TOWARDS COMPREHENSIVE BASKETBALL UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding basketball requires recognizing events, identifying participants, localizing actions, and connecting visual evidence to structured knowledge. Existing sports evaluations cover many of these capabilities but provide limited insight into failures in linking them. We introduce **BasketballBench**, a multimodal benchmark comprising 7,600 questions across ten tasks, built from 2,501 possession-level broadcast clips and structured records from the 2025–2026 NBA season. Its tasks evaluate individual capabilities and their integration through event–participant associations, spatial and temporal grounding, and identity-conditioned knowledge retrieval. We conduct a detailed evaluation of nine general-purpose multimodal large language models, examining task performance and decomposing event understanding into event-type recognition and participant association. The results identify linking events to the correct players as an important bottleneck, with substantially lower performance when both event types and participants must be correct. We also present **BasketballAgent**, which combines specialized perception and knowledge retrieval to achieve the best available results on seven of the ten benchmark tasks. By composing its existing capabilities, the system additionally answers open-ended questions involving multiple players and mixed image–video inputs, demonstrating support for task formats beyond the benchmark's ten predefined tasks.

1 INTRODUCTION

Multimodal large language models (MLLMs) have made substantial progress in visual recognition and question answering, yet reliable understanding of professional basketball remains challenging. Understanding a single basketball play may require a model to recognize the event, identify the relevant players, localize the action in space and time, interpret the broadcast context, and connect visual evidence with structured basketball knowledge. We refer to the integration of these heterogeneous sources of evidence as comprehensive basketball understanding.

Research on sports understanding has addressed a broad spectrum of capabilities, ranging from player- and event-level perception to tasks such as professional QA, rule-aware reasoning, long-form temporal analysis, and multi-view understanding (Ramanathan et al., 2016; Li et al., 2021; Cui et al., 2023; Meng et al., 2026; Li et al., 2024; Rao et al., 2024; Xia et al., 2025; 2026; Cao et al., 2026; Chen et al., 2026; Pan et al., 2025). Existing works still tend to evaluate these capabilities separately, often using different annotation schemes, output formats, and sports, while some recent work has begun to bring multiple capabilities together within a unified framework (Rao et al., 2025a). This motivates a basketball-specific benchmark that jointly evaluates these capabilities and supports analysis of how failures in associating their outputs affect question answering.

To provide a unified evaluation setting, we construct **BasketballBench**, a multimodal benchmark comprising 7,600 annotated instances across ten basketball-specific tasks, examining basketball understanding from three perspectives: (1) basketball knowledge and retrieval, (2) broadcast and player perception, and (3) spatiotemporal and event understanding. The ten tasks span text, image, and video inputs and range from single-capability tests to composite questions so that the contributing capabilities can be read off separately. This organization supports consistent model comparison and fine-grained diagnosis of failures in connecting events, players, and knowledge.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We evaluate nine general-purpose MLLMs across these tasks and analyze their performance at the level of events and participants. A central finding is that associating an event with the correct player remains difficult even when the event type is recognized. For example, GPT-5.4 achieves 59.6% event-type F1 on Play Event QA, but only 29.8% when participants must also match. This gap motivates explicit evaluation of event–player associations alongside individual perception capabilities.

To explore how specialized perception and retrieval can address these difficulties, we also present **BasketballAgent**. The system connects event participants to player identities and structured knowledge, achieving the best available results on seven of the ten tasks against the evaluated standalone MLLMs. Its execution is driven by the user's question, without requiring a predefined task label. Two additional open-ended evaluations combine multiple event participants or image- and video-grounded identities, showing that the system can reuse its capabilities to answer questions beyond the benchmark's original task formats.

Our **three contributions** are: (i) **BasketballBench**, a multimodal benchmark with 7,600 questions across ten tasks that evaluate basketball perception, knowledge, and their integration; (ii) **a detailed evaluation** of nine MLLMs, identifying the association of events with specific players as an important bottleneck through event- and participant-level analysis; (iii) **BasketballAgent**, a perception-and-retrieval system that improves over standalone MLLMs on seven benchmark tasks and supports additional question compositions beyond the ten predefined formats.

2 Related Work

Sports understanding and evaluation. Sports benchmarks cover event recognition (Giancola et al., 2018; Deliège et al., 2021; Rao et al., 2025b), actor and action localization (Ramanathan et al., 2016; Li et al., 2021), tracking (Cui et al., 2023), and jersey recognition (Koshkina & Elder, 2024). Basketball studies additionally address court calibration (Sha et al., 2020), player identification (Senocak et al., 2018), knowledge-enhanced description (Wang et al., 2021; Xi et al., 2025), and player evaluation (Pan et al., 2025). Beyond these specialized tasks, Sports-QA examines temporal, causal, and counterfactual questions (Li et al., 2024); SPORTU and SportR study rule- and strategy-aware reasoning (Xia et al., 2025; 2026); SportsTime and SportMV-Bench extend evaluation to long-form temporal evidence and multiple views (Cao et al., 2026; Chen et al., 2026). BasketballBench emphasizes the associations between events, participants, and knowledge, with diagnostic conditions that probe failures in these associations.

Skills for multimodal reasoning. ReAct interleaves reasoning, actions, and observations (Yao et al., 2023); VisProg and ProViQ translate requests into visual or video programs (Gupta & Kembhavi, 2023; Choudhury et al., 2024). Voyager and MMSkills explore reusable procedural representations (Wang et al., 2024; Zhang et al., 2026a), while SportMV-Agent uses iterative planning and evidence collection for sports reasoning (Chen et al., 2026). BasketballAgent uses manually authored, on-demand instructions to guide evidence association and tool use within basketball. Our controlled ablation evaluates their effect on execution overhead and task performance with the controller and tools fixed; Section 5.5 examines unseen task compositions, while transfer across controllers remains outside the evaluation.

3 BasketballBench

3.1 Overview

BasketballBench contains 7,600 questions across ten text, image, and video tasks. Its annotations associate events with spatial locations, timestamps, participants, and structured records as required by each task. For example, a video knowledge question (Q10) links a specified event to its participant and then to that player's requested fact. The tasks require both acquiring the relevant evidence and connecting it to the requested answer.

3.2 Data Sources

BasketballBench is built from three complementary data sources from the 2025–2026 NBA season.

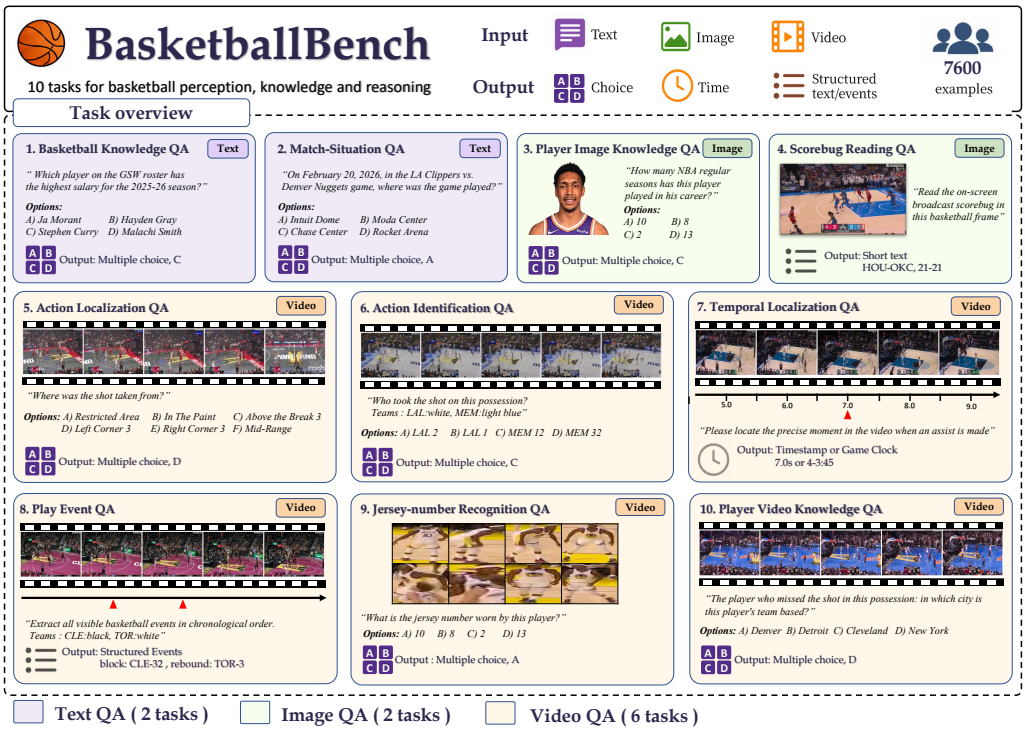


Figure 1: Representative examples of the ten BasketballBench tasks. The benchmark covers different aspects with outputs ranging from multiple-choice answers and timestamps to structured event sequences.

Broadcast Videos. We select 2,501 possession-level broadcast clips following the test split of BasketEvent (Zhang et al., 2026b). Each clip is associated with its source game and aligned with the corresponding structured play-by-play records.

Structured Records. Schedules, play-by-play, officials, uniform assignments, rosters, and coaching staffs for all 30 teams, together with profiles for 530 active players, are collected from NBA.com (National Basketball Association, 2026b) and Sportradar (Sportradar, 2026). A normalized SQL database supports question generation and answer retrieval.

Player Images. The reference gallery contains one official NBA.com headshot for each of the 530 players, linked to the database. For Q3, we separately collect 400 evaluation photographs and manually verify their identities and difference from the corresponding gallery images. Identity annotations are not exposed in the visual input; the official gallery headshots are not separately audited.

3.3 Benchmark Tasks

Table 1 lists the tasks and Figure 1 illustrates their inputs and outputs. We group them by the evidence they require.

Basketball Knowledge and Retrieval. Q1 and Q2 query general basketball facts and game-specific records, respectively. Q3 requires identifying a photographed player before retrieving a fact; Q10 additionally requires finding the participant of a specified video event.

Broadcast and Player Perception. Q4 reads teams and scores or game-clock information from scorebugs. Q9 reads a jersey number from supplied player crops, whereas Q6 must first associate the target action with its player in the broadcast video.

Spatiotemporal and Event Understanding. Q5 assigns a shot to one of six court regions; Q7 localizes an event in video time or game-clock coordinates. Q8 recovers an ordered sequence of visible events and their participants, requiring both event recognition and participant association.

Table 1: Summary of the ten BasketballBench tasks (7,600 examples) across text, image, and video inputs.

ID	Task	Input	N	Output	Primary capability
Q1	Basketball Knowledge QA	Text	1,200	Multiple Choice	Domain Knowledge Grounding
Q2	Match-Situation QA	Text	1,200	Multiple Choice	Contextual Retrieval
Q3	Player Image Knowledge QA	Image	400	Multiple Choice	Domain Knowledge Grounding
Q4	Scorebug Reading QA	Image	200	Structured Text	Broadcast Understanding
Q5	Action Localization QA	Video	600	Multiple Choice	Spatial Understanding
Q6	Action Identification QA	Video	500	Multiple Choice	Player Identification
Q7	Temporal Localization QA	Video	1,580	Timestamp	Temporal Understanding
Q8	Play Event QA	Video	1,000	Structured Events	Event Understanding
Q9	Jersey-Number Recognition QA	Video	300	Multiple Choice	Broadcast Understanding
Q10	Player Video Knowledge QA	Video	620	Multiple Choice	Domain Knowledge Grounding

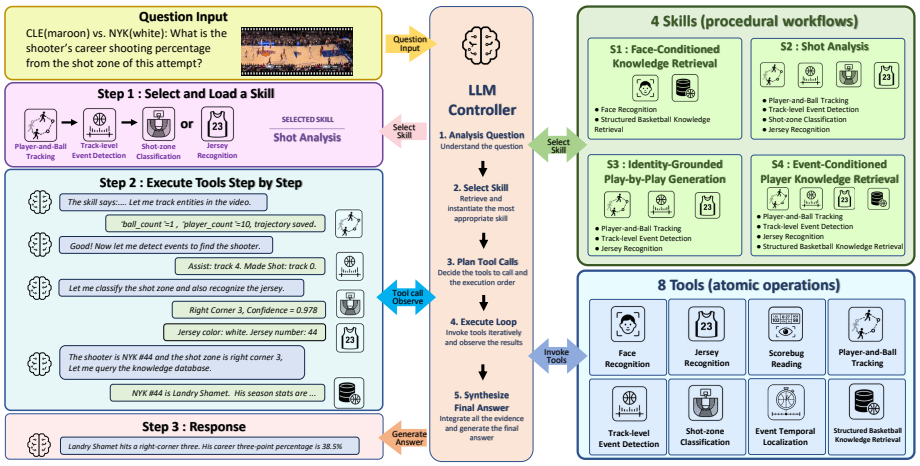


Figure 2: BasketballAgent architecture. The controller loads procedural guidance and invokes tools step by step. Skills guide evidence association and execution; tools produce perception and retrieval results.

3.4 Benchmark Construction

We instantiate manually designed QA templates from structured records and pair them with photographs or event-aligned clips. Aligned events, participants, clocks, and shot locations yield task-specific targets; uniform color–number pairs distinguish participants within a game.

Human Quality Control. Human review complements task-specific automatic validation. For Q4 and Q9, models assist with input preparation and candidate annotation, but their outputs are preliminary: final examples and reference labels are retained only after human verification and selection. Q7 timestamps are manually reannotated, and Q8 event sequences are manually checked and corrected. Appendix A.3.4 details model assistance and the scope of human review for each task.

