacity for locating the approximate temporal windows of key actions without any prior training.

In summary, these early metrics provided a solid proof of concept for the pipeline framework, confirming that the code logic and multi-view synchronization function as intended. Nevertheless, the clear deficiency in precise boundary localization proved that zero-shot inference is insufficient for a production-ready system. This directly validated the necessity for domain-specific training and helped with the transition to the EOS dataset for the subsequent phases of development.

6.4 Results on Internvideo2 Model with IK EOS Dataset

In this section, we present the evaluation of the InternVideo2 model acting as the feature extractor within the CG-DETR framework for the TSGV task. We constructed a dedicated dataset built upon video recordings provided by Sony (see Section 3.2), comprising 55 valid scoring events from actual basketball games. The objective is to validate the feasibility of the proposed architecture and analyze its learning dynamics.

6.4.1 MR Task and Small-Scale Data Strategy

In this work, the core objective is formulated as a Temporal Moment Retrieval (MR) task. Given a continuous video sequence and a corresponding target action query as input, the system is required to perform precise temporal localization.

Unlike basic classification or ranking tasks, moment retrieval strictly demands that the model outputs exact temporal boundaries, specifically the start and end timestamps of the target event. This makes the task highly challenging, as it requires sub-second precision to accurately regress the exact frames where an action initiates and terminates, while effectively filtering out all irrelevant background content.

Due to the limited size of our dataset, which comprises a total of 55 segments across four periods (7, 16, 16, and 16 segments for Periods 1, 2, 3, and 4, respectively), we adopt a specific data splitting strategy. Specifically, the training set is formed by combining Period 1 with one of the other periods, while the remaining two periods are designated as the validation set and the test set, respectively. The

experimental results presented below are obtained under the configuration where Periods 1 and 4 serve as the training set, Period 2 as the validation set, and Period 3 as the test set.

6.4.2 Convergence Analysis and Learning Dynamics

To gain deeper insights into the optimization process and evaluate the training stability of our framework, we monitor the learning dynamics of the individual loss components defined in Section 4.5.6.

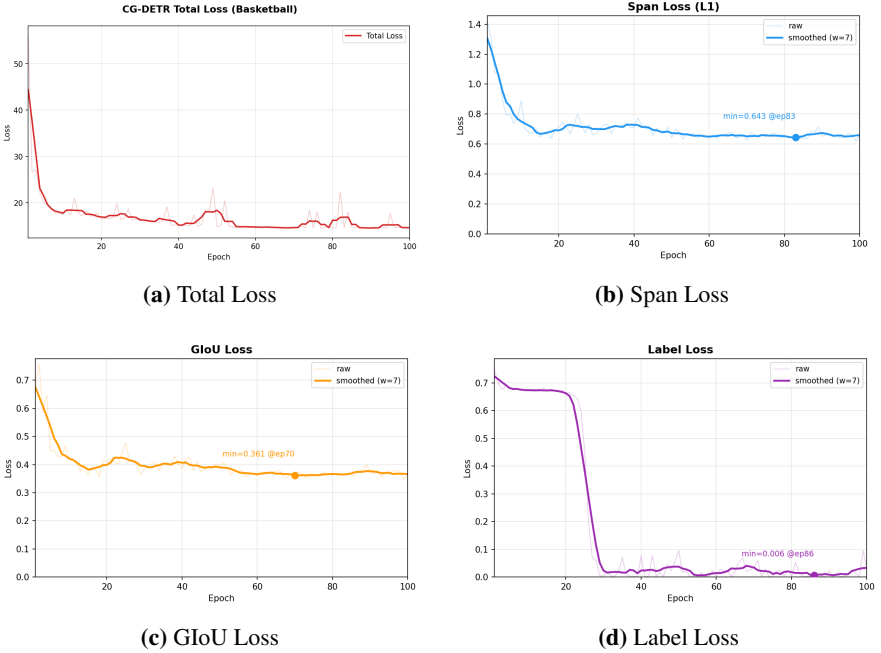


Figure 6.1 Training curves for all loss functions.

The total loss in Figure 6.1a decreases rapidly during the early training stage, dropping from approximately 45 to below 20 within the first 10 epochs. After around epoch 40, the curve gradually converges near 7 and remains highly stable with only minor fluctuations. This behavior suggests that the optimization process reaches a steady convergence state during the later training stage.

The span regression in Figure 6.1b loss exhibits a clear two-stage convergence pattern. During the first 15 epochs, the loss decreases rapidly from approximately 1.3 to 0.65, indicating that the model quickly learns coarse temporal boundary localization. However, after this stage, the curve enters a long plateau phase and only slowly decreases to approximately 0.64 over the remaining epochs. The early appearance of this plateau suggests that the model's temporal boundary regression

capability becomes saturated relatively early, and the remaining gradient signal for further localization refinement is limited.

A similar trend can also be observed in the GIoU loss in Figure 6.1c. The loss first decreases rapidly from approximately 0.68 to 0.38 within the first 15 epochs. After approximately 20 epochs, the curve enters a plateau region and even shows slight fluctuations, which is highly consistent with the stagnation observed in the span regression loss. Since both losses supervise temporal boundary quality, this phenomenon indicates strong coupling between the two optimization objectives. The GIoU loss resumes a slow downward trend and eventually converges near 0.361, indicating that the temporal overlap prediction has largely stabilized.

In contrast, the label classification loss in Figure 6.1d demonstrates a substantially different optimization behavior. During the first 20 epochs, the loss remains nearly unchanged around 0.72. Afterward, it undergoes a sharp decrease and drops below 0.1 within approximately 10 epochs, eventually converging near 0.006. One possible explanation is that the optimization process during early training is dominated by the heavily weighted span regression objective ($\lambda_{span} = 10$), causing the model to prioritize temporal localization learning. Once the localization-related losses become relatively saturated, the optimization gradually shifts toward the classification objective, resulting in rapid foreground-background discrimination improvement during the later training stage.

