Mind the (Language) Gap: Towards Probing Numerical and Cross-Lingual Limits of LVLMs

Somraj Gautam*, **Abhirama Subramanyam Penamakuri***,
Abhishek Bhandari, and Gaurav Harit
Indian Institute of Technology Jodhpur
{gautam.8,penamakuri.1,bhandari.1,gharit}@iitj.ac.in
https://huggingface.co/datasets/DIALab/MMCricBench

Abstract

We introduce MMCricBench-3K, a benchmark for Visual Question Answering (VQA) on cricket scorecards, designed to evaluate large vision-language models (LVLMs) on complex numerical and cross-lingual reasoning over semi-structured tabular images. MMCricBench-3K comprises 1,463 synthetically generated scorecard images from ODI, T20, and Test formats, accompanied by 1,500 English QA pairs. It includes two subsets: MMCricBench-E-1.5K, featuring English scorecards, and MMCricBench-H-1.5K, containing visually similar Hindi scorecards, with all questions and answers kept in English to enable controlled cross-script evaluation. The task demands reasoning over structured numerical data, multi-image context, and implicit domain knowledge. Empirical results show that even state-of-the-art LVLMs, such as GPT-4o and Qwen2.5VL, struggle on the English subset despite it being their primary training language and exhibit a further drop in performance on the Hindi subset. This reveals key limitations in structure-aware visual text understanding, numerical reasoning, and cross-lingual generalization. The dataset is publicly available via Hugging Face at https://huggingface.co/datasets/DIALab/MMCricBench, to promote LVLM research in this direction.

1 Introduction

Text-centric visual question answering (VQA) has seen considerable progress with benchmarks such as TextVQA (Singh et al., 2019b), ST-VQA (Xia et al., 2023), DocVQA (Mathew et al., 2021), VisualMRC (Tanaka et al., 2021), and OCR-Bench (Liu et al., 2024c), which evaluate models on tasks requiring OCR-based understanding and textual reasoning. More recently, tabular VQA datasets like TableVQA-Bench (Kim et al., 2024), TabComp (Gautam et al., 2025), and ComTQA (Zhao et al., 2024) have introduced structure-aware challenges focusing on numerical reasoning and table comprehension. However, as summarized in Table 1, these benchmarks often fall short in one or more dimensions: they are primarily monolingual (mostly English), lack multi-image contextual reasoning, and offer limited evaluation of fine-grained domain-specific numerical reasoning.

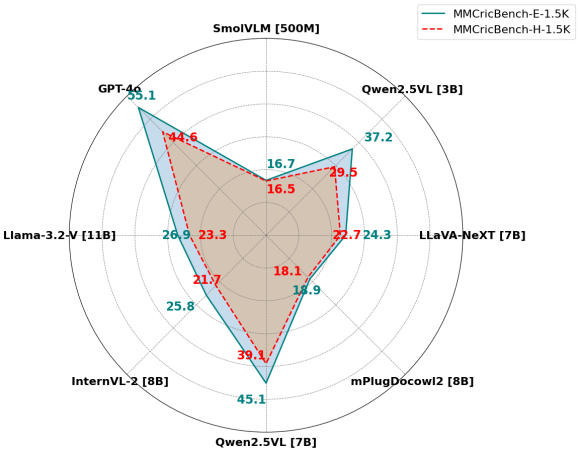


Figure 1: LVLM performance on MMCricBench-E-1.5K (English) and MMCricBench-H-1.5K (Hindi) cricket scorecards. While accuracy on English scorecards peaks at 55.1%, performance on visually similar Hindi scorecards remains consistently lower, highlighting a persistent gap in cross-lingual structure-aware numerical reasoning over images.

Cricket scorecard images, on the other hand, represent a compelling testbed for evaluating such capabilities. These semi-structured layouts combine tabular numeric data (runs, overs, wickets) with implicit contextual information (e.g., Which bowler has bowled the most wides in the match? Q3 in Figure 2), sometimes spanning across multiple images. In this work, we introduce MMCricBench-3K, a novel benchmark for visual question answering on cricket score-

*Equal contribution.

Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025), pages 568–584
November 8-9, 2025

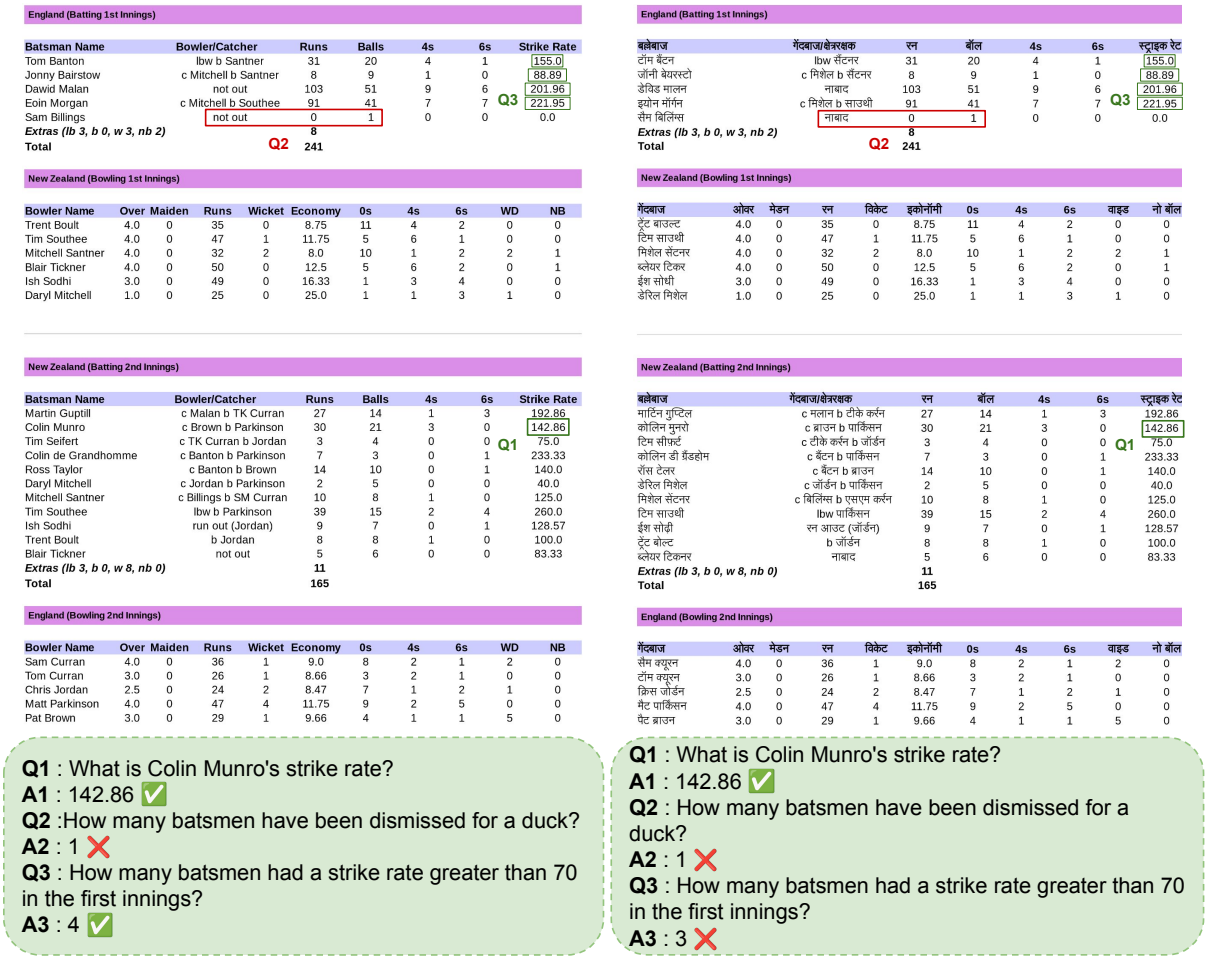


Figure 2: Examples of LVLMs (dis)parity between MMCRICBENCH-E-1.5K and MMCRICBENCH-H-1.5K. Example predictions by Qwen2.5VL-7B on English (left) and Hindi (right) scorecards. Q1 is a simple retrieval question, correctly answered in both cases. Q2 requires structure-aware, domain-specific reasoning, leading to failure in both. Q3 reveals a cross-lingual gap answered correctly on the English scorecard but incorrectly on the Hindi one, despite identical content.