4 Methodology

BasketballAgent combines eight perception and retrieval tools with four on-demand procedural skills, coordinated by a language-model controller.

4.1 Agent and Tool Interfaces

The input comprises a question and its images, videos, or game context. Tools return structured observations with identifiers and artifact references for subsequent calls: event detection assigns participant track IDs, jersey recognition attaches attributes to those tracks, and retrieval resolves identities and facts. Tools remain directly accessible without loading a skill.

4.2 Tools and Procedural Skills

Atomic tools. Face Recognition uses the open-source `face_recognition` library (Geitgey, 2017) to match an input face against official player headshots. Jersey Recognition, Scorebug Reading, and Event Temporal Localization use Qwen3.5-9B (Qwen Team, 2026a) with task-specific prompts and structured outputs. Player-and-Ball Tracking combines SAM 3 (Carion et al., 2025) with a fine-tuned RF-DETR (Robinson et al., 2026), whose detections filter erroneous player trajectories. Track-Level Event Detection applies PlayNet (Zhang et al., 2026b) to recognize events and their participant tracks. For Shot-Zone Classification, we add a six-class head to PlayNet and fine-tune it to predict the target shot's court region. Structured Basketball Knowledge Retrieval exposes fixed queries over the database in Section 3.2. RF-DETR and shot-zone training and validation data are disjoint from the benchmark videos; Appendix B.2 details training and interfaces.

Procedural guidance. Each skill is a manually authored instruction document; loading it returns guidance, while the controller still issues each tool call. The base prompt already provides tool dependencies and general execution rules. Skills add task-specific participant selection, evidence bindings, branches, batching, and stopping conditions through four procedures. These instructions make the intended execution constraints explicit and aim to reduce redundant exploration and tool calls. They provide a means of guiding and inspecting execution without changing the tools' predictive models; they do not guarantee correct intermediate evidence or a large accuracy gain.

(1) Face-Conditioned Knowledge Retrieval. Face Recognition identifies the player, and Structured Basketball Knowledge Retrieval obtains the requested fact. The procedure stops if the visual identity cannot be resolved.

(2) Shot Analysis. Player-and-Ball Tracking followed by Track-Level Event Detection identifies the shot and shooter track. The procedure branches to Shot-Zone Classification for action localization (Q5), or Jersey Recognition for shooter identification (Q6), combining jersey color and number with the question-provided team mapping.

(3) Identity-Grounded Play-by-Play Generation. After tracking and event detection, the procedure collects unique participant track IDs for one batched Jersey Recognition call. Results are joined back to events strictly by track ID and mapped to teams using the supplied color–team mapping; recognized identities are reused across events.

(4) Event-Conditioned Player Knowledge Retrieval. Tracking and event detection locate the requested event participant. Jersey attributes and the supplied team context determine a team–number pair. A roster query resolves the player before fact retrieval; unresolved identities trigger stopping.

These procedures reuse tools across the supported questions; Section 5.5 examines their reuse on additional compositional tasks. Transfer to other controllers is not evaluated. Full descriptions and instructions appear in Appendix B.4.

4.3 Inference Workflow

The DeepSeek-V4-Flash controller (DeepSeek-AI et al., 2026) receives the query, tool interfaces, and a catalog of skill names and descriptions. It can load matching guidance or invoke tools directly, proceeding through successive observations without an externally supplied task label.

Each run uses temperature zero and allows at most 12 controller turns and 10 tool calls. A lightweight verifier checks tool names, required arguments and types, media paths, and duplicate calls; failed checks are returned to the controller for revision. These checks validate call execution but do not establish the semantic correctness of a perception result. Execution traces record skill loading, calls, observations, and the final answer.

Table 2: Performance comparison across the ten BasketballBench tasks and their modality-level averages. Scores are reported using the primary metric of each task, and the best result in each column is highlighted in bold. Overall scores are unweighted means of the reported task scores: Q1–Q2 for Text, Q3–Q4 for Image, and Q5–Q10 for Video, rounded to one decimal place.

Model	TextQA		ImageQA		VideoQA						Overall		
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Text	Image	Video
Commercial APIs													
GPT-5.4	64.5	35.9	47.5	93.5	57.2	85.4	**77.5**	29.8	92.3	34.0	50.2	70.5	62.7
Claude Sonnet 5	54.1	7.9	36.5	92.5	23.5	55.8	51.3	14.3	89.7	27.1	31.0	64.5	43.6
Gemini 3.5 Flash	64.8	32.4	36.6	**94.0**	41.3	78.2	63.2	27.0	94.3	26.9	48.6	65.3	55.2
Open-Source Models													
Qwen2.5-VL-7B	37.3	36.8	29.6	75.5	19.7	34.2	23.0	1.6	92.3	19.4	37.1	52.6	31.7
Qwen3.5-4B	36.9	30.8	33.6	77.5	22.5	38.6	48.4	8.2	93.0	27.3	33.9	55.6	39.7
Qwen3.5-9B	42.2	34.6	36.3	66.5	19.7	47.2	47.2	10.4	93.7	26.0	38.4	51.4	40.7
VideoLLaMA3-7B	34.2	34.1	28.0	63.5	18.5	46.8	0	0	91.7	24.5	34.2	45.8	30.3
InternVL3.5-8B	41.1	34.1	29.0	66.0	16.7	38.6	13.1	0.5	**95.0**	25.0	37.6	47.5	31.5
Molmo2-8B	41.6	36.8	30.8	66.0	18.8	32.8	21.5	0.1	91.0	25.3	39.2	48.4	31.6
BasketballAgent (Ours)	**96.4**	**99.2**	**85.5**	88.5	**71.5**	**92.0**	47.3	**45.1**	94.3	**81.1**	**97.8**	**87.0**	**71.9**

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Baselines and comparison settings. We evaluate three commercial MLLMs—GPT-5.4 (OpenAI, 2026), Claude Sonnet 5 (Anthropic, 2026), and Gemini 3.5 Flash (Google DeepMind, 2026)—and six open-source models: Qwen3.5-4B, Qwen3.5-9B (Qwen Team, 2026a), Qwen2.5-VL-7B (Bai et al., 2025), VideoLLaMA3-7B (Zhang et al., 2025), InternVL3.5-8B (Wang et al., 2025), and Molmo2-8B (Clark et al., 2026). The main comparison measures system-level performance against standalone MLLMs: the agent additionally uses specialized tools and structured retrieval. Section 5.4 isolates the incremental effect of skills with other agent settings fixed.

Inference and evaluation protocol. For direct MLLM evaluation, broadcast videos are sampled at 2 fps; Q9 supplies eight uniformly sampled player crops. The Jersey Recognition tool likewise uses eight crops of the requested player, with tool-specific preprocessing detailed in Appendix B.2. Q1–Q3, Q6, Q9, and Q10 use four-way multiple-choice accuracy, while Q5 uses six shot-zone choices. Q4 is evaluated by exact match over the requested scorebug fields. Q7 uses all-example Acc@1s, macro-averaged over its equally sized video-timestamp and game-clock modes; the latter additionally requires the correct period. Q8 uses full-event F1, requiring both event type and participants to match in temporal order. Unparseable responses receive no credit under the task-specific evaluation rules. VideoLLaMA3-7B Q7/Q8 results are left unverified because the supplied evaluation package contains no prediction records for these tasks. Parsing normalization, sequence matching, and additional metrics are detailed in Appendix C.1.

5.2 BENCHMARK PERFORMANCE

General-purpose MLLMs. In Table 2, the strongest standalone results reach 94.0% on Scorebug Reading (Q4) and 95.0% on Jersey-Number Recognition (Q9). Structured event generation and visual knowledge retrieval remain difficult, with best available scores of 29.8% on Q8 and 26.0% among the completed open-source evaluations on the revised Q10; the within-task analyses below examine grounding more directly.

BasketballAgent. BasketballAgent achieves the best available results on seven of the ten tasks; For example, it exceeds the strongest standalone baseline by 62.4 percentage points on Q2 and the

Table 3: Q8 performance (%). Type + Participant F1 requires both event type and participants to match; Type F1 ignores participant identity. JSON and Part. Acc. denote JSON validity and participant accuracy.

Model	Type + Participant F1	Type F1	JSON	Part. Acc.
Commercial APIs				
GPT-5.4	29.8	59.6	99.9	46.7
Claude-Sonnet-5	14.3	43.9	95.5	24.5
Gemini 3.5 Flash	27.0	51.8	97.8	49.4
Open-source models				
Qwen2.5-VL-7B	1.6	27.5	**100.0**	5.1
Qwen3.5-4B	8.2	39.1	**100.0**	20.6
Qwen3.5-9B	10.4	45.8	99.9	22.2
VideoLLaMA3-7B	–	–	–	–
InternVL3.5-8B	0.5	11.1	**100.0**	2.7
Molmo2-8B	0.1	4.7	**100.0**	1.9
BasketballAgent	**45.1**	**60.6**	**100.0**	**69.4**

Table 4: Effect of target-player grounding on Q5 and Q6 accuracy (%). *Target* highlights the event-relevant player; Δ is the absolute gain over *Raw* in percentage points.

Model	Q5			Q6		
	Raw	Target	Δ	Raw	Target	Δ
GPT-5.4	57.2	**61.5**	+4.3	85.4	**97.0**	+11.6
Claude Sonnet 5	23.5	**33.7**	+10.2	55.8	**84.4**	+28.6
Gemini 3.5 Flash	41.5	**47.8**	+6.3	78.2	**94.6**	+16.4
Qwen3.5-9B	19.7	**36.3**	+16.7	47.2	**85.8**	+38.6
Molmo2-8B	18.8	**21.5**	+2.7	32.8	**45.0**	+12.2
InternVL3.5-8B	16.7	**17.3**	+0.7	38.6	**49.4**	+10.8
Mean	29.6	**36.4**	+6.8	56.3	**76.0**	+19.7

strongest completed open-source baseline by 55.2 points on the revised Q10. The knowledge-task gains also reflect access to structured records. The agent remains below the strongest standalone models on Q4, Q7, and Q9, so tool augmentation does not uniformly improve every capability.

5.3 Diagnosing Player–Event Grounding

Recognizing events versus grounding participants. Q8 separates event-type recognition from the stricter requirement of associating events with the correct participants. As shown in Table 3, GPT-5.4 approaches the agent in Type F1 (59.6% versus 60.6%), but the gap widens for Type + Participant F1 (29.8% versus 45.1%). Participant accuracy among type-aligned event pairs is also lower for GPT-5.4 (46.7% versus 69.4%). GPT-5.4's 99.9% JSON validity further shows that its low Full F1 is not primarily a formatting failure.

Target-player highlighting. To examine this difficulty within the same tasks, Table 4 compares the original video input with a diagnostic condition that highlights the ground-truth event-relevant player using a bounding box in every frame. Across the six evaluated models, this additional grounding information improves Q6 shooter-identification accuracy by 19.7 percentage points on average and Q5 shot-zone accuracy by 6.8 points. Together with the Q8 decomposition, these results provide evidence that player–event association is an important bottleneck. This diagnostic condition uses additional ground-truth localization, which is unavailable during normal agent inference.

Table 5: Modality-level ablation of procedural skills in BasketballAgent. Values are macro-averaged over tasks within each modality. Relative change is computed from unrounded modality averages as $(\text{Skills} - \text{No Skills})/\text{No Skills} \times 100\%$. Calls are actual atomic-tool invocations per question, including failed calls and counting each call in a batch separately; skill-loading calls and verifier-rejected proposals are excluded.

Modality	**Performance** (↑)			**Atomic Tool Calls** (↓)		
	No Skills	Skills	Rel. Δ	No Skills	Skills	Rel. Δ
TextQA	98.0	97.8	-0.25%	1.83	1.79	-2.26%
ImageQA	88.3	87.0	-1.42%	1.87	1.69	-9.49%
VideoQA	72.5	71.9	-0.76%	3.85	3.27	-15.16%

Table 6: BasketballAgent on Cross-Event Player Knowledge QA (A) and Cross-Modal Player Knowledge QA (B). Both requested attributes must be correct and in order.

Task	**Questions**	**Correct pairs**	**Pair accuracy (%)**
A: Cross-Event QA	50	24	48.0
B: Cross-Modal QA	50	32	64.0

5.4 Effect of Procedural Skills

Controlled removal of skills. The No-Skills setting removes only the four skills: the controller sees none of their descriptions and loads no skill instructions. All other settings remain unchanged, including the controller, atomic tools and their descriptions, structured database, base system prompt, verifier, and execution limits. The base prompt continues to provide the tool dependencies and general execution rules reproduced in Appendix B.4.1.

Performance and tool calls. Table 5 evaluates the execution role of procedural guidance by reporting tool calls together with task performance. The skills are designed to make tool ordering, evidence bindings, and stopping decisions explicit and to limit unnecessary exploration; large accuracy gains are not their primary objective. Task scores track any performance changes accompanying this guidance. Tool calls measure invocation overhead, not latency or total inference cost, and do not by themselves quantify compliance with the prescribed procedures. Skills reduce mean atomic-tool calls by 2.26%, 9.49%, and 15.16% for text, image, and video tasks, respectively, while modality scores decrease by 0.25, 1.25, and 0.55 percentage points. Including skill-loading calls, the corresponding call means are 1.79, 2.12, and 3.95, versus 1.83, 1.87, and 3.85 without skills. Thus, the reduction concerns atomic-tool use and does not establish lower total inference cost.

5.5 Beyond Predefined Task Formats

BasketballAgent accepts user questions without an externally supplied task label, allowing it to combine available capabilities beyond the ten predefined benchmark task formats. To evaluate this flexibility, we construct two additional tasks with 50 questions each that require multiple identity–knowledge associations.