These curves indicate that training is essentially complete for the current data volume and model scale. We can now proceed to the testing phase with this model.

6.4.3 MR Task Results

We evaluate our method on the standard MR task in TSGV. The MR task aims to localize the temporal segment in an untrimmed video that best corresponds to a given textual query, requiring precise temporal boundary prediction under potentially long and complex video contexts.

Performance is assessed under multiple evaluation settings to capture both localization accuracy and robustness across different temporal tolerances. These include recall-based measures and intersection-based agreement with ground truth.

All subsequent results were obtained using a fixed clip length of 0.3 seconds. For the experimental dataset partition, Periods 1 and 4 were utilized as the training set, while Periods 2 and 3 were designated as the validation and test sets, respectively.

Following the evaluation framework established in Section 6.2, the primary metrics used to assess this task are defined as follows:

The model achieves **68.27% mIoU**, **73.43% mTP**, and **89.78% mTR** on the test set. This indicates that approximately 26.57% of the predicted temporal window lies outside the ground-truth segment, while 89.78% of the actual action duration is successfully covered by the prediction, leaving around 10.22% missed. These results suggest that the model is able to cover most ground-truth moments effectively, although some boundary-level misalignment still exists.

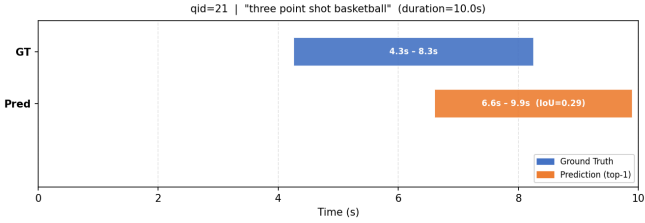


Figure 6.2 Grounding failure example of InternVideo2 on the IK EOS dataset (IoU = 0.29).

The quantitative results are shown in Table 6.4.

Table 6.4 Performance of the proposed framework regarding the MR-full task on the test set

Metric	Value
Recall1 @IoU=0.3	93.75%
Recall1 @IoU=0.5	87.50%
Recall1 @IoU=0.7	43.75%
mAP	50.62%
mAP@0.5	93.75%
mAP@0.75	34.38%
FP-count@0.5	2 out of 16
FP-count@0.7	9 out of 16
FP-total	16 out of 16

6.5 Failure Cases Analysis

As shown in Figure 6.2, the InternVideo2 model fails in this case on the IK EOS dataset, achieving an IoU of only 0.29. Based on the visualization in Figure 6.2, we analyze the main reasons for this low performance.

- **Small and Similar Training Data:** The training dataset is too small, and the videos are very similar because they all come from the same match. This makes it difficult for the model to learn general features.
- **Feature Loss of Distant Objects:** The event happens far from the camera. After frame extraction, resizing, and downsampling, these small targets lose their details, making them very hard for the model to recognize.

6.6 Best-cam Evaluation

The evaluation of the optimal camera selection module is conducted through a comparative analysis between human aesthetic preferences and the decision-making logic of the Multimodal Large Language Model. Given that the "best" viewing angle in sports broadcasting is inherently subjective and tied to cinematic narrative standards, a survey-based approach is employed to establish a reliable ground-truth baseline. This methodology follows established practices in the field of automated directing, such as the Smart Director system [65], which leverages human feedback to validate event-driven broadcasting decisions.

To comprehensively evaluate the directing intelligence and computational feasibility of utilizing MLLMs, a competitive benchmark is established to compare the primary system configuration, Qwen2.5-VL-7B, against three baseline vision-language architectures of varying scales and generations:

- **Qwen2.5-VL-7B (Primary Engine):** A 7.2-billion parameter multimodal model optimized for dynamic video understanding and fine-grained visual grounding, serving as the default directing framework in the proposed pipeline.
- **Qwen3-VL-2B (Baseline):** A lightweight next-generation model designed for resource-constrained, on-device execution, used to test the lower bound of computational costs.
- **Qwen3-VL-8B (Baseline):** A heavier next-generation model optimized for complex spatial reasoning and dynamic coordinate prediction.
- **Llama-3.2-Vision-11B (Baseline):** An 11-billion parameter foundation model renowned for general-domain visual reasoning and open-vocabulary understanding.

The models are benchmarked across 11 representative basketball scoring events extracted from the synchronized dataset. This specific subset of 11 events was selected based on two primary methodological considerations. First, the chosen sequences adequately capture the diverse spatial-temporal patterns of the match (e.g., fast breaks, set plays, and perimeter shooting), while omitting highly repetitive scenarios that exhibit identical camera angles and player distributions. Second, restricting the evaluation to these 11 representative events ensures the feasibility of the subsequent user preference survey by preventing cognitive fatigue among the expert participants, thereby maintaining the quality and focus of the subjective directing assessments. The evaluation is systematically divided into two critical dimensions: quantitative execution latency (computational efficiency) and cross-model camera selection alignment (directing logic consensus).

6.6.1 Quantitative Inference Latency and Hardware Feasibility

In live sports production, system latency constitutes the primary engineering bottleneck. For a highlight generation pipeline to be commercially viable, edited clips must be delivered shortly after the occurrence of a physical play. Thus, evaluating the inference footprint of the vision-language decision layer on workstation-class hardware is paramount for local deployment feasibility.

Table 6.5 details the precise execution latency (measured in seconds) required by each MLLM to perform a centralized joint evaluation pass across the multi-view matrix for each event. The benchmarking was conducted locally on the same machine used for TSM model(detailed specifications are provided in Section 5.3.3).