cards, designed to evaluate the *structure-aware*, *mathematical*, *multi-image*, and *cross-lingual* reasoning capabilities of large vision-language models (LVLMs). MMCRICBENCH-3K comprises 1,463 synthetically generated scorecard images (822 single-image and 641 multi-image examples), along with 1,500 English QA pairs. It includes two subsets: MMCRICBENCH-E-1.5K (English scorecards) and MMCRICBENCH-H-1.5K (Hindi scorecards), with all questions and answers provided in English to enable controlled evaluation across script variations.

Large Vision-Language Models (LVLMs) (LLaVA-1.5 (Liu et al., 2024b), MiniGPT4 (Chen et al., 2023), mPLUG-Owl (Ye et al., 2024), Qwen-VL (Wang et al., 2024), and InternVL2 (Chen et al., 2024)) have become the de facto approaches for visual question answering tasks, including text-aware visual tasks (Penamakuri and Mishra, 2024a). Recent LVLMs such as Qwen2.5VL (Bai et al., 2025), mPLUG-DocOwl2 (Hu et al., 2024), InternVL2 (Chen et al., 2024), and TextMonkey (Liu et al., 2024d) have further advanced the bar on text-aware tasks, including VQA, by incorporating text-aware objectives into their pretraining or instruction-tuning stages. While studies exist to show strong performance of these models on English benchmarks, similar studies to understand their robustness across low-resource languages like Hindi[1] remains unexplored in the literature.

To this end, we leverage our MMCRICBENCH-H-1.5K benchmark to understand and evaluate cross-lingual mathematical reasoning abilities of

[1]Hindi as a textual language is not a low-resource language, however, when we look at visual text space, Hindi is a low-resource language.

LVLMs. Our experiments reveal a consistent performance drop when these LVLMs are evaluated on MMCricBench-H-1.5K (As illustrated in Figure 1), highlighting significant shortcomings in structure-aware, cross-lingual, and intensive numerical reasoning. Although advanced paradigms like Chain-of-Thought (CoT) prompting improve performance over naive variant, they still fall short compared to their performance on English scorecards.

In summary, our contributions are three-fold: (i) We introduce MMCricBench-3K, a novel structure-aware text-centric VQA benchmark to cover for the shortcomings of existing OCR and table-based VQA benchmarks by incorporating cross-lingual, multi-image, structure-aware, and numerically rich reasoning tasks grounded in the domain of cricket analytics. (ii) We comprehensively benchmark a range of leading LVLMs (open and closed-source) across different model sizes and show that they struggle on this benchmark, revealing key limitations in structure-aware visual understanding, numerical reasoning, and cross-lingual robustness. (iii) We conduct extensive ablations incorporating specialized components such as Optical Character Recognition (OCR), Table Structure Recognition (TSR), and advanced prompting strategies including Chain-of-Thought (CoT) reasoning. While these methods improve performance, they still fall short compared to the model's strong results on conventional text-centric benchmarks, highlighting the unique difficulty of our task.

2 MMCricBench-3K Dataset

We introduce MMCricBench-3K, a novel dataset designed to study a visual question answering (VQA) task on cricket scorecard images. Cricket scorecard images represent unstructured yet complex tabular images. VQA on such scorecards requires structural understanding, numerical data extraction, and implicit contextual reasoning across image(s). To the best of our knowledge, our work is the first principled work on studying VQA over cricket scorecard images. Specifically, we present two sub-benchmarks under MMCricBench-3K: MMCricBench-E-1.5K (with English scorecards) and MMCricBench-H-1.5K (with Hindi scorecards), with English question-answer annotations. This dataset is aimed at benchmarking the capabilities of Large Vision-Language Models (LVLMs) in performing cross-lingual deep mathematical reasoning over semi-structured content.