A: Cross-Event Player Knowledge QA requires identifying players associated with different events within a video, such as the shooter and assister or the shooter and rebounder, and answering knowledge questions about these players. **B: Cross-Modal Player Knowledge QA** requires identifying a player in an image and an event-associated player in a video, and answering knowledge questions about both. Each task requests the players' heights or colleges in a specified order. Answers are open-ended to avoid candidate-answer shortcuts, and a pair is correct only when both normalized attributes match the references in the requested order.

As shown in Table 6, BasketballAgent achieves 48% pair accuracy on A and 64% on B. On A, it reuses *Event-Conditioned Player Knowledge Retrieval* to resolve multiple participants and re-

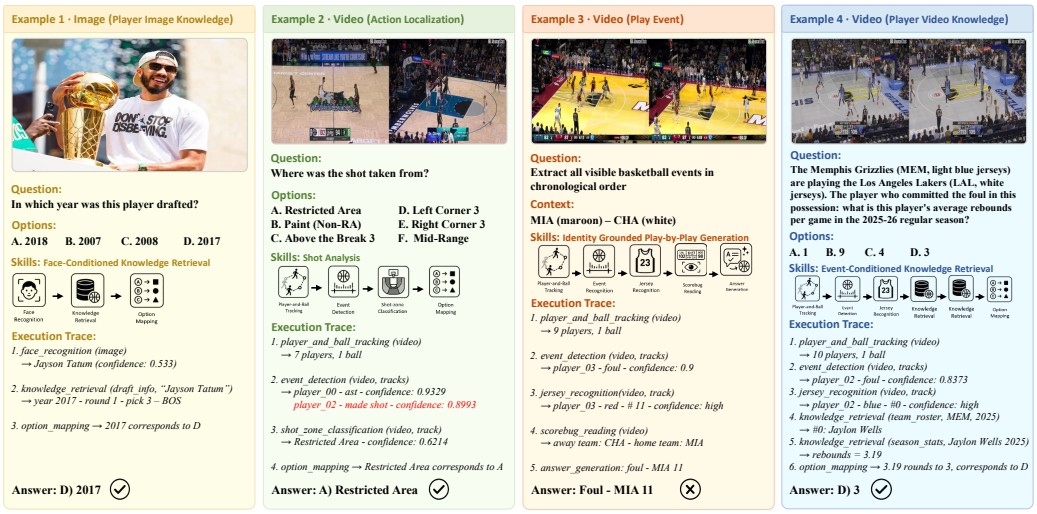


Figure 3: Representative BasketballAgent execution traces, showing selected procedural guidance, individual tool calls, intermediate observations, and final answers. The examples illustrate evidence association across tools, as well as unnecessary calls and upstream perception errors.

trieve their attributes; this skill is loaded in 47/50 runs. On B, 36/50 runs combine *Face-Conditioned Knowledge Retrieval* with either event-conditioned retrieval or *Shot Analysis*, joining evidence from two modalities into one ordered answer. No pair-specific skill is loaded in these runs. The results demonstrate that the system can execute new question compositions using its existing capabilities, rather than requiring a separate skill for each benchmark format. They establish this flexibility for the evaluated compositions, with remaining errors in jointly resolving both requested attributes. Appendix C.3 provides task construction, examples, execution details, and media-overlap analysis.

5.6 Qualitative Results

The third example in Figure 3 exposes two limitations: an unnecessary Scorebug Reading call despite the supplied team context, and an erroneous Track-Level Event Detection result that propagates into the answer. Skills do not eliminate redundant calls, and the verifier's call-validity checks cannot correct upstream semantic errors.

6 Conclusion

We introduced **BasketballBench**, a comprehensive multimodal benchmark comprising 7,600 questions across ten tasks spanning text, image, and video modalities. Our experiments reveal that current MLLMs remain limited in the fine-grained spatial and temporal perception of basketball games. Their performance degrades further on compositional questions that require multiple capabilities, such as jointly recognizing events and players, localizing actions in space and time, and incorporating basketball knowledge. To address these challenges, we developed **BasketballAgent**, a hierarchical agent that organizes basketball-specific tools into reusable procedural skills. BasketballAgent outperforms general-purpose MLLMs, demonstrating the effectiveness of composing specialized perception and retrieval capabilities for comprehensive basketball understanding. Together, BasketballBench and BasketballAgent provide a foundation for advancing fine-grained, compositional multimodal understanding in basketball and other complex sports domains.

AI USE STATEMENT

Generative AI tools were used to assist the authors in searching for related work in the field, developing code, and polishing the language of the manuscript. The authors take full responsibility for the final manuscript and accompanying code.

REFERENCES

Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022.

Anthropic. Claude Sonnet 5 System Card. https://www.anthropic.com/claude-sonnet-5-system-card, June 2026. Accessed: 2026-07-27.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Siyu Cao, Lu Zhang, Ruizhe Zeng, and Zhi-Yong Liu. Towards temporal compositional reasoning in long-form sports videos. *arXiv preprint arXiv:2604.22226*, 2026.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Radle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Kerui Chen, Jinglu Wang, Xiaoyi Zhang, and Yan Lu. Beyond the single camera: Agentic multi-view reasoning in sports video understanding. *arXiv preprint arXiv:2607.11844*, 2026.

Rohan Choudhury, Koichiro Niinuma, Kris M. Kitani, and László A. Jeni. Video question answering with procedural programs. In *Computer Vision – ECCV 2024*, pp. 315–332. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72920-1_18.

Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Rohun Tripathi, Sangho Lee, Mohammadreza Salehi, Jason Ren, Chris Dongjoo Kim, Yinuo Yang, Vincent Shao, Yue Yang, Weikai Huang, Ziqi Gao, Taira Anderson, Jianrui Zhang, Jitesh Jain, George Stoica, Ali Farhadi, and Ranjay Krishna. Molmo2: Open weights and data for vision-language models with video understanding and grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28652–28668, 2026.

Yutao Cui, Chenkai Zeng, Xiaoyu Zhao, Yichun Yang, Gangshan Wu, and Limin Wang. SportsMOT: A large multi-object tracking dataset in multiple sports scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9921–9931, 2023.

DeepSeek-AI et al. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Adrien Deliège, Anthony Cioppa, Silvio Giancola, Meisam J. Seikavandi, Jacob V. Dueholm, Kamal Nasrollahi, Bernard Ghanem, Thomas B. Moeslund, and Marc Van Droogenbroeck. SoccerNet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4508–4519, 2021.

Adam Geitgey. `face_recognition`: A face recognition library for python. https://github.com/ageitgey/face_recognition, 2017. Accessed: 2026-07-29.

Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. SoccerNet: A scalable dataset for action spotting in soccer videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1711–1721, 2018.

Google DeepMind. Gemini 3.5 Flash: Model Card. https://deepmind.google/models/model-cards/gemini-3-5-flash/, May 2026. Accessed: 2026-07-27.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962, 2023.

Maria Koshkina and James H. Elder. A general framework for jersey number recognition in sports video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 3235–3244, 2024.

Haopeng Li, Andong Deng, Jun Liu, Hossein Rahmani, Yulan Guo, Bernt Schiele, Mohammed Bennamoun, and Qiuhong Ke. Sports-QA: A large-scale video question answering benchmark for complex and professional sports. *arXiv preprint arXiv:2401.01505*, 2024.

Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. MultiSports: A multi-person video dataset of spatio-temporally localized sports actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13536–13545, 2021.

Zi Meng, Wanli Song, Yi Hu, Jiayuan Rao, and Gang Chen. SoccerRef-Agents: Multi-agent system for automated soccer refereeing. In *Sports Analytics: Third International Conference, ISACE 2026, Proceedings*, volume 16610 of *Lecture Notes in Computer Science*, pp. 403–427. Springer, 2026. doi: 10.1007/978-3-032-27272-0_27.

National Basketball Association. NBA LockerVision. https://lockervision.nba.com/, 2026a. Accessed: 2026-07-31.

National Basketball Association. NBA.com: The Official Website of the National Basketball Association. https://www.nba.com/, 2026b. Accessed: 2026-07-27.

OpenAI. GPT-5.4 Thinking System Card. https://openai.com/index/gpt-5-4-thinking-system-card/, March 2026. Accessed: 2026-07-27.

Yulu Pan, Ce Zhang, and Gedas Bertasius. BASKET: A large-scale video dataset for fine-grained skill estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28952–28962, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026a. Accessed: 2026-07-27.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model. https://qwen.ai/blog?id=qwen3.6-27b, April 2026b. Accessed: 2026-07-29.

Vignesh Ramanathan, Jonathan Huang, Sami Abu-El-Haija, Alexander Gorban, Kevin Murphy, and Li Fei-Fei. Detecting events and key actors in multi-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3043–3053, 2016.

Jiayuan Rao, Haoning Wu, Chang Liu, Yanfeng Wang, and Weidi Xie. MatchTime: Towards automatic soccer game commentary generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1671–1685. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.99.

Jiayuan Rao, Zifeng Li, Haoning Wu, Ya Zhang, Yanfeng Wang, and Weidi Xie. Multi-agent system for comprehensive soccer understanding. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 3654–3663, 2025a.

Jiayuan Rao, Haoning Wu, Hao Jiang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Towards universal soccer video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8384–8394, 2025b.

Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. RF-DETR: Neural architecture search for real-time detection transformers. In *International Conference on Learning Representations*, 2026.

Arda Senocak, Tae-Hyun Oh, Junsik Kim, and In So Kweon. Part-based player identification using deep convolutional representation and multi-scale pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1732–1739, 2018.

Long Sha, Jennifer Hobbs, Panna Felsen, Xinyu Wei, Patrick Lucey, and Sujoy Ganguly. End-to-end camera calibration for broadcast videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13627–13636, 2020.

Sportradar. NBA API Basics. https://developer.sportradar.com/basketball/docs/nba-ig-api-basics, 2026. Accessed: 2026-07-29.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.

Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6847–6857, 2021.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Zeyu Xi, Ge Shi, Xuefen Li, Junchi Yan, Zun Li, Lifang Wu, Zilin Liu, and Liang Wang. A simple yet effective knowledge guided method for entity-aware video captioning on a basketball benchmark. *Neurocomputing*, 619:129177, 2025.

Haotian Xia, Zhengbang Yang, Junbo Zou, Rhys Tracy, Yuqing Wang, Chi Lu, Christopher Lai, Yanjun He, Xun Shao, Zhuoqing Xie, Yuan-Fang Wang, Weining Shen, and Hanjie Chen. SPORTU: A comprehensive sports understanding benchmark for multimodal large language models. In *International Conference on Learning Representations*, 2025.

Haotian Xia, Haonan Ge, Junbo Zou, Hyun Woo Choi, Xuebin Zhang, Danny Suradja, Botao Rui, Ethan Tran, Wendy Jin, Zhen Ye, Xiyang Lin, Christopher Lai, Shengjie Zhang, Junwen Miao, Shichao Chen, Rhys Tracy, Vicente Ordonez, Weining Shen, and Hanjie Chen. SportR: A benchmark for multimodal large language model reasoning in sports. In *International Conference on Learning Representations*, 2026.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. VideoLLaMA 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

Kangning Zhang, Shuai Shao, Qingyao Li, Jianghao Lin, Lingyue Fu, Shijian Wang, Wenxiang Jiao, Yuan Lu, Weiwen Liu, Weinan Zhang, and Yong Yu. MMSkills: Towards multimodal skills for general visual agents. *arXiv preprint arXiv:2605.13527*, 2026a.

Yu Zhang, Jiayuan Rao, Haoning Wu, and Weidi Xie. BasketEvent: Understanding who did what and when in basketball videos. *arXiv preprint arXiv:2607.21267*, 2026b.

Supplementary Material

Towards Comprehensive Basketball Understanding

CONTENTS

1 Introduction **1**

2 Related Work **2**

3 BasketballBench **2**
3.1 Overview . . . 2
3.2 Data Sources . . . 2
3.3 Benchmark Tasks . . . 3
3.4 Benchmark Construction . . . 4

4 Methodology **4**
4.1 Agent and Tool Interfaces . . . 5
4.2 Tools and Procedural Skills . . . 5
4.3 Inference Workflow . . . 5

5 Experiments **6**
5.1 Experimental Setup . . . 6
5.2 Benchmark Performance . . . 6
5.3 Diagnosing Player–Event Grounding . . . 7
5.4 Effect of Procedural Skills . . . 8
5.5 Beyond Predefined Task Formats . . . 8
5.6 Qualitative Results . . . 9

6 Conclusion **9**

A BasketballBench Construction and Specifications **14**
A.1 Data Sources . . . 14
A.2 Structured Basketball Knowledge Base . . . 14
A.3 Benchmark Construction Protocol . . . 15
A.4 Task Definitions and Generation . . . 17

B BasketballAgent Architecture and Implementation **19**
B.1 Agent Architecture . . . 19
B.2 Atomic Tool Library . . . 20
B.3 Composite Skill Library . . . 22
B.4 Complete Agent-Visible Prompts . . . 23

C Additional Evaluation Results **29**
C.1 Evaluation Metrics . . . 29
C.2 Fine-Grained Results . . . 31
C.3 Compositional Player-Pair Tasks . . . 34

A BASKETBALLBENCH CONSTRUCTION AND SPECIFICATIONS

This section describes the data sources, structured knowledge base, common construction protocol, and task-specific generation procedures of BasketballBench.