Table 6.5 Comparative Inference Latency (Seconds) across Different MLLM Backbones

Event ID	Qwen2.5-VL-7B	Qwen3-VL-2B	Qwen3-VL-8B	Llama-3.2-Vision-11B
13312981	19.81	31.35	354.60	76.50
13316627	20.10	37.67	242.54	59.82
13316817	20.99	82.43	187.97	64.37
13318238	19.97	27.70	415.23	61.72
13318658	20.11	34.70	210.93	53.88
13314211	19.99	35.74	360.35	59.60
13300002	19.80	82.04	251.57	61.69
13315317	19.64	58.74	427.47	68.68
13300004	19.70	73.29	254.37	63.19
13317921	19.83	45.26	223.30	54.26
13318533	20.11	57.34	290.55	48.73
Mean	20.01	54.21	292.63	61.13
Std. Dev.	0.35	20.73	81.04	7.57

The empirical findings presented in Table 6.5 yield several critical insights regarding pipeline integration:

- **Timing Predictability of Qwen2.5-VL-7B:** The Qwen2.5-VL-7B architecture displays exceptional temporal stability. It operates with a highly predictable mean inference latency of 20.01 seconds and a negligible standard deviation of $\sigma \approx 0.35$ s. This deterministic behavior is highly desirable for scheduling live streaming workflows.
- **Inherent Latency Fluctuation of Qwen3-VL-2B:** Despite its smaller parameter footprint, Qwen3-VL-2B demands an average of 54.21 seconds per execution pass. Furthermore, it suffers from severe timing jitter ($\sigma \approx 20.73$ s), with individual runs ranging unpredictably from a swift 27.70 s to a prolonged 82.43 s.
- **Prohibitive Resource Penalties of Heavy Models:** The heavier models, Qwen3-VL-8B and Llama-3.2-Vision-11B, incur dramatic computational

overheads on the local workstation hardware environment. Qwen3-VL-8B exhibits a mean latency of 292.63 seconds (≈ 4.88 minutes), representing an unacceptable systemic lag for post-event distribution. Llama-3.2-Vision-11B stabilizes at 61.13 seconds, which, while significantly faster than the 8B model, remains over three times slower than Qwen2.5-VL-7B.

Consequently, while next-generation and larger architectures offer strong theoretical benefits, the **Qwen2.5-VL-7B** model serves as the primary operational directing engine. It successfully circumvents the critical latency bottlenecks inherent to heavier multi-stream reasoning frameworks, fitting optimally within the hardware-constrained “low-cost” paradigm.

6.6.2 Directing Alignment Analysis and User Survey Evaluation

Evaluating the quality of optimal camera selection is non-trivial, as sports directing is inherently a subjective aesthetic endeavor. To establish a reliable ground truth for evaluating directing logic, a systematic user preference survey was conducted with a panel of 15 participants from the Computer Science field. Eleven representative basketball play events were evaluated to identify the optimal viewing perspective (Human Best View) for each sequence.

Table 6.6 maps the specific camera view selected by each model alongside the human preferences. For clarity, the eight available camera perspectives are indexed as A through H, representing the distinct spatial coverage of the deployed multi-camera array.

Table 6.6 Comparison of Optimal Camera Selection and Alignment with Human Preference

Event ID	Qwen2.5-VL-7B	Qwen3-VL-2B	Qwen3-VL-8B	Llama-3.2-Vision-11B	Human Preference
13312981	Event F	Event E	Event E	Event F	Event F
13316627	Event G	Event E	Event F	Event F	Event F
13316817	Event F	Event E	Event F	Event F	Event F
13318238	Event G	Event E	Event E	Event H	Event G
13318658	Event F	Event E	Event G	Event G	Event F
13314211	Event B	Event C	Event A	Event A	Event B
13300002	Event C	Event A	Event C	Event A	Event C
13315317	Event C	Event A	Event A	Event C	Event C
13300004	Event C	Event A	Event A	Event C	Event C
13317921	Event C	Event A	Event D	Event C	Event B
13318533	Event C	Event A	Event A	Event D	Event C
Accuracy	81.82%	0.00%	27.27%	45.45%	100.00%

The experimental findings mapped in Table 6.6 yield profound insights into the spatial-semantic alignment capabilities of MLLMs:

1. **State-of-the-Art Aesthetic Alignment by Qwen2.5-VL-7B:** The Qwen2.5-VL-7B model demonstrates exceptional fidelity to human directing prefer-

ences, achieving a remarkable alignment accuracy of **81.82%**. This robust alignment demonstrates that its architectural priors successfully replicate high-level spatial heuristics, including optimal ball-to-player proximity, occlusion minimization, and focus on the primary physical action.

2. **Performance Degradation in Alternative Backbones:** Despite its larger parameter size, Llama-3.2-Vision-11B matches human selections in only **45.45%** of cases, often failing to track scoring zones during rapid spatial transitions. Qwen3-VL-8B displays even lower alignment (**27.27%**), while the lightweight Qwen3-VL-2B collapses to **0.00%** accuracy, exhibiting systemic spatial bias and modal collapse.
3. **Discrepancy and Edge Cases:** Event 13317921 presents a notable edge case where all evaluated models diverged from the human gold standard (B). Both Qwen2.5-VL-7B and Llama-3.2-Vision-11B selected camera C, indicating ambiguous visual scenarios where competing directing trade-offs exist between defensive setup and wide-angle transition views.

In summary, the empirical results derived from the user survey and cross-model comparison provide robust, quantitative validation for the architectural design of the proposed pipeline. The selected **Qwen2.5-VL-7B** backbone not only operates within the target low-latency profile for real-time edge workstation deployment but also exhibits unparalleled semantic capability in replicating professional-grade sports broadcasting aesthetic decisions.