MMCricBench-3K consists of cricket scorecards sourced from various international game formats: ODI, T20, Test Match, and popular regional leagues: the Big Bash League (BBL, Australia) and the Indian Premier League (IPL, India). We provide carefully curated QA annotations to evaluate the numerical comprehension and deep mathematical reasoning abilities of LVLMs. Next, we explain the dataset curation pipeline.

Data Collection and Annotation: We begin to collect data for our benchmark by identifying publicly available datasets and repositories that contain cricket scorecard information. The initial dataset was obtained from Kaggle[2], which provides detailed cricket match statistics in CSV format. This dataset includes essential match statistics such as runs, wickets, and strike rates across different cricket formats (international game formats and regional leagues). Note that the data curated from the above-mentioned source does not contain scorecard images.

Scorecard Image Generation: We employed the open-source library Weasy Print[3] to convert CSV records into visually coherent scorecard tables. The generation process was inspired by design templates from various publicly accessible sports websites, ensuring diversity in fonts, styles, and table structures. We generated two distinct types of scorecard visualizations to support different VQA scenarios: (i) single-image scorecards for limited-overs formats (ODI, T20, and league matches) containing both innings in one comprehensive image, and (ii) multi-image scorecards for Test matches, where each image contains one inning, resulting in n images per match where n is the number of innings in the match. This dual approach allows us to evaluate both standard single-image VQA capabilities and more complex multi-image reasoning where models must synthesize information across multiple visual inputs. The multi-image format particularly challenges models to maintain contextual awareness and perform cross-referential numerical reasoning across separate visual sources. Each scorecard image contains semi-structured tabular information such as player names, runs, balls faced, boundaries, and

[2] https://www.kaggle.com/datasets/raghuvansht/cricket-scorecard-and-commentary-dataset

[3] https://weasyprint.org/

Benchmark	Reasoning type				QA Lang.	VT Lang.
	Cross-lingual	Multi-Image	Tabular	Mathematical		
Text-centric VQA						
TextVQA	✗	✗	✗	✗	English	English
ST-VQA	✗	✗	✗	✗	English	English
DocVQA	✗	✗	✓	✗	English	English
EST-VQA	✗	✗	✗	✗	English, Chinese	English, Chinese
VisualMRC	✗	✗	✓	✗	English	English
MTVQA	✗	✗	Not a major focus	✗	9 langauges	10 languages
OCRBench	✗	✗	Not a major focus	✗	English	English
Tabular VQA						
TableVQA-Bench	✗	✗	✓	✓	English	English
TabComp	✗	✗	✓	Not a major focus	English	English
ComTQA	✗	✗	✓	Not a major focus	English	English
Ours						
MMCricBench-E-1.5K	✓	✓	✓	✓	English	English
MMCricBench-H-1.5K	✓	✓	✓	✓	English	Hindi
MMCricBench-3K	✓	✓	✓	✓	English	English, Hindi

Table 1: Summary of Text-centric and Tabular VQA benchmarks, highlighting reasoning types and language support.

bowling figures, visually embedded in layouts typical of real-world cricket statistics. More details regarding the specific fonts, structural variations, and template designs are provided in the Appendix A.3.

Data Translation: To create images for MMCricBench-H-1.5K benchmark, we translate the English CSV records into Hindi using Google Translate[4] and follow a similar synthetic image generation procedure as of MMCricBench-E-1.5K. The MMCricBench-H-1.5K sub-benchmark is introduced to evaluate the cross-lingual semantic landscape of LVLMs associated with answering complex visual questions. Note that the translation process was performed at the cell level, ensuring that cricket-specific terms and numerical patterns remained consistent across languages. Additionally, we conducted a manual review of translated records to account for any potential translation inaccuracies and domain-specific inconsistencies.