A.1 DATA SOURCES

BasketballBench integrates structured basketball records, official player images, possession-level broadcasts with game metadata, play-by-play (PBP) records, and game-specific uniform metadata. Instances are anchored in the 2025–2026 NBA season, with historical statistics and drafts retained where needed. All sources are frozen before generation.

Structured Basketball Records. Sportradar's NBA API (Sportradar, 2026) provides information on all 30 franchises, including teams, coaches, players, season statistics, schedules, game and period summaries, box scores, officials, injuries, drafts, and free agents. The normalized SQLite database described in Section A.2 grounds the four knowledge-oriented tasks.

Player Images. NBA.com (National Basketball Association, 2026b) supplies one official headshot for each of 530 registered players, forming the face-recognition gallery. We additionally collect 400 evaluation photographs, each manually verified to depict the associated player and to be distinct from that player's official gallery headshot.

Broadcast Videos and Associated Game Metadata. The video set comprises 2,501 possession-level broadcasts from the BasketEvent test split (Zhang et al., 2026b), covering 33 games in 2025–2026. The 1280×720, 25-fps clips last 3.48–18.96 s (mean 9.45 s; standard deviation 1.81 s). Each clip is linked to its game date, teams, and active rosters, including game-specific jersey numbers. These metadata support PBP alignment and conversion of identity annotations into observable participant labels; they are not model inputs unless explicitly included in a task prompt.

Play-by-play Records. NBA.com (National Basketball Association, 2026b) provides 43,558 chronological PBP events from 266 games, including all 33 video-source games. Each record specifies the game clock, action, participants, and, when applicable, shot outcome and location. We use ten event types—made shot, missed shot, free throw, rebound, turnover, foul, steal, assist, block, and jump ball—and six shot regions: Restricted Area, In the Paint (Non-RA), Mid-Range, Above the Break 3, Left Corner 3, and Right Corner 3. A clip may align with several consecutive PBP events (Section A.3.2).

Game-specific Uniform Metadata. NBA LockerVision (National Basketball Association, 2026a) provides reference images for the uniform editions of all 30 teams and the edition assigned to each team in each 2025–2026 game. GPT-5.4 (OpenAI, 2026) identifies dominant and secondary colors from the static reference images; these descriptions are normalized to canonical English labels and manually verified (Section A.3.3). Joining the game assignment with team and roster metadata converts a PBP identity into a visible color–number label (e.g., `white-23`); prompts then map colors to team tricodes. LockerVision data and derived colors are used only for offline construction and verification: they are neither stored in the structured basketball knowledge base described in Section A.2 nor exposed to tools or evaluated models. GPT-5.4 never annotates or answers questions about benchmark broadcast clips.

A.2 STRUCTURED BASKETBALL KNOWLEDGE BASE

Overview. `NBA_DB` is a SQLite knowledge base comprising 18 normalized tables derived from Sportradar records. It grounds the four knowledge-oriented tasks while keeping generated questions, choices, visual annotations, and sample mappings outside the database.

A.2.1 TEMPORAL SNAPSHOT AND DATA COVERAGE

The primary snapshot corresponds to the 2025–2026 NBA season and covers games from the October 2025 preseason through Game 5 of the NBA Finals on June 13, 2026. It contains 1,423 schedule

records across preseason, regular-season, in-season tournament, play-in, and postseason competition, with detailed box scores for all 1,230 regular-season games. Historical player-season statistics cover season years 2012–2025, and draft records span 2003–2026. Table A.1 summarizes the benchmark-relevant coverage.

Table A.1: Coverage of the structured basketball knowledge base.

Data Category	**Coverage**	**Scale**
Teams	All current NBA teams	30 teams
Players	Player profiles, including a small number of box-score-only stub records	633 players
Seasonal schedules	All 2025–2026 games across PRE, REG, IST, PIT, and PST	1,423 games
Detailed game records	Team-, player-, and period-level box scores for all regular-season games	1,230 games
Player seasons	Player–team affiliations by season and competition type, covering season years 2012–2025	7,940 records
Player-season statistics	Season totals and per-game averages across the collected statistical fields	15,880 rows
Draft records	NBA draft records from 2003 through 2026	551 records
Officials	Referee profiles and regular-season game assignments	82 officials, 3,714 assignments

A.2.2 Logical Organization

The tables form four conceptual modules: team and player records (profiles, rosters, coaches, jersey numbers, and drafts); season records (affiliations and statistics); schedules (games and teams); and game details (summaries, officials, box scores, and period statistics). Season totals and per-game averages are separated so that templates can specify their statistical scope.

A.2.3 Entity Relationships

Player profiles link to current rosters, drafts, and season-specific affiliations, which in turn link to total and per-game statistics. Separating current membership from historical affiliation prevents present-day team or jersey data from being applied to another season. Schedules connect the home and away teams to summaries, officials, box scores, and period records, preserving the player, team, season, and game context of each fact.

A.2.4 Use in Benchmark Construction

Templates specify the required entities, temporal constraints, relations, and answer type. Basketball Knowledge QA retrieves profiles, rosters, drafts, schedules, and season statistics; Match-Situation QA uses game and period records. The two visual knowledge tasks first resolve a player and then retrieve the requested fact. Questions, answers, and distractors are stored separately, leaving `NBA_DB` as a reusable factual source.

A.3 Benchmark Construction Protocol

A.3.1 General Generation Procedure

Each task defines admissible source entities, answer relations, and an unambiguous output schema. Before quota-based sampling, we remove missing or unresolved fields, duplicate prompts, and cases admitting multiple answers. Multiple-choice answers are derived from source records; type-compatible distractors use other observed categorical values or nearby numerical values, with no duplicate choices and balanced answer positions. Open-ended answers are serialized from aligned annotations under a fixed schema.

Multimodal instances additionally require consistent links between visual evidence and source records: image questions must consistently link the evaluation photograph, the official gallery head-

shot, the roster entry, and the database profile to the same player, while video questions require aligned PBP events, participant attributes, and game metadata.

A.3.2 Video–Record Alignment and Temporal Annotation

Each clip is first linked to its source game and the corresponding PBP interval, retaining the ordered events, participants, shot outcomes, and shot locations recorded in the official data. As an initial automatic alignment stage, we process all 4,244 possession clips in the BasketEvent test split before task-specific sampling. Each clip is sampled at 1 fps, and Qwen3.6-27B (Qwen Team, 2026b) reads the scorebug clock in every sampled frame, producing a per-second mapping between clip-relative video time and game-clock time. We retain only clips for which all sampled scorebugs are readable and the recovered clocks form a consistent countdown sequence; clips with clock stoppages, missing or unreadable scorebugs, or inconsistent OCR results are excluded. Matching an event's official PBP clock to this mapping provides an initial estimate of its clip-relative video timestamp. However, this automatically derived timestamp is not used as the final temporal ground truth. Official PBP timestamps may lag the visually observable occurrence of an event by one or two seconds, and the 1-fps OCR mapping introduces additional temporal quantization. We therefore manually reannotate all 790 target-event instances selected for Temporal Localization QA (Q7). For each instance, annotators inspect the clip and record both the clip-relative video timestamp of the visible event and the corresponding period and remaining game-clock value shown in the broadcast at that moment. These manually corrected annotations serve as the final reference labels for the video-timestamp and game-clock modes of Q7. The OCR-based mapping and official PBP timestamps are used only for candidate alignment and quality filtering. Game-day rosters and uniform assignments convert source identities into team, jersey-number, and color attributes. Prompts expose the two team–color correspondences, allowing outputs to use observable team and jersey labels rather than hidden player identities.

A.3.3 Uniform-Color Normalization

Edition-level colors are mapped to a controlled English vocabulary and joined with each game's home and away assignments. Tasks requiring color grounding retain only games with unambiguous metadata for both teams. Model-assisted descriptions are manually checked against the reference images before use.

A.3.4 Human Quality Control and Model-Assisted Annotation

Model assistance produces intermediate artifacts and candidate annotations; it does not determine the final reference labels for Q4 or Q9. Human reviewers inspect the corresponding visual evidence and select the examples whose annotations are correct and unambiguous. The final benchmark thus uses human-approved references rather than unverified model predictions.

Q4: candidate scorebug readings. Qwen3.6-27B (Qwen Team, 2026b) proposes team identities, scores, periods, and game-clock readings from broadcast frames. After filtering incomplete parses or invalid team metadata, human reviewers check candidate readings against the visible scorebug. The final set retains 200 correct, readable examples from 300 model-generated candidates. Reference strings are serialized from the human-verified fields. Evaluated methods receive the broadcast frame and question, not the annotation model's preliminary reading. This procedure selects verified candidates; model outputs are not accepted as ground truth merely because they parse successfully.

Q9: player crops and candidate jersey numbers. SAM 3 (Carion et al., 2025) assists in obtaining player tracks and crops, which are intermediate inputs for constructing the jersey-number questions. It does not supply jersey-number labels. Qwen3.6-27B (Qwen Team, 2026b) proposes candidate jersey-number readings for the selected player crops. Human reviewers examine all candidates against the visual evidence and retain 300 correct, unambiguous examples. The approved jersey number is used as the reference answer; model-generated readings alone are insufficient for inclusion. Each evaluated method receives eight chronologically ordered crops of the target player, together with the question and answer choices, rather than the preliminary model reading. This separates automated preparation of visual inputs from human approval of the final labels.

Human review for other tasks. For Q3, all 400 evaluation photographs are manually checked for player identity and difference from the corresponding official gallery headshot; the gallery itself is not separately audited. For Q7, annotators reannotate all 790 target-event instances at the visible event time, recording both video time and the corresponding period and game-clock value. These labels yield 1,580 questions and replace the automatic PBP–video timestamp estimates, as detailed in Section A.3.2. All 1,000 Q8 examples are checked for event completeness, order, type, team, and jersey number; 258 examples (25.8%) require correction. Q1, Q2, Q5, Q6, and Q10 derive from structured records and undergo schema validation, missing-field filtering, and duplicate removal, without additional instance-level human review. Human verification is therefore task-specific rather than a claim that every benchmark instance has been manually relabeled.

A.4 Task Definitions and Generation

The benchmark contains 7,600 instances: 2,400 text, 600 image, and 4,600 video questions. Below we summarize task-specific selection and answer construction; shared rules follow Section A.3. The examples transcribe released benchmark instances; line breaks and list formatting are compacted for typesetting, while the options and output constraints are preserved. Videos are identified by game and clip IDs because the same assets are shared across video tasks.

A.4.1 Q1: Basketball Knowledge QA

This text-only task contains 1,200 four-way questions from 47 manually designed types spanning player profiles, team attributes, rosters and coaches, drafts, 2025–2026 statistics, career aggregates, and schedules. Candidates require valid entity links, and season questions distinguish totals from per-game averages. Templates retrieve and type-normalize the reference value; distractors use other observed categorical values or nearby values of the same statistic. Examples are allocated across question types, with answers balanced over A–D.

Media. None (text-only input).
Prompt. What is Deandre Ayton's career regular-season total rebounds? A. 4453; B. 4742; C. 5101; D. 5113. Return only the letter of your choice (A, B, C, or D), with no extra words.
Ground truth. B (4742).

A.4.2 Q2: Match-Situation QA

This task contains 1,200 four-way questions (40 per type) about specific regular-season games. Its 30 types cover game context, final and period scores, officials, attendance, lead changes and ties, box scores, leaders, player statistics, and derived outcomes. Date and matchup must uniquely identify a game with all required records. References join schedules with summaries, officials, period scoring, and box scores; distractors are valid same-domain names, venues, or nearby numerical values. Answer positions are balanced.

Media. None (text-only input).
Prompt. On March 12, 2026, in the Indiana Pacers vs. Phoenix Suns game, who was the top scorer of the game? A. Isaiah Jackson; B. Buddy Hield; C. Jalen Suggs; D. Devin Booker. Return only the letter of your choice (A, B, C, or D), with no extra words.
Ground truth. D (Devin Booker).

A.4.3 Q3: Player Image Knowledge QA

This task pairs a photo of an NBA player with one of 23 knowledge-question types, covering player profiles, 2025–2026 statistics, draft history, and career aggregates. Each sample requires consistent roster, image, and database links, and replaces the player's name with "this player" to force visual recognition before retrieval. All photos are human-verified to ensure that they contain a clear frontal view of the player's face and are distinct from the official headshots. The 400 questions follow Q1's typed answer and distractor rules. Image assignment favors player diversity and unused evaluation photographs, with balanced answer positions.

Media. One player photograph.
Prompt. How many NBA regular seasons has this player played in his career? A. 10; B. 8; C. 2; D. 13. Return only the letter of your choice (A, B, C, or D), with no extra words.
Ground truth. C (2).

A.4.4 Q4: SCOREBUG READING QA

This task evaluates broadcast-graphic reading on 200 human-selected frames. Qwen3.6-27B proposes preliminary scorebug readings; the final references come from human verification and selection, as detailed in Section A.3.4. Half of the open-ended questions use `AWAY-HOME, away_score-home_score`, and half use `PERIOD, MM:SS`; references are serialized from the verified labels.

Media. One broadcast frame.
Prompt. Read the on-screen broadcast scorebug in this basketball frame. The left side is the away team and the right side is the home team (standard NBA layout). What is the current score? Report the away-team tricode, home-team tricode, and both team scores. Output exactly one line in the format `AWAY-HOME, away_score-home_score`; for example, `LAL-MEM, 80-95`. Return only that line, with no extra words.
Ground truth. `DEN-POR, 65-62`.