7

Discussion

This chapter evaluates the empirical findings, interprets their broader implications for autonomous sports broadcasting, analyzes systemic trade-offs, and outlines limitations alongside future research directions.

7.1 Thematic Synthesis of Research Findings

The evaluation of the proposed system is structured around the three core research questions formulated at the inception of this study, mapping empirical data directly to these programmatic themes.

7.1.1 Socio-Technical Feasibility and Pipeline Integration

In addressing the first research question (**RQ1**) regarding the feasibility and integration of an end-to-end autonomous sports production pipeline, the empirical results provide strong validation. The successful coordination of scoreboard trigger extraction, active half-court filtering, temporal action grounding, Vision-Language Model (VLM) directing, and automatic post-production editing demonstrates that the workflow can be fully automated.

The feasibility of this multi-stage orchestration is supported by two key milestones:

1. **Precise Action Localization:** The InternVideo2 + CG-DETR framework achieved a Recall@IoU=0.5 of 87.50% and a mean Temporal Recall of 89.78% on the IK EOS dataset, ensuring that highlight clips contain complete action sequences.
2. **Autonomous View Selection:** The Qwen2.5-VL-7B model mapped raw multi-stream inputs to structured directing choices, matching human expert decisions with 81.82% accuracy.

Furthermore, the implementation of a global timestamp synchronization mechanism across all eight cameras successfully eliminated frame-drift, verifying that a

decentralized software framework can operate as an autonomous director without chronological discrepancy.

7.1.2 Computational Efficiency and Hierarchical Reductions

To address RQ2 concerning the optimisation of computational efficiency against the semantic demands of Vision-Language Models, a hierarchical data-reduction strategy was designed and integrated into the pipeline. The strategy operates through two sequential gates applied before any heavy multimodal inference is invoked.

Temporal pruning. Rather than performing dense feature extraction over the entire broadcast stream, the pipeline remains dormant until a scoreboard change is detected, at which point inference is restricted to a fixed 10-second window around the trigger timestamp. On the IK EOS dataset, which contains 55 annotated scoring events (43 two-point and 12 three-point field goals) distributed across a full four-period game of approximately 2400 seconds of active footage, this constraint limits active processing to $55 \times 10 = 550$ seconds, yielding a temporal coverage ratio of roughly 22.9%. The remaining 77.1% of the broadcast stream bypasses all downstream feature extraction and VLM inference entirely.

Spatial pruning. At each trigger point, a lightweight YOLOv8 detector identifies the active half-court from player spatial distribution, immediately excluding the four cameras covering the inactive half. This reduces the number of camera views passed to the VLM from eight to four, directly halving the multimodal input size of each joint inference pass.

Observed latency outcome. Together, these two gates ensure that Qwen2.5-VL-7B operates on a single, pre-filtered four-camera frame matrix rather than on continuous eight-stream footage. The resulting mean inference latency of 20.01 seconds ($\sigma \approx 0.35$ s, Table 6.5) is both low and highly predictable, a property that is critical for scheduling post-event clip distribution in live production workflows.

It should be noted that a direct ablation study comparing pipeline throughput with and without these reduction gates was not conducted, as the full-stream dense-inference configuration was infeasible on the available single-GPU workstation. The cost reduction figures reported above are therefore derived from analytical estimates based on dataset statistics rather than controlled measurements, and this constitutes a limitation of the current evaluation. Nevertheless, the stable latency profile of the chosen backbone, combined with the structural design of the reduction gates, provides a principled basis for the low-cost claim and motivates future benchmarking under full-stream conditions on multi-GPU infrastructure.

7.1.3 Aesthetic Human-VLM Alignment and Directing Logic

Regarding the third research question (RQ3) on the alignment of VLM-driven decisions with professional broadcasting standards, the user preference survey and consensus analysis yield encouraging insights. Under a zero-shot configuration, the

quantized Qwen2.5-VL-7B model matched the consensus choices of human preference in 9 out of 11 play sequences (81.82% accuracy, Table 6.6).

This high level of alignment indicates that advanced VLMs possess robust visual-semantic priors. Qwen2.5-VL-7B prioritized camera angles based on semantic importance—such as player-to-ball proximity, minimization of visual occlusions from referees or defensive blocks, and preservation of global play context (capturing both the shooter and the hoop). This performance significantly outperformed alternative backbones, including Llama-3.2-Vision-11B (45.45% alignment) and Qwen3-VL-8B (27.27% alignment), demonstrating the viability of using advanced vision-language architectures for autonomous, aesthetic-aware sports directing.

7.2 Comparative Analysis: Discriminative vs. Generative Paradigms

A consistent finding across our experiments is the large gap between the off-the-shelf, zero-shot TSM (MMAAction2) baseline and the trained InternVideo2+CG-DETR pipeline on the IK EOS test split. Using comparably defined localization metrics, the TSM baseline reaches a Recall1@IoU= 0.5 of 12.50% and an mIoU of 0.1350, whereas InternVideo2+CG-DETR attains 87.50% and 0.6827, respectively.

Two caveats qualify this comparison. First, the two systems are scored by different evaluation toolchains: TSM with a custom recall-oriented script and InternVideo2 with the official CG-DETR evaluator. We therefore base the comparison on R1@ θ and mIoU, whose definitions align across both toolchains, and we do *not* compare the recall-based AR of Table 6.2 with the precision-based mAP of Table 6.4. Second, the comparison contrasts an untrained, zero-shot discriminative detector with a foundation-model pipeline whose CG-DETR head is trained on the in-domain Sony data. The observed gap therefore reflects the combined effect of in-domain training, the richer multimodal representation of InternVideo2, and the trigger-window fallback of the TSM baseline, rather than an intrinsic limit of discriminative architectures alone.