Question-Answer Generation: We leverage cricket scorecard images to construct the MMCricBench-3K dataset, designed to evaluate Large Vision-Language Models (LVLMs) on structure-aware, cross-lingual, and numerically intensive reasoning tasks. We manually designed question templates and categorized them into three categories depending on the complexity associated with answering these questions. The categories are: **(i) Direct Retrieval & Simple Inference - C1:** In this task, questions target

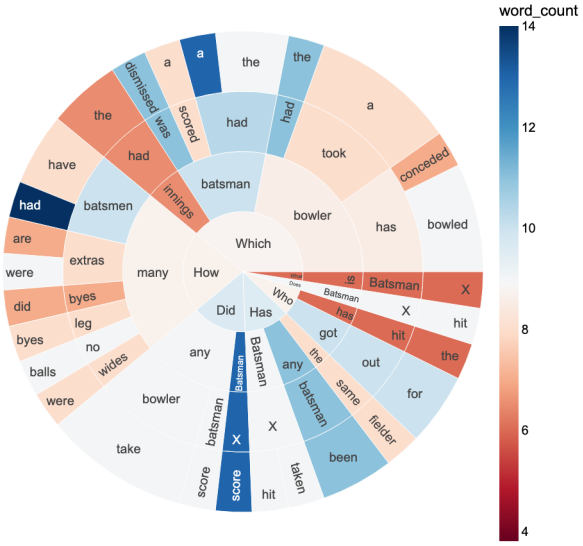


Figure 3: Distribution of questions based on their first few words in the dataset, illustrating common question prefixes after preprocessing.

direct information extraction from the scorecard image. For example, given an image containing the entry: *Aaron Finch | 89 | 52 | 4 | 7*, we generate a question: *"Who hit the most sixes?"* with the answer *"Aaron Finch"*. The model must read the image, interpret the tabular layout, understand the required context, and identify the relevant value. **(ii) Basic Arithmetic Reasoning & Conditional Logic - C2:** Here, questions require numerical reasoning based on arithmetic operations or conditional checks applied to one or more rows in the image. For example, from a scorecard showing: *Virat Kohli | 94 runs | 50 balls*, we generate the question: *"What is Virat Kohlis strike rate?"* with the answer *"188.0"*, computed

[4] https://translate.google.com

Category	Category Name	Example Question
C1	Direct Retrieval & Simple Inference	Which bowler has bowled the most wides in the match? Who got out for a duck in the first innings? Did any bowler take a 4-fer in the match? Has [Batsman X] taken more wickets than [Batsman Y]? Which bowler has conceded the most extras? Who has hit the maximum sixes? Does [Batsmax X] hit more sixes than [Batsman Y]? How many extras were bowled in the first innings?
C2	Basic Arithmetic Reasoning & Conditional Logic	What is [Batsman X] strike rate? Did [Batsman X] score better in the first innings or the second innings? Which batsman scored a century in the match? Which bowler took a 4-fer in the match? Has [Batsman X] hit more boundaries than [Batsman X]? Which batsman was dismissed for a golden duck in the match?
C3	Multi-step Reasoning & Quantitative Analysis	Which batsman had the highest strike rate (minimum 10 balls faced)? Which batsman had the highest boundary percentage? Which bowler had the better economy rate in the first innings? Which innings had the higher run rate? Which batsman had a strike rate greater than 70 in the first innings? Has the same fielder caught any batsman twice? Has any batsman been dismissed twice by the same bowler?

Table 2: Category and example questions. A full table containing statistics for each one of the single-image and multi-image questions is provided in the Appendix (Table A.4).

using: Strike Rate $= \frac{\text{Runs}}{\text{Balls}} \times 100$. The model must correctly localize relevant cells, extract values, and apply the correct reasoning. **(iii) Multi-step Reasoning & Quantitative Analysis - C3:** This task involves combining information across multiple players or sections in the scorecard. For instance, to answer: *"Who has the highest boundary percentage?"*, the model needs to compute $\frac{(4s \times 4 + 6s \times 6)}{\text{Total Runs}} \times 100$ for each player and select the maximum. This requires layout-aware text extraction, numerical computation, and multi-row comparison across the image.

Few question templates across the three categories are shown in Table 2. Detailed questions and statistics under all three categories are shown in Appendix A.4.