A.4.5 Q5: ACTION LOCALIZATION QA

This task retains possessions with exactly one made or missed shot whose PBP location maps unambiguously to Restricted Area, In the Paint (Non-RA), Mid-Range, Above the Break 3, Left Corner 3, or Right Corner 3. Each video asks "Where was the shot taken from?" and presents the complete six-region taxonomy. The 600 examples are class-balanced; multiple-shot possessions and missing locations are excluded.

Media. One possession video.
Prompt. Where was the shot taken from? The six zones are: Restricted Area (the semicircle under the rim); In The Paint (Non-RA) (the key excluding that semicircle); Above the Break 3 (the top-center three-point arc); Left and Right Corner 3 (the corresponding sideline–baseline intersections); and Mid-Range (outside the paint but inside the arc). A. Restricted Area; B. In The Paint (Non-RA); C. Above the Break 3; D. Left Corner 3; E. Right Corner 3; F. Mid-Range. Return only the letter (A–F), with no extra words.
Ground truth. D (Left Corner 3).

A.4.6 Q6: ACTION IDENTIFICATION QA

This task asks which player took the possession's unique made or missed shot. Eligible shooters require valid teams and jersey numbers, and both teams require game-specific colors. Prompts provide the team–color mapping, and choices use team tricodes plus jersey numbers. The ground truth shooter is paired with one same-team and two opposing-team distractors, testing team discrimination and jersey recognition. The 500 four-way questions have balanced answer positions.

Media. One possession video.
Prompt. Who took the shot on this possession? The shooter is identified by team tricode and jersey number (e.g., "LAL 23"). Teams (tricode: jersey color): LAL: white, MEM: light blue. A. LAL 2; B. LAL 1; C. MEM 12; D. MEM 32. Return only the letter of your choice (A, B, C, or D), with no extra words.
Ground truth. C (MEM 12).

A.4.7 Q7: TEMPORAL LOCALIZATION QA

This task covers made shots, missed shots, assists, blocks, rebounds, steals, turnovers, and fouls. A clip is eligible only when the queried event type occurs exactly once and the event can be localized unambiguously in the video. All selected target events are manually annotated at their visually observable occurrence. Each annotation records two synchronized temporal coordinates: the clip-relative video timestamp and the period plus remaining game-clock value displayed on the scorebug at that moment. Each of the 790 target-event instances is converted into two open-ended questions, one for each temporal mode, yielding 1,580 questions in total. The dataset is stratified by event class and question formulation.

Media. One possession video.
Prompt (video-time variant). Please locate the precise moment in the video when an assist is made (a pass leading directly to a made basket), and output the corresponding video timestamp for that exact instant. Please strictly follow the output format example: `[timestamp_in_seconds]`, e.g., `[5.0]`.
Ground truth. `[7.0]`.

A.4.8 Q8: PLAY EVENT QA

This task represents an entire possession as ordered JSON over the ten-event taxonomy. Every event requires resolvable team and jersey identities, and the prompt supplies both team–color mappings.

Assists, blocks, and steals are emitted as distinct events. The 1,000 clips are sampled to preserve coverage of rare assists, steals, and blocks before filling the remaining quota.

Media. One possession video.
Prompt. You are given a basketball possession clip. Game context: Team A is MIA in maroon; Team B is CHA in white. Extract all visible basketball events in chronological order. Allowed event types are `missed_shot`, `made_shot`, `rebound`, `assist`, `steal`, `turnover`, `block`, `free_throw`, `foul`, and `jump_ball`. Output only valid JSON with an `events` list, where each item has `event_type` and `participants`. Participants use `TRICODE-JERSEY_NUMBER`; the tricode must be MIA, CHA, or `unclear`, and an unreadable number is written as `unclear`. Every non-jump-ball event has one role-specific participant—the shooter, rebounder, assisting passer, stealer, turnover committer, blocker, free-throw shooter, or fouler—and `jump_ball` has two. Repeat recurring event types in chronological order; if none occurs, output `{"events": []}`. Do not add explanations, timestamps, confidence scores, or text outside the JSON.
Ground truth. `{"events": [{"event_type": "foul", "participants": ["MIA-24"]}]}`.

A.4.9 Q9: Jersey-Number Recognition QA

This task selects one sufficiently long player track per possession and samples eight valid crops in temporal order. SAM 3 assists crop preparation, and Qwen3.6-27B proposes preliminary jersey readings; human reviewers verify and select the final annotations (Section A.3.4). A four-way question uses the human-approved jersey number as its answer and NBA-profile jersey numbers as distractors. The 300 examples use distinct possession–player pairs and balanced answer positions.

Media. Eight chronologically ordered player crops.
Prompt. These 8 images show crops of a single basketball player, uniformly sampled across a possession clip. The images are ordered chronologically, from the beginning to the end of the clip. What is the jersey number worn by this player? A. 34; B. 10; C. 22; D. 35. Return only the letter of your choice (A, B, C, or D), with no extra words.
Ground truth. B (10).

A.4.10 Q10: Player Video Knowledge QA

This task combines event-conditioned player recognition with knowledge retrieval. A clip must contain a unique primary participant describable by an event role (e.g., "the player who made the shot"), linked to a database profile. Ambiguous multi-team cases are excluded where current-team membership is queried. Prompts provide the team–color context and ask one of 27 four-way question types spanning statistics, profiles, teams, drafts, and source-game box scores. Database answers use same-type or nearby-value distractors. The revised set contains 620 questions with 155 answers at each of the four option positions.

Media. One possession video.
Prompt. In this clip, the New York Knicks (NYK, black jerseys) are playing the Miami Heat (MIA, white jerseys). The player who missed the shot in this possession: in which city is this player's team based? A. Denver; B. Detroit; C. Miami; D. New York. Return only the letter of your choice (A, B, C, or D), with no extra words.
Ground truth. D (New York).

B BasketballAgent Architecture and Implementation

This section details *BasketballAgent* using the paper-facing names of all four skills and eight tools. Repository identifiers appear only in the verbatim controller-visible prompts (Section B.4).

B.1 Agent Architecture

A temperature-zero DeepSeek-V4-Flash controller (DeepSeek-AI et al., 2026) interprets the query, optionally loads a skill, invokes tools, and iterates over their structured outputs. Skills specify which tool outputs are required and the order in which tools should be called; they do not generate tool results. Their role is to expose execution constraints and guide evidence reuse and stopping, with the aim of reducing unnecessary calls. The controller remains responsible for execution, and the instructions do not guarantee semantic correctness or improve the underlying tool models.

Each run allows at most 12 controller turns and 10 tool calls. The lightweight verifier mentioned in the main paper is a fixed Python program that checks each function call before execution. It validates the tool name, required arguments and their types, media and path availability, and duplicate calls; a failed check is returned to the controller for revision. Media and artifact paths are anonymized to

hide source-game identifiers and resolved only at execution. The trace records turns, verification, tools, skills, outputs, and the final answer for reproducibility.

B.2 Atomic Tool Library

The eight atomic tools expose typed, task-bounded interfaces. Four trajectory-dependent video tools reuse frozen precomputed artifacts when available and otherwise run online; caching affects latency, not semantics.

B.2.1 Player-and-Ball Tracking

Implementation. Player-and-Ball Tracking combines SAM 3 (Carion et al., 2025), a fine-tuned RF-DETR detector (Robinson et al., 2026), and BoT-SORT association (Aharon et al., 2022). SAM 3 propagates player masks, while RF-DETR detects on-court players and the ball. To remove referees, bench players, and staff, a SAM trajectory must overlap an RF-DETR player detection by at least 0.3 IoU in at least half of its observed frames; ball trajectories come directly from detector–tracker output. The result stores normalized (x, y, w, h) boxes, explicit missing observations, stable track identifiers, and anomaly flags for unreliable entity counts.

RF-DETR Training Data. BasketEvent (Zhang et al., 2026b) provides frame-level player and ball boxes plus track-level event labels; only the boxes train the detector. Preserving the original partitions, we randomly select 3,000/200/200 train/validation/test videos and uniformly sample one frame per video. The resulting 3,400-image, two-class COCO dataset contains 28,266/2,069/1,856 boxes, respectively (32,191 total); the test set is held out for final evaluation.

Data Leakage Prevention. All *BasketballBench* clips belong to the BasketEvent test split and are therefore disjoint from every frame used for detector training or validation.

RF-DETR Training and Evaluation. RF-DETR Medium is initialized from its official COCO checkpoint, given a two-class head, and fully fine-tuned for 100 epochs on 576×576 inputs. Training uses batch size 16, four-step accumulation (effective 64), a learning rate of 10^{-4} for all non-encoder detector parameters (including the newly initialized classification head), and 1.5×10^{-4} for the DINOv2 encoder, with EMA decay 0.993 and seed 42. One NVIDIA RTX 3090 requires approximately four hours. The selected epoch-25 EMA checkpoint obtains 0.6409 validation mAP@50:95. On the 200-image test set, it reaches 0.6504 mAP@50:95, 0.8709 mAP@50, 0.7303 mAP@75, and 0.7489 AR@100. Mapping the official COCO model's *sports ball* and *person* classes yields 0.3882 mAP@50:95 under the same evaluation. Table B.1 gives the corresponding class-wise comparison.

Table B.1: Class-wise RF-DETR Medium detection performance on the held-out test set. Both metric columns report AP@50:95. For the original COCO model, *sports ball* and *person* are mapped to basketball and player, respectively.

Model	Basketball AP@50:95	Player AP@50:95
Original RF-DETR Medium	0.1734	0.6029
Fine-tuned RF-DETR Medium	**0.5122**	**0.7886**
Improvement	+0.3388	+0.1857

Information Exposed to the Agent. The controller receives the trajectory- and run-artifact references, player and ball counts, and anomaly flags, but no masks, service configuration, or source-game identifier.

B.2.2 Track-Level Event Detection

Implementation. Track-Level Event Detection applies BasketEvent's PlayNet (Zhang et al., 2026b) to video and participant trajectories, assigning track identifiers to made and missed shots,

free throws, fouls, turnovers, jump balls, rebounds, steals, blocks, and assists. Duplicate predictions for one action are reduced to the highest-confidence track and ordered by temporal midpoint. Internal timestamps support ordering only; exact moments require Event Temporal Localization.

Information Exposed to the Agent. Given the video and tracking artifact, the tool returns ordered track-level records with canonical event type, class label, confidence, and top-three alternatives.

B.2.3 SHOT-ZONE CLASSIFICATION

Implementation. Shot-Zone Classification uses an author-implemented six-class head added to PlayNet's pretrained PlayerEventModel. Given a made- or missed-shot track identified by Track-Level Event Detection, it predicts one of the six benchmark regions. It neither selects the shooter independently nor processes isolated free throws.

Training Data. The six-way dataset joins BasketEvent videos and trajectories with official PBP records. Each retained possession has one made or missed shot, an unambiguous shooter trajectory that overlaps the ball in at least one frame, and a valid court-area label. It contains 16,391 examples from 220 games: 13,966/206/2,219 examples from 185/2/33 games for train/validation/test. Class counts are 5,386 Above the Break 3, 4,077 Restricted Area, 3,290 In the Paint (Non-RA), 1,934 Mid-Range, 879 Left Corner 3, and 825 Right Corner 3, motivating class-balanced training.

Data Leakage Prevention. Because all benchmark clips come from the BasketEvent test split, they are disjoint from every shot-zone training and validation video.

Training and Evaluation. We add a six-class head to PlayNet's pretrained PlayerEventModel, reusing its TimeSformer backbone, trajectory-conditioned features, and relation modules. The event heads remain frozen; the new head and backbone are fine-tuned on 12 eight-frame clips per video at 4 fps and 224×224. Training runs for 20 epochs on four NVIDIA H800 GPUs with distributed data parallelism, one video per GPU, 16-step accumulation (effective batch 64), and AdamW. Head and backbone learning rates are 5×10^{-5} and 10^{-6}, with weight decay 0.05, gradient clipping 1.0, and seed 123. To address the training-set imbalance, weighted cross-entropy uses 0.1 label smoothing and the class weight

$$w_c = \mathrm{clip}\left(\frac{N}{6n_c}, 0.2, 5.0\right),$$

where n_c is the number of training examples in class c and $N = 13{,}966$. Table B.2 reports the exact frequencies and weights used in training.

Table B.2: Training-set class frequencies and loss weights for Shot-Zone Classification.

Class ID	Shot zone	Training examples	Weight
0	Above the Break 3	4,592	0.5069
1	Restricted Area	3,471	0.6706
2	Left Corner 3	755	3.083
3	Right Corner 3	695	3.3492
4	Mid-Range	1,602	1.453
5	In The Paint (Non-RA)	2,851	0.8164

Training takes approximately 8.75 hours. The checkpoint selected by validation macro-F1 (epoch 13) attains 0.7885 validation top-1 and 0.7660 macro F1. On 2,219 test examples, it obtains 0.7071/0.9148/0.9739 top-1/2/3 accuracy, 0.6813 macro precision, 0.7610 macro recall, 0.6935 macro F1, and 0.7100 weighted F1.

Information Exposed to the Agent. Given a video, tracking artifact, and shooter track, the tool returns the canonical region, class label, track, confidence, and, when available, the six-class distribution.

B.2.4 JERSEY RECOGNITION

Implementation. In video mode, Jersey Recognition uniformly samples eight observations from each requested track, crops the player with context, and applies a fixed jersey-reading instruction to Qwen3.5-9B (Qwen Team, 2026a). Multi-image mode applies the same recognizer to supplied crops. Responses are normalized to color, number, and confidence; invalid inputs are reported rather than imputed. Returning color with number disambiguates opponents sharing a jersey number and permits team mapping from question context.

Information Exposed to the Agent. For each track, the controller receives its identifier, color, number or null, and confidence, without access to the internal instruction or raw VLM output.