7.2.1 Rationale for Retaining TSM as an Unoptimized Baseline

A deliberate design decision in this work is to employ the TSM (MMAAction2) detector strictly as an off-the-shelf, zero-shot baseline, without any basketball-specific fine-tuning or dedicated temporal head. Three considerations motivate this choice.

First, the two detectors occupy fundamentally different positions in the design space. TSM is a discriminative action classifier whose temporal modeling is repurposed here for localization, whereas InternVideo2 is a large-scale multimodal foundation model whose video-text alignment is intrinsically suited to open-vocabulary temporal grounding. As the target system is oriented toward practical deployment and high-level semantic understanding, development effort was concentrated on the architecture with the higher performance ceiling for cross-modal reasoning, rather

than on extending a classifier whose design objective does not match the grounding task.

Second, optimizing the discriminative baseline to a competitive level—fine-tuning a basketball-specific classifier, attaching a trainable temporal regression head, and re-running hyperparameter search on in-domain data—would require additional annotated data, training infrastructure, and engineering time that lie outside the scope of this project. TSM is therefore retained solely as a reference lower bound that characterizes the performance floor of traditional discriminative models in this multi-view setting.

Third, the empirical gap observed under in-domain evaluation reinforced this decision. On the same IK EOS test split, the InternVideo2+CG-DETR pipeline substantially exceeds the TSM baseline across the localization metrics common to both evaluators (cf. Tables 6.2 and 6.4). Even acknowledging that part of this gap stems from the training asymmetry between the two systems rather than from architecture alone, the magnitude of the difference indicated that the limited project resources were more productively invested in the foundation-model branch than in tuning the baseline.

7.3 Best-Cam Selection and Pragmatic Edge-Computing Viability

The evaluation of the best-camera selection module (Section 6.5) highlights a critical trade-off between model parameters, inference latency, and directing accuracy. While pursuing the highest parameter scale yields marginal accuracy gains in theoretical benchmarks, live sports production demands a **low-cost, practical development paradigm** deployable on accessible, standard hardware:

- **Hardware Agnosticism and Local Deployment:** When quantized to INT4, Qwen2.5-VL-7B runs smoothly on local workstations or edge PCs with mid-tier consumer GPUs (8GB to 16GB VRAM), requiring no expensive cloud API subscriptions or high-end GPU clusters.
- **Deterministic Low Latency:** Qwen2.5-VL-7B demonstrated highly predictable execution times, with a mean latency of 20.01 seconds and an extremely low standard deviation ($\sigma \approx 0.35$ s). In contrast, the next-generation Qwen3-VL-2B model suffered from severe timing jitter ($\sigma \approx 20.73$ s) and took nearly three times longer (54.21 s) on average. Heavier architectures like Qwen3-VL-8B (292.63 s) and Llama-3.2-Vision-11B (61.13 s) introduced unacceptable delays for near-real-time highlight distribution.
- **Resistance to Modal Collapse:** Smaller models like Qwen3-VL-2B displayed a complete lack of spatial awareness, repeatedly choosing the same camera MAC address across distinct events (modal collapse, Table 6.6).

Qwen2.5-VL-7B maintained high spatial-semantic sensitivity, achieving an 81.82% match with human experts.

Furthermore, due to the decoupled, modular software architecture of the pipeline, the system remains highly future-proof. Future researchers can easily swap the directing core with newer, more advanced VLMs as they emerge, without refactoring the underlying multi-camera infrastructure.

7.4 Systemic Limitations and Threats to Validity

Despite the strong performance of the automated pipeline, several systemic limitations must be acknowledged:

1. **Data Constraint and Generalization:** The primary evaluation was conducted on the IK EOS dataset, which contains 55 annotated scoring events from a single basketball game. A larger and more diverse multi-game dataset is required to confirm the system’s generalization across different lighting conditions, court layouts, and team jerseys.
2. **Heuristic Pipeline Dependencies:** The spatial pruning logic relies heavily on the YOLOv8-based active half-court detector. In plays involving extremely fast transitions or players distributed evenly across both sides of the court, this simple heuristic side gate may fail, potentially filtering out the camera views containing the actual highlight action.
3. **Offline Feature Extraction:** To manage GPU memory overhead on the local workstation, InternVideo2 features were extracted and cached offline before training. This decoupling prevents end-to-end backpropagation throughout the entire pipeline, limiting the potential benefits of joint feature-and-grounding optimization.
4. **Visual Ambiguity in Fast Plays:** In scenarios with rapid passing, complex off-ball screens, or dense crowding in the paint, the compressed, single-frame representation provided to the VLM can occasionally be ambiguous, leading to suboptimal camera selection.

7.5 Future Research Directions

Building upon the successful prototype and addressing the identified limitations, several pathways are identified for future research and commercial productization:

7.5.1 Live Ingestion and Real-time Streaming

Transitioning the system to a live broadcast environment is a natural next step. This would involve integrating live RTMP/RTSP stream ingestion, performing on-the-fly scoreboard OCR change detection, and feeding buffered video chunks into the grounding and directing modules. Optimizing inference code to run within sub-second latencies would enable near-instantaneous live highlight generation during commercial breaks or active game delays.

7.5.2 Hardware Integration and Standalone Application

For real-world deployment, the software pipeline should be packaged into a standalone desktop or cloud application. Integrating the directing outputs directly with professional production hardware (e.g., ATEM video switchers, Sony PTZ cameras, or software-based suites like vMix) would allow the system to physically trigger camera cuts in real-time. This integration would enable a single operator to manage multi-camera setups with minimal manual effort.