Answer Extraction via SQL: To ensure accuracy and consistency in answer generation, we used SQL queries to derive answers directly from the structured CSV data. This approach minimized manual errors and ensured the traceability of answers back to the original data. The SQL queries were formulated based on the question type and corresponding data structure. For instance:

- To retrieve highest boundary percentage:
```
SELECT Batsman_Name FROM batting WHERE Innings = 1 AND Balls > 0 ORDER BY ((([4s]*4 + [6s]*6) * 100.0 / Runs)) DESC LIMIT 1;
```
- To retrieve better economy rate in innings 1:
```
SELECT Bowler_Name, ROUND((SUM(Runs) * 1.0 / SUM(Over)), 2) AS Economy_Rate FROM bowling WHERE Innings = 1 GROUP BY Bowler_Name ORDER BY Economy_Rate ASC LIMIT 1;
```

SQL queries for every question template in MMCricBench-3K are shown in Table 12 in the Appendix. Further, the question-answer pairs are subjected to manual verification for possible factual and mathematical errors.

Further, we categorized answers into four categories, namely, (i) Binary (Yes/No), (ii) Numerical, (iii) Categorical (1/2/3/4 for innings-based questions), and (iv) Open-ended (Person names). Detailed statistics of MMCricBench-3K for are shown in the Figure 6 (a). Further, a selection of a few QA samples for each of the answer categories is shown in Table 9 in the Appendix.

3 Experiments

Baselines. We chose the VLMs from three selection criteria: **(a) VLMs with no OCR-aware tasks** during their pretraining or instruction tuning stages: LLaVA-Next (Liu et al., 2024a), and **(b) VLMs based on the size of their parameters**: *(i) Small VLMs (SVLMs)* with parameters less than 5B: SmolVLM-500M (Marafioti et al., 2025), Qwen2.5VL-2B (Wang et al., 2024), *(ii) Large*

Model [#params]	MMCricBench-E-1.5K C1	C2	C3	Avg.	MMCricBench-H-1.5K C1	C2	C3	Avg.	Σ ↑	Δ ↓
Open-source										
Small VLM (≤ 3B params)										
SmolVLM [500M]	19.5	21.6	15.9	19.2	20.4	12.9	24.3	19.0	19.1	0.2
Qwen2.5VL [3B]	38.7	40.1	41.7	40.2	39.8	24.5	**35.5**	33.3	36.8	6.9
Large VLM (params >3B and <10B)										
LLaVA-NeXT [7B]	40.2	10.8	**33.9**	28.3	35.7	10.8	33.3	26.6	27.4	1.7
mPlugDocowl2 [8B]	33.9	13.9	14.2	20.7	33.6	13.7	12.3	19.9	20.3	0.8
Qwen2.5VL [7B]	**64.6**	**52.1**	30.6	**49.1**	**62.7**	**39.8**	25.2	**42.6**	**45.8**	6.5
InternVL-2 [8B]	33.6	26.3	28.2	29.4	28.5	16.4	25.2	23.4	26.4	6.0
X-Large VLM (>10B)										
Llama-3.2-V [11B]	26.7	35.3	19.8	27.3	25.2	26.9	22.2	24.8	26.0	2.5
Closed-source										
GPT-4o	56.0	65.1	50.6	57.3	54.6	49.7	30.9	45.1	50.5	12.2

Table 3: Results on single-image questions split of MMCricBench-3K.

Method [#params]	MMCricBench-E-1.5K C1	C2	C3	Avg.	MMCricBench-H-1.5K C1	C2	C3	Avg.	Σ ↑	Δ ↓
LLMs+OCR										
Llama-3.2 [3B]	32.1	**31.4**	**22.8**	**28.8**	24.1	7.4	**18.3**	16.6	22.7	12.2
Qwen2.5 [3B]	**36.6**	**31.4**	16.5	28.2	**34.2**	**13.1**	13.5	**20.3**	**24.2**	7.9
VLMs Chain-of-Thought										
Qwen2.5VL [7B]	69.1	55.7	36.0	53.6	65.2	40.7	31.5	45.8	49.7	7.8

Table 4: Results on single-image of our ablation: LLMs+OCR vs VLMs on MMCricBench-3K.