B.2.5 FACE RECOGNITION

Implementation. Face Recognition uses `face_recognition` (Geitgey, 2017) and an offline gallery of encoded official headshots. It returns the nearest identity only at distance ≤ 0.6, otherwise null; confidence is one minus the best distance, clipped to $[0, 1]$.

Information Exposed to the Agent. The controller receives only the matched full name and confidence; null denotes insufficient identity evidence.

B.2.6 STRUCTURED BASKETBALL KNOWLEDGE RETRIEVAL

Implementation. Structured Basketball Knowledge Retrieval provides fixed query families over the frozen `NBA_DB` (Section A.2), covering player profiles, biographies, statistics, career highs, game logs, rosters, team metadata, search, drafts, date-conditioned games, leaderboards, comparisons, box scores, summaries, and officials. It returns structured records and never arbitrary SQL. A year denotes the season start, competition defaults to regular season, and omitted filters broaden rather than silently select a result. Career aggregates combine all-season records; game details require a game identifier or unambiguous date–team combination.

Information Exposed to the Agent. The controller sees the query family, typed parameters, records, and `NBA_DB` source marker, but no credentials, schema, or answer mappings.

B.2.7 SCOREBUG READING

Implementation. Scorebug Reading uses Qwen3.5-9B to read away/home tricodes, scores, period, and remaining clock from an image or from the first and last video frames. A fixed instruction encodes the dataset's left-away/right-home convention and forbids guessing illegible fields; unreadable scorebugs return an error.

Information Exposed to the Agent. The controller selects image or video mode and receives typed fields plus a formatted summary, not general access to the VLM or its instruction.

B.2.8 EVENT TEMPORAL LOCALIZATION

Implementation. Event Temporal Localization uses Qwen3.5-9B for an already identified event; it is not a detector. At 2 fps, video-time mode locates the event relative to the first frame, while game-clock mode reads the period and remaining clock at that moment. Neither time system is arithmetically converted into the other, and a fixed prompt selects the output schema.

Information Exposed to the Agent. Given a video, known-event description, and mode, the controller receives a timestamp or period–clock pair with confidence and a formatted answer.

B.3 COMPOSITE SKILL LIBRARY

The controller loads one of four skills when its recurring evidence pattern matches the query.

Face-Conditioned Knowledge Retrieval. Face Recognition first identifies the player; Structured Basketball Knowledge Retrieval then obtains the requested fact. The instructions require visual identity evidence instead of an identity inferred from choices or prior knowledge.

Shot Analysis. Player-and-Ball Tracking and Track-Level Event Detection identify the shot and shooter, followed by Shot-Zone Classification for location or Jersey Recognition for identity using the supplied color–team mapping.

Identity-Grounded Play-by-Play Generation. Tracking and event detection produce an ordered sequence; one Jersey Recognition call resolves all participant tracks, which are joined back to events and serialized by team and jersey number.

Event-Conditioned Player Knowledge Retrieval. The skill specifies that an event participant must be grounded from jersey attributes and team context before querying that player's facts through Structured Basketball Knowledge Retrieval, to discourage premature player queries.

B.4 Complete Agent-Visible Prompts

This section reproduces the text made available to the controller. Prompt construction is dynamic: the fixed system template contains a rendered tool catalog and, when skills are installed, a skill catalog. We therefore present the exact template followed by the exact text used to fill those dynamic components. Tool-internal VLM instructions are not included here because they are not visible to the controller and cannot be edited or invoked as general prompts.

B.4.1 System-Prompt Template

The following is the complete fixed system-prompt template. At run time, `tool_list` and `skill_block` are replaced by the blocks shown in Sections B.4.2 and B.4.3, respectively.

```
You are a basketball analysis assistant. Based on the user's question
and available media (image/video/game context), call tools step by
step to gather evidence, then answer the question in a single text
message.

Available tools ({num_tools} total):
{tool_list}

Tool dependencies (important):
- detect_events / classify_shot_zone require tracks_path (produced by
track_entities)
- recognize_jersey has two modes: video+tracks mode (requires
tracks_path from track_entities) OR multi-image mode (media_paths of
pre-cropped player images, no tracks needed)
- classify_shot_zone requires shooter_player_id (identify the shooter
from detect_events' players list where event is Made Shot or Missed
Shot)
{skill_block}
Workflow:
0. If <available_skills> are listed above and one matches this task,
prefer calling the `skill` tool first and follow its instructions
before calling other tools.
1. Analyze the question and decide which tools to call and in what
order (guided by any loaded skill)
2. Call tools and observe the returned results
3. Based on the results, decide the next step: call more tools, or
give the final answer
4. Finally, answer the question in a plain text message (no more tool
calls)
```

```
Rules:
1. Tool use is optional: call tools only when they will add evidence
you actually need; you may answer directly from already-gathered
evidence or from your own knowledge when a question does not require
tool-based evidence. Tool arguments must be real values you know - do
not fabricate file paths
2. When referencing fields from a previous tool's result, use the
actual value (do not use ${{...}} placeholders)
3. Tools may fail or return errors; if a tool fails, adjust parameters
and retry or switch to another tool
4. Answer in the same language as the question (Chinese/English)
5. Answer only based on collected tool evidence (when tools were used)
- do not fabricate facts not present in the evidence
6. If evidence is insufficient to answer, state so honestly and
provide a reasonable fallback value matching the expected answer
format
7. Call at most 10 tools; if the limit is reached without sufficient
evidence, give the best answer with what you have
8. Do not call the same tool with the same arguments more than once -
repeated identical calls waste time and return identical results.
Reuse the result you already have
```

The initial user message is constructed as follows, with all media paths already anonymized:

```
Question: {question}
Media: {anonymized media description}
```

B.4.2 AGENT-VISIBLE TOOL PROMPT

The eight tools are rendered into `tool_list` in the following order. This text is supplied together with the corresponding function-calling schemas, which enforce the parameter types and required arguments described in Section B.2. Output schemas are not separately injected; the controller learns the output semantics from the descriptions below and from the returned observations.

```
- track_entities(video_path: string, backend: string) [required:
['video_path']]: Track basketball players and the ball through the
video. Returns tracks_path (string, path to the trajectory JSON
consumed by downstream tools), player_count (integer), ball_count
(integer, 0 or 1), anomalies (array of strings, e.g.
"too_many_players"), run_dir (string). The trajectory JSON keys players
as "player_NN" and the ball as "ball".
- detect_events(video_path: string, tracks_path: string) [required:
['video_path', 'tracks_path']]: Detect basketball events for each
tracked player. Returns players (array, one entry per non-blank player,
ordered by event time from earliest to latest). Each entry has
player_id (string, e.g. "player_01"), event (string: Made Shot/Missed
Shot/Free Throw/Foul/Turnover/Jump Ball/Rebound/steal/block/ast),
event_label (integer), confidence (float), top3 (array of [name, prob]
pairs).
- classify_shot_zone(video_path: string, tracks_path: string,
shooter_player_id: string) [required: ['video_path', 'tracks_path',
'shooter_player_id']]: Classify a shooter's shot attempt into one of
six court zones. Returns shooter_player_id (string), shot_location
(string: Restricted Area/In The Paint (Non-RA)/Above the Break 3/Left
Corner 3/Right Corner 3/Mid-Range), shot_location_label (integer),
confidence (float), probs (object mapping each zone name to its prob,
optional).
- recognize_jersey(video_path: string, tracks_path: string,
tracking_ids: array, media_paths: array) [required: []]: Recognize
jersey color and number for selected players from a video+tracks pair
```

```
or pre-cropped images. Returns players (array). Each entry has
tracking_id (string, e.g. "player_00"; "image_0" in multi-image mode),
jersey_color (string), jersey_number (integer or null), confidence
(string like "high"/"low" from cache, or numeric from VLM). Note:
jersey_number alone is NOT unique across the clip - opposing players
may share the same number; the (jersey_color, jersey_number) pair is
unique per player.
- recognize_face(image_path: string) [required: ['image_path']]:
Identify an NBA player from a face in a single image by matching
against precomputed face encodings. Returns name (string, the matched
player's full name; null when no stored face clears the tolerance) and
confidence (float in [0,1], computed as 1 - best_face_distance; 0.0
when no match).
- query_nba_database(query_type: string, player_name: string,
player_names: array, year: integer, season_type: string, team_alias:
string, keyword: string, college: string, position: string, date:
string, game_id: string, stat: string, top_n: integer, limit: integer,
offset: integer) [required: ['query_type']]: Query structured NBA
records (stats, rosters, games, drafts) from the local SQLite database.
Returns data (object whose shape varies by query_type) and source
(string, "NBA_DB"); on failure an "error" field is returned instead.
year is the season start year (2024 = 2024-25); season_type defaults to
REG. Most optional parameters BROADEN the result when omitted:
season_stats/game_log/team_roster without year return ALL seasons,
team_info without team_alias lists ALL teams. To get career-spanning
figures call season_stats once WITHOUT year and aggregate the returned
seasons list yourself; there is no separate career-total query type, so
do not re-query per season. Prefer one broad query over many narrow
ones.
- read_scorebug(image_path: string, video_path: string) [required: []]:
Read the broadcast scorebug from one image or a video's first+last
frames. Image input returns away_team, home_team (tricode strings),
away_score, home_score (integers), period (integer 1-4 or "OT"),
game_clock (string like "05:42"), answer (formatted summary),
input_type="image". Video input returns first_frame and last_frame
(objects with the above fields), answer (formatted "First: ... | Last:
..."), input_type="video". When the input has no readable scorebug,
returns an error ("no scorebug visible in the input") instead of null
fields.
- ground_event_time(video_path: string, event_description: string,
time_mode: string) [required: ['video_path', 'event_description']]: It
only localizes an event you already know happened. It CANNOT detect
events from the video. time_mode="video_time" returns timestamp_seconds
(float, seconds from video start), confidence (string), answer (string
like "[5.0]"), time_mode="video_time". time_mode="game_clock" returns
period (integer 1-4 or "OT"), game_clock (string read from the scorebug
at that instant), confidence (string), answer (string like "[3 -
05:42]"), time_mode="game_clock".
```

B.4.3 AGENT-VISIBLE SKILL PROMPTS

The skill catalog exposes the following repository names and descriptions. These map respectively to the paper-facing skills Shot Analysis, Event-Conditioned Player Knowledge Retrieval, Face-Conditioned Knowledge Retrieval, and Identity-Grounded Play-by-Play Generation.

```
analyze-shot-attempt: Analyze a basketball shot attempt in video to
determine its court zone or identify the shooter by team and jersey
number. Use for shot-zone classification and shooter identification
questions that require event detection followed by shot-specific
analysis.
answer-event-conditioned-player-knowledge: Answer an NBA knowledge
question about the player who performs a specified event in a
```

```
basketball video. Use when solving the question requires event-role
detection, jersey-based player resolution, and an NBA database lookup.
answer-face-conditioned-player-question: Identify a basketball player
from a face image and answer an NBA knowledge question about that
player. Use only when the question requires both face recognition and
an NBA database lookup; do not use for database-only, jersey-only, or
video-event questions.
generate-identity-grounded-play-by-play: Generate identity-grounded
basketball play-by-play from a raw possession video by orchestrating
entity tracking, event-role detection, jersey recognition, and
context-provided team mapping. Use when the output requires
chronological basketball events whose participants are represented by
on-court identities such as team and jersey number; do not use for
event labels, jersey recognition, or temporal localization alone.
```

When a skill is selected, its body is returned inside a `skill_content` wrapper. The following four blocks reproduce the full provided `SKILL.md` texts.