7.5.3 UI Evolution and Interactive Human-in-the-Loop Editing

The visualization interface developed in this project has significant potential for further evolution into a fully featured cloud-based video editor. By adding multi-track editing capabilities, manual override controls, and active feedback loops, human directors could easily correct or refine automated selections.

Furthermore, integrating a feedback button on the UI would allow users to flag suboptimal camera angles or incorrect temporal boundaries. This human-labeled data could be collected to build a Reinforcement Learning from Human Feedback (RLHF) pipeline, continuously tuning the VLM's directing logic and alignment with professional aesthetic standards.

8

Conclusion

This thesis investigates the design, implementation, and evaluation of an automated, end-to-end sports production pipeline driven by multimodal vision-language architectures. The research successfully demonstrates a functional 0-to-1 software pipeline capable of ingesting and syncing 8 simultaneous camera feeds, automatically localizing basketball shooting highlights, and selecting the optimal camera angle for broadcast-quality clips.

To address the high computational costs of processing multiple high-resolution video streams, a hierarchical data-reduction strategy was introduced. This approach utilizes temporal triggers from scoreboard changes and spatial heuristics from YOLOv8 player distributions to prune temporal and spatial redundancies. Consequently, heavy vision-language reasoning is only applied to relevant, pre-filtered segments, significantly reducing system latency.

The quantitative and qualitative evaluations validated the effectiveness of the proposed framework. The integration of InternVideo2 with CG-DETR achieved a Recall@IoU=0.5 of 87.50% and an mIoU of 68.27% on the IK EOS dataset, significantly outperforming the off-the-shelf, zero-shot TSM baseline. For camera selection, the quantized Qwen2.5-VL-7B model demonstrated high alignment with human directing experts, achieving an 81.82% match rate with a predictable latency of 20.01 seconds. These results demonstrate that combining lightweight sports-temporal filters with deep multimodal foundation models can achieve professional-grade, autonomous sports directing while remaining computationally and financially affordable.

Additionally, a full-stack, decoupled visualization interface was designed and implemented. This "control tower" dashboard provides real-time telemetry, synchronized multi-view playback, and an interactive SVG timeline, significantly improving the observability and human-in-the-loop verification of the pipeline. This work establishes a solid foundation for autonomous sports broadcasting, demonstrating that modern generative AI can be practically deployed on accessible, edge-computing hardware.

9

Use of AI Tools

Generative AI tools were used only in a supporting role. ChatGPT (OpenAI) assisted with language polishing, English translation, and the visual design of the system overview figure (Figure 1.1); Claude Code (Anthropic) assisted with writing and debugging code.

Bibliography

- [1] Sony Group Corporation. *Sony Group Portal - Home*. <https://www.sony.com/en/>. Accessed: 2026-06-01. 2026.
- [2] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem. “Soccernet: a scalable dataset for action spotting in soccer videos”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 1711–1721.
- [3] H.-C. Shih. “A survey of content-aware video analysis for sports”. *IEEE Transactions on circuits and systems for video technology* **28**:5 (2017), pp. 1212–1231.
- [4] A. Deliege, A. Cioppa, S. Giancola, M. J. Seikavandi, J. V. Dueholm, K. Nasrollahi, B. Ghanem, T. B. Moeslund, and M. Van Droogenbroeck. “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*. 2021, pp. 4508–4519.
- [5] A. Cioppa, A. Deliege, S. Giancola, B. Ghanem, M. V. Droogenbroeck, R. Gade, and T. B. Moeslund. “A context-aware loss function for action spotting in soccer videos”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 13126–13136.
- [6] B. Vanherle, T. Vervoort, N. Michiels, and P. Bekaert. “Automatic camera control and directing with an ultra-high-definition collaborative recording system”. In: *Proceedings of the 18th ACM SIGGRAPH European Conference on Visual Media Production*. 2021, pp. 1–10.
- [7] S. Giancola, A. Cioppa, A. Deliege, F. Magera, V. Somers, L. Kang, X. Zhou, O. Barnich, C. De Vleeschouwer, A. Alahi, et al. “Soccernet 2022 challenges results”. In: *Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports*. 2022, pp. 75–86.
- [8] A. Cioppa, S. Giancola, V. Somers, F. Magera, X. Zhou, H. Mkhallati, A. Deliege, J. Held, C. Hinojosa, A. M. Mansourian, et al. “Soccernet 2023 challenges results”. *Sports Engineering* **27**:2 (2024), p. 24.

- [9] K. Chen, V. Srivastav, D. Mutter, and N. Padoy. “Learning from synchronization: self-supervised uncalibrated multi-view person association in challenging scenes”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 24419–24428.
- [10] F. Yang, C. Jung, R. Kawamura, and H. Y. Wong. “Multi-camera self-calibration in sports motion capture: leveraging human and stick poses”. *arXiv preprint arXiv:2604.17567* (2026).
- [11] R. Decorte, J. De Bock, J. Taelman, M. Slembrouck, and S. Verstockt. “Fully automatic camera for personalized highlight generation in sporting events”. *Sensors* **24**:3 (2024), p. 736.
- [12] S. Ren, L. Yao, S. Li, X. Sun, and L. Hou. “Timechat: a time-sensitive multi-modal large language model for long video understanding”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 14313–14323.
- [13] J. Lin, C. Gan, and S. Han. “Tsm: temporal shift module for efficient video understanding”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 7083–7093.
- [14] Y. Wang, K. Li, X. Li, J. Yu, Y. He, G. Chen, B. Pei, R. Zheng, Z. Wang, Y. Shi, et al. “Internvideo2: scaling foundation models for multimodal video understanding”. In: *European conference on computer vision*. Springer. 2024, pp. 396–416.
- [15] H. Zhang, X. Li, and L. Bing. *Video-llama: an instruction-tuned audio-visual language model for video understanding*. 2023. arXiv: 2306.02858 [cs.CL]. URL: <https://arxiv.org/abs/2306.02858>.
- [16] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao. “Videochat: chat-centric video understanding”. *Science China Information Sciences* **68**:10 (2025), p. 200102.
- [17] P. Xu, X. Zhu, and D. A. Clifton. “Multimodal learning with transformers: a survey”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**:10 (2023), pp. 12113–12132.
- [18] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen, et al. “Parameter-efficient fine-tuning of large-scale pre-trained language models”. *Nature machine intelligence* **5**:3 (2023), pp. 220–235.
- [19] B. Korbar, D. Tran, and L. Torresani. “Scsampler: sampling salient clips from video for efficient action recognition”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6232–6242.
- [20] L. Chen, M. Zaharia, and J. Zou. “Frugalgpt: how to use large language models while reducing cost and improving performance”. *arXiv preprint arXiv:2305.05176* (2023).

- [21] S. Majumder, T. Nagarajan, Z. Al-Halah, R. Pradhan, and K. Grauman. “Which viewpoint shows it best? language for weakly supervising view selection in multi-view instructional videos”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 29016–29028.
- [22] J. Xu, Y. Huang, J. Cheng, Y. Yang, J. Xu, Y. Wang, W. Duan, S. Yang, Q. Jin, S. Li, et al. “Visionreward: fine-grained multi-dimensional human preference learning for image and video generation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 40. 13. 2026, pp. 11269–11277.
- [23] J. Liu, G. Liu, J. Liang, Z. Yuan, X. Liu, M. Zheng, X. Wu, Q. Wang, M. Xia, X. Wang, et al. “Improving video generation with human feedback”. *Advances in Neural Information Processing Systems* **38** (2026), pp. 82155–82192.
- [24] A. Scott, I. Uchida, N. Ding, R. Umemoto, R. Bunker, R. Kobayashi, T. Koyama, M. Onishi, Y. Kameda, and K. Fujii. “Teamtrack: a dataset for multi-sport multi-object tracking in full-pitch videos”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 3357–3366.
- [25] M. Gutiérrez-Pérez and A. Agudo. “Soccernet-v3d: leveraging sports broadcast replays for 3d scene understanding”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 5978–5987.
- [26] G. Jocher, A. Chaurasia, and J. Qiu. *Ultralytics YOLOv8*. Version 8.0.0. 2023. URL: <https://github.com/ultralytics/ultralytics>.
- [27] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren. “Distance-iou loss: faster and better learning for bounding box regression”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 07. 2020, pp. 12993–13000.
- [28] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang. “Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection”. *Advances in neural information processing systems* **33** (2020), pp. 21002–21012.
- [29] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. *The kinetics human action video dataset*. 2017. arXiv: 1705.06950 [cs.CV]. URL: <https://arxiv.org/abs/1705.06950>.
- [30] I. Laptev. “On space-time interest points”. *International Journal of Computer Vision* **64**:2 (2005), pp. 107–123. DOI: 10.1007/s11263-005-1838-7. URL: <https://doi.org/10.1007/s11263-005-1838-7>.
- [31] H. Wang and C. Schmid. “Action recognition with improved trajectories”. In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 3551–3558. DOI: 10.1109/ICCV.2013.441.

- [32] M. A. R. Ahad, T. Jie, H. Kim, and S. Ishikawa. “Motion history image: its variants and applications”. *Machine Vision and Applications* **23** (2010), pp. 255–281. DOI: 10.1007/s00138-010-0298-4.
- [33] L. Xia, C.-C. Chen, and J. K. Aggarwal. “View invariant human action recognition using histograms of 3d joints”. *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2012), pp. 20–27. URL: <https://api.semanticscholar.org/CorpusID:14239485>.
- [34] H. Liu, R. Feris, V. Krueger, and M.-T. Sun. “Unsupervised action classification using space-time link analysis”. In: vol. 2010. 2010, pp. 3437–3440. DOI: 10.1155/2010/626324.
- [35] H.-C. Shih. “A survey of content-aware video analysis for sports”. *IEEE Transactions on Circuits and Systems for Video Technology* **28**:5 (2018), pp. 1212–1231. ISSN: 1558-2205. DOI: 10.1109/tcsvt.2017.2655624. URL: <http://dx.doi.org/10.1109/TCSVT.2017.2655624>.
- [36] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. *Cdc: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos*. 2017. arXiv: 1703.01515 [cs.CV]. URL: <https://arxiv.org/abs/1703.01515>.
- [37] Z. Shou, D. Wang, and S.-F. Chang. *Temporal action localization in untrimmed videos via multi-stage cnns*. 2016. arXiv: 1601.02129 [cs.CV]. URL: <https://arxiv.org/abs/1601.02129>.
- [38] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia. *Turn tap: temporal unit regression network for temporal action proposals*. 2017. arXiv: 1703.06189 [cs.CV]. URL: <https://arxiv.org/abs/1703.06189>.
- [39] H. Xu, A. Das, and K. Saenko. *R-c3d: region convolutional 3d network for temporal activity detection*. 2017. arXiv: 1703.07814 [cs.CV]. URL: <https://arxiv.org/abs/1703.07814>.
- [40] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang. *Bsn: boundary sensitive network for temporal action proposal generation*. 2018. arXiv: 1806.02964 [cs.CV]. URL: <https://arxiv.org/abs/1806.02964>.
- [41] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen. *Bmn: boundary-matching network for temporal action proposal generation*. 2019. arXiv: 1907.09702 [cs.CV]. URL: <https://arxiv.org/abs/1907.09702>.
- [42] Z. Qing, H. Su, W. Gan, D. Wang, W. Wu, X. Wang, Y. Qiao, J. Yan, C. Gao, and N. Sang. *Temporal context aggregation network for temporal action proposal refinement*. 2021. arXiv: 2103.13141 [cs.CV]. URL: <https://arxiv.org/abs/2103.13141>.
- [43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention is all you need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: <https://arxiv.org/abs/1706.03762>.