Shot Analysis prompt.

```
---
name: analyze-shot-attempt
description: Analyze a basketball shot attempt in video to determine
its court zone or identify the shooter by team and jersey number. Use
for shot-zone classification and shooter identification questions that
require event detection followed by shot-specific analysis.
---

# Analyze a Shot Attempt

## Workflow

1. Call `track_entities` on the video.
2. Pass the tracks to `detect_events`.
3. Select the relevant `Made Shot` or `Missed Shot` event and use its
shooter tracking ID.
4. Follow the branch required by the question:
   - For a shot-zone question, call `classify_shot_zone` with the
   shooter tracking ID.
   - For a shooter-identification question, call `recognize_jersey`
   for the shooter tracking ID.
5. For shooter identification, map the recognized jersey color to the
team mapping explicitly provided in the question, then combine the
team identifier with the jersey number.
6. Return exactly the requested label or option.

## Constraints

- Obtain the shooter role only from `detect_events`.
- Do not select a player by visual prominence or proximity to the
basket.
- Treat jersey identity as the pair `(jersey_color, jersey_number)`.
- Do not infer team identity from general NBA knowledge.
- Do not call `query_nba_database` unless the question explicitly
requests external player knowledge.
```

Event-Conditioned Player Knowledge Retrieval prompt.

```
---
name: answer-event-conditioned-player-knowledge
description: Answer an NBA knowledge question about the player who
performs a specified event in a basketball video. Use when solving the
question requires event-role detection, jersey-based player
resolution, and an NBA database lookup.
---

# Answer an Event-Conditioned Player Knowledge Question

## Workflow

1. Call `track_entities` on the video.
2. Pass the tracks to `detect_events`.
3. Select the requested event and participant role, such as shooter,
rebounder, fouler, or free-throw shooter.
4. Call `recognize_jersey` for that participant's tracking ID.
5. Map the jersey color to the team using only information supplied in
the question.
6. Query the team roster with `query_nba_database`.
7. Resolve the player using the pair `(team, jersey_number)`.
8. Query `query_nba_database` for the fact requested about the
resolved player.
9. Return exactly the requested answer format.

## Constraints

- Obtain the event role only from `detect_events`.
- Do not treat a jersey number alone as a unique player identity.
- Do not infer identity from answer choices, reputation, appearance,
or basketball knowledge.
- Do not query a player fact before resolving the player from team and
jersey number.
- Do not use `recognize_face` unless the question explicitly provides
a suitable face image.
- Stop rather than guess if the event participant, team, jersey
number, or roster identity cannot be resolved.
```

Face-Conditioned Knowledge Retrieval prompt.

```
---
name: answer-face-conditioned-player-question
description: Identify a basketball player from a face image and answer
an NBA knowledge question about that player. Use only when the
question requires both face recognition and an NBA database lookup; do
not use for database-only, jersey-only, or video-event questions.
---

# Answer a Face-Conditioned Player Question

## Workflow

1. Call `recognize_face` on the provided face image.
2. Treat the returned player name as the only valid identity evidence.
3. If no player is recognized, stop rather than infer identity from
the question or answer choices.
4. Call `query_nba_database` with the recognized name and the query
type appropriate for the requested fact.
5. Derive the answer only from the database result.
6. Return exactly the format requested by the user.

## Constraints

- Do not identify the player visually without `recognize_face`.
- Do not infer identity from answer choices.
- Do not substitute basketball knowledge for missing tool evidence.
- Do not call tracking, event, jersey, or shot-zone tools.
```

Identity-Grounded Play-by-Play Generation prompt.

```
---
name: generate-identity-grounded-play-by-play
description: Generate identity-grounded basketball play-by-play from a
raw possession video by orchestrating entity tracking, event-role
detection, jersey recognition, and context-provided team mapping. Use
when the output requires chronological basketball events whose
participants are represented by on-court identities such as team and
jersey number; do not use for event labels, jersey recognition, or
temporal localization alone.
---

# Generate Identity-Grounded Play-by-Play

Transform a raw possession video into chronological play-by-play whose
participants have on-court basketball identities.

## Workflow

1. Call `track_entities` on the raw video.
2. Pass the tracks to `detect_events`.
3. Treat the returned event types, roles, track IDs, and ordering as
authoritative.
4. Collect the unique, non-null track IDs assigned to event roles.
5. Call `recognize_jersey` once for those track IDs.
6. Join each jersey result to its event role strictly by track ID.
7. Map each jersey color to a team using only the mapping supplied in
the question.
8. Represent each participant using the required basketball identity,
normally `(team, jersey_number)`.
```

```
9. Serialize the identity-grounded events exactly according to the
requested play-by-play schema.

## Evidence Rules

- Obtain event types, roles, and ordering only from `detect_events`.
- Obtain jersey colors and numbers only from `recognize_jersey`.
- Treat `(jersey_color, jersey_number)` as the visual identity; a number
alone is not unique.
- Use question-provided context as the only source of the color-to-team
mapping.
- Reuse one recognized identity when the same track ID appears in
multiple events.

## Boundaries

- Do not independently detect, add, remove, merge, or reorder events.
- Do not infer a team from NBA knowledge or uniform familiarity.
- Do not infer a participant from answer choices.
- Do not resolve real player names unless the requested output
explicitly requires them.
- Do not include track IDs, confidence scores, or intermediate evidence
in the final play-by-play unless requested.
- Preserve uncertainty when a required participant identity cannot be
grounded.
```

C Additional Evaluation Results

We define the non-multiple-choice metrics, make the treatment of unparsable outputs explicit, and report fine-grained results. Multiple-choice tasks use option accuracy; other scores are percentages and temporal errors are seconds unless noted otherwise.

C.1 Evaluation Metrics

This subsection defines how parsing failures are treated for Scorebug Reading (Q4), Temporal Localization QA (Q7), and Play Event QA (Q8), the three non-multiple-choice tasks in BasketballBench.

Multiple-choice Parsing. We recompute option accuracy from the recorded responses with a fixed, reference-independent parser. It accepts an explicit answer option, a standalone option letter, a single answer-prefixed final paragraph, or an exact option text. Refusals and truncated reasoning without an explicit selection are unparsed and receive zero credit. Ordinary letters inside words and the English article "a" are not interpreted as option A. The same rules are applied to every model and both agent settings.

Parsed versus All-example Evaluation. For an accuracy-based task with N examples, let N_{parsed} be the number of outputs that can be mapped to the task schema and let N_{correct} be the number of correct parsed outputs. We distinguish

$$\text{ParseRate} = \frac{N_{\text{parsed}}}{N}, \qquad \text{ParsedAcc} = \frac{N_{\text{correct}}}{N_{\text{parsed}}}, \qquad \text{Acc} = \frac{N_{\text{correct}}}{N}. \tag{C.1}$$

Thus, $\text{Acc} = \text{ParseRate} \times \text{ParsedAcc}$, and unparsable, missing, or failed responses receive zero credit in Acc.

C.1.1 Scorebug Reading (Q4)

Scorebug Reading has equal score and clock subsets. A score prediction is parsed as (t_a, t_h, s_a, s_h), containing the away and home team tricodes and their scores. Team accuracy and score accuracy are slot-level measures:

$$\text{Acc}_{\text{team}} = \frac{\text{number of correct team slots}}{2N_s}, \qquad \text{Acc}_{\text{score}} = \frac{\text{number of correct score slots}}{2N_s}, \tag{C.2}$$

where N_s is the number of score prompts. Tricodes are uppercased and scores converted to integers before comparison.

A clock prediction is correct only when its period and normalized MM:SS both match. The primary Q4 metric, Full Exact Match, likewise requires all four score fields or both clock fields to match:

$$\mathrm{Acc}_{\mathrm{full}} = \frac{\sum_{i\in\mathcal{S}} \mathbf{1}[\hat{\mathbf{t}}_i = \mathbf{t}_i \wedge \hat{\mathbf{s}}_i = \mathbf{s}_i] + \sum_{i\in\mathcal{C}} \mathbf{1}[\hat{p}_i = p_i \wedge \hat{c}_i = c_i]}{|\mathcal{S}| + |\mathcal{C}|}. \tag{C.3}$$

This all-example Full Exact Match is the Q4 score in Table 2 of the main paper: an unparseable or missing response is counted as incorrect. The same convention is used for the three field-level scores in Table C.1: every field in an unparseable output is evaluated as false.

C.1.2 Temporal Localization QA (Q7)

Video-timestamp mode predicts elapsed clip seconds; game-clock mode predicts the period and remaining MM:SS. Acc@1s requires absolute error of at most one second and, for game-clock predictions, a matching period. Regulation periods match exactly, while all overtime labels are equivalent:

$$\mathrm{ParsedAcc@1s}_m = \frac{1}{N_m^{\mathrm{parsed}}} \sum_i \mathbf{1}[d_i \le 1], \qquad m \in \{\mathrm{video}, \mathrm{clock}\}, \tag{C.4}$$

with the period constraint included in the indicator for game-clock mode. The corresponding all-example score replaces the denominator by N_m, so unparsed outputs receive zero credit.

The primary Q7 metric reported in Table 2 is the macro-average of the all-example Acc@1s scores for the video-timestamp and game-clock modes:

$$\mathrm{Q7} = \frac{1}{2}\left(\mathrm{Acc@1s}_{\mathrm{video}} + \mathrm{Acc@1s}_{\mathrm{clock}}\right). \tag{C.5}$$

Each mode contains 790 examples, so this macro-average is equivalent to accuracy over all 1,580 Q7 questions. An output that remains missing or unparseable after normalization receives zero credit.

Table C.3 additionally reports Parsed Acc@1s, Parse Rate, and MAE. Video-timestamp MAE averages all parsed timestamps. Game-clock MAE averages remaining-time error only when the period matches; mismatched periods remain incorrect for Acc@1s but are excluded from MAE to avoid arbitrary cross-period conversion.

Period-Alias Normalization. The requested game-clock format uses a numeric regulation-period identifier, for example, `[1 - 06:53]`. We observe a recurrent parsing failure in which a model instead emits the semantically equivalent ordinal form `[1ST - 06:53]`. The clock value and period are both present, but a minor label-format mismatch can make a substantial fraction of predictions unparseable. For the fine-grained evaluation, we therefore normalize `1ST`/`2ND`/`3RD`/`4TH` to `1`/`2`/`3`/`4`. This rule is deterministic, applied identically to every model, defined solely from the output schema, and independent of the reference timestamp. It changes neither the predicted period nor the clock value. Values that change under this repair are marked with * in Table C.3.

C.1.3 Play Event QA (Q8)

LCS-based event F1. Each Q8 output is a chronologically ordered list of event records, where each event contains an `event_type` and a `participant`. We evaluate the predicted and reference sequences under two matching rules:

- **Type:** two events match if their event types are identical.
- **Type+Participant:** two events match only if both their event types and participants are identical.

For each rule, we use the longest common subsequence (LCS) to count correctly ordered matches. LCS gives partial credit for correctly predicted events while penalizing missing, additional, and incorrectly ordered events. For example, if the reference sequence is [shot, rebound, foul] and the prediction is [shot, foul], the LCS contains two matched events.

We sum the LCS match counts over all evaluation clips before computing micro-averaged precision, recall, and F1:

$$P_r = \frac{M_r}{N_{\text{pred}}}, \qquad R_r = \frac{M_r}{N_{\text{ref}}}, \qquad F1_r = \frac{2P_r R_r}{P_r + R_r},$$

where M_r is the total number of LCS matches under rule $r \in \{\text{Type}, \text{Type} + \text{Participant}\}$, and N_{pred} and N_{ref} are the total numbers of predicted and reference events. Metrics with a zero denominator are defined as zero.

The primary Q8 metric is $\text{F1}_{\text{Type+Participant}}$, which evaluates event extraction and participant grounding jointly. We additionally report F1_{Type} to isolate event-type recognition.

Participant Accuracy. Participant Accuracy (PA) measures participant identification after the event type has already been matched. Using the event pairs selected by the Type-only LCS alignment, PA is the fraction for which the predicted participant also matches the reference participant:

$$\text{PA} = \frac{\#\text{type-aligned pairs with the correct participant}}{\#\text{type-aligned event pairs}}.$$

PA does not directly penalize missing, additional, or incorrectly typed events and is therefore reported only as a diagnostic metric.

Conservative JSON Recovery. Some models produce recognizable event records in malformed JSON. We apply a deterministic, model-agnostic recovery procedure that extracts embedded JSON, removes trailing commas, completes only unambiguous closing delimiters, and wraps a root event list in the required `events` object. For truncated responses, only fully closed event records are retained. Recovery never changes predicted event types, participants, or event order and never consults the reference answer.

The Q8 metrics are computed after this recovery step. An output that remains invalid is treated as an empty predicted sequence. Models for which recovery changes at least one output are marked with * in Table C.4. The table additionally reports Type-only metrics, Participant Accuracy, and the post-recovery JSON validity rate.

C.2 Fine-Grained Results

This subsection presents a more fine-grained breakdown of the experimental results reported in Table 2 of the main paper.

C.2.1 Scorebug Reading (Q4)

Table C.1: Detailed Scorebug Reading results (%). Team and Score are all-example slot accuracies, and Clock requires an exact match of both the period and normalized clock value. An unparseable or missing output is evaluated as incorrect for every corresponding field.

Model	Team Acc.	Score Acc.	Clock Acc.
GPT-5.4	95.5	99.0	97.0
Claude Sonnet 5	94.5	99.0	97.0
Gemini 3.5 Flash	95.5	98.0	97.0
Qwen2.5-VL-7B	72.0	97.0	95.0
Qwen3.5-4B	76.5	98.0	91.0
Qwen3.5-9B	85.5	96.0	59.0
VideoLLaMA3-7B	42.0	91.0	95.0
InternVL3.5-8B	47.0	93.0	96.0
Molmo2-8B	55.5	90.5	92.0
BasketballAgent	89.0	98.0	97.0

Table C.1 reports the team identification, score recognition, and game-clock recognition performance of MLLMs and BasketballAgent on the Scorebug Reading task. Team identification exhibits

the lowest accuracy, primarily because scorebugs often represent teams using logos, which MLLMs struggle to associate with the correct teams. Parse Rates are generally high for Scorebug Reading, so Parsed Accuracy and all-example Accuracy are close for most models. We therefore report only the more conservative all-example field accuracies, for which an unparseable output is directly evaluated as incorrect.

C.2.2 Action Localization QA (Q5)

Table C.2: Row-normalized confusion matrix (%) for the final Skills-on BasketballAgent run on the 600 Action Localization QA questions, matching Table 2. Rows are ground-truth zones and columns are predictions; U denotes an unparsed response.

Ground truth	RA	Paint	ATB3	LC3	RC3	Mid	U
Restricted Area	**85.0**	6.0	2.0	1.0	0.0	6.0	0.0
Paint (Non-RA)	35.0	**44.0**	2.0	0.0	0.0	19.0	0.0
Above Break 3	1.0	0.0	**64.0**	15.0	4.0	16.0	0.0
Left Corner 3	4.0	2.0	1.0	**90.0**	0.0	3.0	0.0
Right Corner 3	9.0	1.0	1.0	3.0	**82.0**	4.0	0.0
Mid-Range	10.0	16.0	4.0	3.0	3.0	**64.0**	0.0

RA: Restricted Area; Paint: In The Paint (Non-RA); ATB3: Above the Break 3; LC3/RC3: Left/Right Corner 3; Mid: Mid-Range.