- [44] G. Bertasius, H. Wang, and L. Torresani. *Is space-time attention all you need for video understanding?* 2021. arXiv: 2102.05095 [cs.CV]. URL: <https://arxiv.org/abs/2102.05095>.
- [45] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. “Video swin transformer”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 3202–3211.
- [46] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. *End-to-end object detection with transformers*. 2020. arXiv: 2005.12872 [cs.CV]. URL: <https://arxiv.org/abs/2005.12872>.
- [47] X. Liu, Q. Wang, Y. Hu, X. Tang, S. Zhang, S. Bai, and X. Bai. “End-to-end temporal action detection with transformer”. *IEEE Transactions on Image Processing* **PP** (2022), pp. 1–1. DOI: 10.1109/TIP.2022.3195321.
- [48] D. Shi, Y. Zhong, Q. Cao, J. Zhang, L. Ma, J. Li, and D. Tao. *React: temporal action detection with relational queries*. 2022. arXiv: 2207.07097 [cs.CV]. URL: <https://arxiv.org/abs/2207.07097>.
- [49] X. Lan, Y. Yuan, X. Wang, Z. Wang, and W. Zhu. *A survey on temporal sentence grounding in videos*. 2021. arXiv: 2109.08039 [cs.CV]. URL: <https://arxiv.org/abs/2109.08039>.
- [50] H. Zhang, A. Sun, W. Jing, and J. T. Zhou. “Temporal sentence grounding in videos: a survey and future directions”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**:8 (2023), pp. 10443–10465. ISSN: 1939-3539. DOI: 10.1109/tpami.2023.3258628. URL: <http://dx.doi.org/10.1109/TPAMI.2023.3258628>.
- [51] M. Zheng, X. Cai, Q. Chen, Y. Peng, and Y. Liu. *Training-free video temporal grounding using large-scale pre-trained models*. 2024. arXiv: 2408.16219 [cs.CV]. URL: <https://arxiv.org/abs/2408.16219>.
- [52] Z. Tong, Y. Song, J. Wang, and L. Wang. *Videomae: masked autoencoders are data-efficient learners for self-supervised video pre-training*. 2022. arXiv: 2203.12602 [cs.CV]. URL: <https://arxiv.org/abs/2203.12602>.
- [53] Y. Wang, K. Li, Y. Li, Y. He, B. Huang, Z. Zhao, H. Zhang, J. Xu, Y. Liu, Z. Wang, S. Xing, G. Chen, J. Pan, J. Yu, Y. Wang, L. Wang, and Y. Qiao. *Internvideo: general video foundation models via generative and discriminative learning*. 2022. arXiv: 2212.03191 [cs.CV]. URL: <https://arxiv.org/abs/2212.03191>.
- [54] W. Moon, S. Hyun, S. Lee, and J.-P. Heo. *Correlation-guided query-dependency calibration for video temporal grounding*. 2024. arXiv: 2311.08835 [cs.CV]. URL: <https://arxiv.org/abs/2311.08835>.

- [55] J. Xu, G. Zhao, S. Yin, W. Zhou, and Y. Peng. “Finesports: a multi-person hierarchical sports video dataset for fine-grained action understanding”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 21773–21782.
- [56] N. Siddiqui, P. Tirupattur, and M. Shah. “Dvanet: disentangling view and action features for multi-view action recognition”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 5. 2024, pp. 4873–4881.
- [57] T. T. Nguyen, Y. Kawanishi, V. John, T. Komamizu, and I. Ide. “Multitsf: transformer-based sensor fusion for human-centric multi-view and multi-modal action recognition”. *arXiv preprint arXiv:2504.02279* (2025).
- [58] J. Zhang, Y. Ji, F. Ren, Y. Li, B. Zeng, Z. Chen, K. Chen, L. Shou, G. Chen, and H. Li. “Efficient inference for large vision-language models: bottlenecks, techniques, and prospects”. *arXiv preprint arXiv:2604.05546* (2026).
- [59] Y. Chen, B. Shan, X. Ye, and C. Chen. “Evoprune: early-stage visual token pruning for efficient mllms”. *arXiv preprint arXiv:2603.03681* (2026).
- [60] C. De Vleeschouwer, F. Chen, D. Delannay, C. Parisot, C. Chaudy, E. Martrou, A. Cavallaro, et al. “Distributed video acquisition and annotation for sport-event summarization”. *NEM summit* **8**:10.1016 (2008).
- [61] S. Jung, H. Kim, H. Park, and A. Choi. “Integrated ai system for real-time sports broadcasting: player behavior, game event recognition, and generative ai commentary in basketball games”. *Applied Sciences* **15**:3 (2025), p. 1543.
- [62] M. Contributors. *Openmmlab’s next generation video understanding toolbox and benchmark*. <https://github.com/open-mmlab/mmaction2>. 2020.
- [63] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. “Generalized intersection over union: a metric and a loss for bounding box regression”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 658–666.
- [64] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, et al. “Qwen3-vl technical report”. *arXiv preprint arXiv:2511.21631* (2025).
- [65] Y. Pan, Y. Chen, Q. Bao, N. Zhang, T. Yao, J. Liu, and T. Mei. “Smart director: an event-driven directing system for live broadcasting”. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **17**:4 (2021), pp. 1–18.

Master's Theses in Mathematical Sciences 2026:E31
ISSN 1404-6342

LUTFMA-3617-2026

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

<http://www.maths.lth.se/>