As shown in Table C.2, Left Corner Threes are most distinctive (90.0%), followed by Restricted Area (85.0%) and Right Corner Threes (82.0%), whereas In The Paint (Non-RA) is hardest (44.0%), chiefly confused with adjacent Restricted Area and Mid-Range regions.

C.2.3 Temporal Localization QA (Q7)

Table C.3: Detailed Temporal Localization QA (Q7) results using the corrected temporal annotations. P-Acc@1s denotes accuracy over parseable outputs, while Acc@1s is computed over all examples, treating an unparseable output as incorrect. Parse denotes the output parse rate, and MAE is measured in seconds. For game-clock predictions, Acc@1s additionally requires the predicted period to match the reference period; game-clock MAE is computed only over parsed predictions with the correct period. * indicates that the reported value differs from strict evaluation at the displayed precision after model-agnostic Period-Alias Normalization (`1ST`/`2ND`/`3RD`/`4TH` to `1`/`2`/`3`/`4`). The primary Q7 score in Table 2 is the macro-average of the two all-example Acc@1s columns. VideoLLaMA3-7B prediction records are unavailable in the supplied evaluation package, so its entries are left unverified (−−).

Model	Video Timestamp				Game Clock			
	P-Acc@1s	Acc@1s	Parse	MAE	P-Acc@1s	Acc@1s	Parse	MAE
GPT-5.4	69.2	69.2	100.0	1.67	86.2	85.7	99.4	1.52
Claude Sonnet 5	36.8	36.8	100.0	2.54	65.8*	65.7*	99.9*	11.44*
Gemini 3.5 Flash	57.6	55.9	97.2	1.30	74.5	70.5	94.7	2.91
Qwen2.5-VL-7B	1.1	1.1	100.0	5.96	45.0*	44.9*	99.9*	16.33*
Qwen3.5-4B	52.5	51.0	97.2	1.40	46.3*	45.8*	99.0*	30.55*
Qwen3.5-9B	48.3	22.9	47.5	1.31	71.8*	71.5*	99.6*	44.80*
VideoLLaMA3-7B	–	–	–	–	–	–	–	–
InternVL3.5-8B	1.1	1.1	100.0	9.72	25.3*	25.1*	99.0*	309.03*
Molmo2-8B	31.9	31.9	100.0	2.36	12.5	11.1	89.1	247.15
BasketballAgent	32.7	32.7	100.0	1.84	62.2	62.0	99.7	15.82

Table C.3 reports the metrics for Temporal Localization QA under two settings: video timestamp and game clock. The former requires the model to output the video timestamp at which the event

occurs, whereas the latter requires it to output the time displayed on the scorebug's game clock when the event occurs.

Period aliases account for a non-trivial share of nominal game-clock parsing failures. Period-Alias Normalization recovers 84 of 790 Qwen2.5-VL-7B outputs, raising its Parse Rate from 89.2% to 99.9% and its all-example Acc@1s from 39.1% to 44.9%. It additionally recovers 5 Claude Sonnet 5, 53 Qwen3.5-4B, 7 Qwen3.5-9B, and 78 InternVL3.5-8B outputs. Because the repair uses only a fixed equivalence between period-label spellings and never accesses the ground-truth time, the recovered scores more faithfully measure temporal localization rather than format compliance. All values that differ from strict evaluation at the reported precision are marked with *.

Qwen3.5-9B exhibits a distinct format-compliance failure in the video-timestamp setting. Although all 790 requests receive a response, 415 predictions specify a temporal interval, such as $[4.0, 4.5]$, rather than the requested single timestamp. Consequently, only 47.5% of its outputs are parseable, reducing its all-example Acc@1s to 22.9%. As a diagnostic analysis, among the 415 interval-valued predictions, selecting the interval start, midpoint, or end makes 143, 165, or 174 of them correct, respectively. Combined with the 181 correct point-valued predictions, these choices yield all-example Acc@1s values of 41.0%, 43.8%, and 44.9%, respectively. Treating an interval as correct whenever it overlaps the reference timestamp within the one-second tolerance gives 50.5%, close to Qwen3.5-4B's 51.0%. However, because no unique point-valued conversion is implied by an interval, we do not repair these outputs in the reported results. VideoLLaMA3-7B Q7 prediction records are unavailable in the supplied evaluation package, so its results remain unverified.

C.2.4 Play Event QA (Q8)

Table C.4: Detailed Play Event QA (Q8) results after Conservative JSON Recovery (%). Under *Type + Participant*, a match requires both the event type and the participant to be correct; under *Type*, only the event type must match. P, R, and F1 denote LCS-based micro-averaged precision, recall, and F1, respectively. PA denotes Participant Accuracy over event-type-aligned event pairs, and Parse denotes the post-recovery schema-validity rate. * indicates that Conservative JSON Recovery changes at least one output from the corresponding model. VideoLLaMA3-7B prediction records are unavailable in the supplied evaluation package; its entries are left unverified (−−).

Model	Type+Participant			Type			PA	Parse
	P	R	F1	P	R	F1		
GPT-5.4	26.1	34.8	29.8	52.2	**69.5**	59.6	46.7	99.9
Claude Sonnet 5	11.3	19.5	14.3	34.6	60.0	43.9	24.5	95.5
Gemini 3.5 Flash*	22.5	33.7	27.0	43.2	64.7	51.8	49.4	97.8
Qwen2.5-VL-7B*	1.3	2.1	1.6	22.7	34.9	27.5	5.1	**100.0**
Qwen3.5-4B*	6.5	11.1	8.2	31.0	53.0	39.1	20.6	**100.0**
Qwen3.5-9B*	8.0	14.8	10.4	35.3	65.2	45.8	22.2	99.9
VideoLLaMA3-7B	–	–	–	–	–	–	–	–
InternVL3.5-8B*	0.3	1.6	0.5	6.6	33.8	11.1	2.7	**100.0**
Molmo2-8B*	0.2	0.1	0.1	10.4	3.0	4.7	1.9	**100.0**
BasketballAgent*	**42.5**	**48.1**	**45.1**	**57.2**	64.5	**60.6**	**69.4**	**100.0**

Table C.4 presents detailed metrics for Play Event QA, showing that BasketballAgent substantially outperforms MLLMs on all key metrics except event-type recall. MLLMs achieve markedly higher precision, recall, and F1 scores when required to predict only event labels than when required to jointly identify events and their associated players. Without player identification, GPT achieves 98% of BasketballAgent's F1 score; when both events and players must be identified, this proportion drops to only 66%. These results highlight event-to-player association as a major weakness of current MLLMs.

Conservative JSON Recovery has its largest effect on the Qwen models. It recovers 11, 189, and 366 outputs for Qwen2.5-VL-7B, Qwen3.5-4B, and Qwen3.5-9B, respectively, increasing their Parse Rates from 98.9%, 81.1%, and 63.3% to 100.0%, 100.0%, and 99.9%. Their repaired all-example

F1 scores are 1.6%, 8.2%, and 10.4%, respectively. Recovery does not necessarily increase F1: making a previously invalid prediction visible to the evaluator can add both matched and unmatched events. The repair is therefore used to expose the model's predicted content, rather than to optimize its score; affected model names are marked with *.

C.3 Compositional Player-Pair Tasks

Motivation and construction. The ten core tasks do not directly test whether the agent can combine two player-identification and knowledge-retrieval paths in one answer. We therefore evaluate two player-pair tasks: A combines event roles within a video, while B combines image-based and video-based identity grounding. Each task contains 50 questions about the two players' listed heights or colleges. To prevent candidate answers from revealing attributes or allowing one resolved player to determine a multiple-choice answer, we remove all options and the answer-letter instruction from the initial pair questions. This conversion retains the original media, event roles, team–jersey-color context, and reference attributes; it does not resample the pair-task pool. Reference values are inherited from the two manually reviewed player records used to construct each pair, rather than recovered from an answer letter. The evaluation uses all 50 questions per task in both conditions, yielding 200 recorded runs.

Task A: Cross-Event Player Knowledge QA. Each question supplies one possession clip and requests the shooter first, followed by the assister after a made shot (34 questions) or the rebounder after a missed shot (16 questions). There are 28 height and 22 college questions, using 50 distinct videos. For example, an Indiana Pacers–Atlanta Hawks clip asks: *"Identify the player who made the shot and the player who assisted it. What are their listed heights, in that order?"* Its reference values are $6'8''$ and $6'7''$ (80 and 79 inches). Solving the task requires binding each event role to a track, recognizing the track's jersey attributes, resolving the player through the team roster, and retrieving the requested fact for both players.

Task B: Cross-Modal Player Knowledge QA. Each question supplies one player headshot and one possession clip, with the image player's attribute requested before the video player's attribute. The 50 questions use 50 distinct headshots and 50 distinct videos; 27 ask for height and 23 for college. The video roles are missed-shot shooter (19), made-shot shooter (10), free-throw shooter (9), rebounder (9), turnover player (2), and fouling player (1). For example, a headshot and a Denver Nuggets–Golden State Warriors clip ask for the colleges of the image player and the player who made the shot, in that order. The reference pair is UCLA and Boston College. The first identity is obtained through face recognition; the second requires event-role grounding, jersey recognition, and roster retrieval.

Inputs and scoring. Both conditions receive identical questions and media. Questions include the team names and jersey-color mapping, but no options, reference answers, target-player names, or player IDs. Media are passed separately from the question text. The output instruction requests exactly {"first": "<value>", "second": "<value>"}, without explanations or Markdown. Height answers are requested in feet-and-inches notation; equivalent representations such as 6-8 are normalized to total inches. College comparison ignores case, spaces, and punctuation, with aliases for USC, UConn, UCLA, and VCU and their full names. The primary metric is pair accuracy: both normalized values must match their respective references. Missing or unparseable values are incorrect, and the two positions cannot be freely exchanged. We additionally report single-position accuracy and strict format compliance. A response containing explanation plus an embedded valid JSON answer can receive semantic credit but fails the strict format check. All 200 recorded responses are parseable, and none has a recorded execution error.

Execution and existing skills. The controller is DeepSeek-V4-Flash at temperature zero; the local vision services use Qwen3.5-9B. Tracking, event detection, and jersey recognition reuse read-only runtime caches. Skills exposes the existing skill catalog and loading tool; No Skills removes both while retaining access to atomic tools. Table C.5 counts actual skill activations, not intended workflows or unexecuted proposals. No additional pair-specific skill is loaded in these traces. On A, event-conditioned retrieval provides the procedure for associating an event participant with a player and a database attribute, which the controller applies to a question with two target roles. Three runs

instead load identity-grounded play-by-play guidance, which also associates events with participant identities. On B, 24 runs jointly load face-conditioned and event-conditioned retrieval, and 12 jointly load face-conditioned retrieval and shot analysis. Eleven load only face-conditioned retrieval, two load only a video skill, and one loads no skill. Tools remain independently callable, so a skill load does not demonstrate complete adherence to its procedure.

Table C.5: Existing skills activated in the 50 Skills runs of each compositional task. Counts are per run and are not mutually exclusive.

Skill	A	B
Face-Conditioned Knowledge Retrieval	0	47
Event-Conditioned Player Knowledge Retrieval	47	25
Shot Analysis	1	13
Identity-Grounded Play-by-Play Generation	3	0

System results and auxiliary ablation. Table C.6 decomposes the results in Table 6, with the No-Skills condition included as an auxiliary control. The main evaluation concerns the full system's ability to answer additional question compositions. On A, the second attribute is correct less often than the first (54% versus 84% with Skills), consistent with the greater difficulty of resolving the assister or rebounder and retrieving that player's attribute. On B, both positions reach 82% with Skills. Pair correctness alone does not isolate identity errors from attribute-retrieval errors. For A, Skills and No Skills are both correct on 19 questions; Skills alone is correct on five, No Skills alone on one, and both fail on 25. For B, these counts are 31, one, zero, and 18, respectively. Two-sided exact McNemar tests give $p = 0.21875$ for A and $p = 1.0$ for B. Thus, the observed gains of eight and two percentage points are descriptive and do not establish a statistically significant accuracy improvement.

Table C.6: Detailed player-pair results. Pair, First, Second, and Format are percentages; Height and College give correct pairs over the corresponding question counts. Format requires a standalone JSON object with exactly the two requested fields.

Task	Setting	Pair	First	Second	Format	Height	College
A	No Skills	40	80	52	70	12/28	8/22
A	Skills	48	84	54	86	13/28	11/22
B	No Skills	62	80	76	52	17/27	14/23
B	Skills	64	82	82	58	18/27	14/23

Invocation overhead. We count dispatched atomic-tool calls, including failed calls and each member of a batch, but excluding skill loading and verifier-rejected proposals. The mean decreases by 11.97% on A (8.52 to 7.50) and 6.75% on B (8.00 to 7.46). These reductions do not establish lower total execution cost. Skills loads guidance 1.02 times per question on A and 1.70 on B; including these calls, the means are 8.52 versus 8.52 on A and 9.16 versus 8.00 on B (Skills versus No Skills). All recorded runs dispatch at most ten calls including skill loading. This is an observed maximum, not an independent verification of the configured budget. Tokens and latency are not controlled by counting invocations.

Media overlap and scope. Matching video identifiers and headshot identifiers in the execution records shows that 29 of the 50 videos in each task also occur in the ten core-task evaluations; all 50 headshots in B are reused from the core image evaluation. These are additional question compositions on partly reused media, rather than an unseen-media test. The records demonstrate reuse of existing named skills, but do not establish that the pair tasks were held out throughout skill development or that all prompts, tools, and database versions were frozen before their construction. We therefore interpret the results as evidence of procedural reuse under the evaluated controller, without claiming development-disjoint, unseen-video, cross-season, or cross-controller generalization.