VLMs (LVLMs) with parameters between 5B-14B: InternVL2-8B (Chen et al., 2024), Qwen2.5VL-7B (Bai et al., 2025), mPLUG-DocOwl2 (Hu et al., 2024), *(c) X-Large VLMs* with parameters greater than 14B: Llama-3.2-V-11B (Grattafiori et al., 2024) and (iii) **closed-source VLMs**: GPT-4o (OpenAI, 2024).

3.1 Result and Discussion

Performance of Open-Source Models Across Scales: Tables 3 and 5 present results for single-image and multi-image setups, revealing a consistent trend: model scale has a notable impact on performance across all question categories. Larger models generally outperform their smaller counterparts, with more pronounced gains on complex reasoning categories such as C2 (arithmetic) and C3 (multi-hop reasoning). For instance, Qwen2.5VL-7B (Bai et al., 2025) significantly outperforms its smaller 3B variant across all settings, with an average performance gap of 8.5 points. While this scaling advantage is particularly evident in higher-complexity tasks, the gains are less pronounced on simpler C1 (retrieval-based) questions, as expected.

Closed-Source vs Open-Source Models: Closed-source models, notably GPT-4o, consistently outperform open-source models across both the English (MMCricBench-E-1.5K) and Hindi (MMCricBench-H-1.5K) subsets. On single-image questions, GPT-4o achieves the highest average accuracy of 57.3% on English and 45.1% on Hindi, while in the multi-image setting, it scores 50.6% on English and 43.6% on Hindi. This reflects a clear cross-lingual drop of 12.2 and 7.0 points in the single- and multi-image settings, respectively. Although GPT-4o is not immune to the challenges posed by script variation, it still outperforms the closest open-source model Qwen2.5VL-7B by an average margin of 8.2 points across all tasks and subsets. These results highlight the robustness gap that remains between open and closed-source models, particularly in structured, cross-lingual VQA settings.

Comparison of cross-lingual capabilities: Models consistently exhibit a significant performance drop when transitioning from English to Hindi scorecards, particularly in categories requiring arithmetic reasoning (C2) and multi-step reasoning (C3). This decline highlights the limitations of cross-lingual generalization that scaling alone fails to address. For instance, GPT-4o, the strongest

Model [#params]	MMCricBench-E-1.5K C1	C2	C3	Avg.	MMCricBench-H-1.5K C1	C2	C3	Avg.	Σ ↑	Δ ↓
Open-source										
Small VLM (≤ 3B params)										
SmolVLM [500M]	14.4	10.8	10.2	11.8	20.0	6.0	9.0	11.6	11.7	0.2
Qwen2.5VL [3B]	34.1	35.3	24.1	31.2	27.5	19.8	18.7	22.0	26.6	9.2
Large VLM (params >3B and <10B)										
LLaVA-NeXT [7B]	27.5	6.6	14.4	16.2	24.5	5.4	14.5	14.8	15.5	1.4
mPlugDocowl2 [8B]	24.7	7.5	13.2	15.2	23.3	7.1	12.6	14.4	14.8	0.8
Qwen2.5VL [7B]	**41.9**	**41.9**	27.1	**37.0**	**37.7**	**33.5**	**25.3**	**32.2**	**34.6**	4.8
InternVL-2 [8B]	29.3	5.4	21.1	18.6	28.1	4.8	21.7	18.2	18.4	0.4
X-Large VLM (>10B)										
Llama-3.2-V [11B]	34.7	14.3	**29.5**	26.2	29.3	11.3	20.4	20.4	23.3	5.8
Closed-source										
GPT-4o	50.3	61.1	40.4	50.6	39.5	53.8	37.3	43.6	47.1	7.0

Table 5: Results on multi-image questions split of MMCricBench-3K.

Method [#params]	MMCricBench-E-1.5K C1	C2	C3	Avg.	MMCricBench-H-1.5K C1	C2	C3	Avg.	Σ ↑	Δ ↓
LLMs+OCR										
Llama-3.2 [3B]	24.5	17.9	25.9	22.8	18.5	1.8	14.4	11.6	17.2	11.2
Qwen2.5 [3B]	**30.5**	**24.5**	**27.7**	**27.6**	**23.3**	**10.7**	**22.8**	**19.0**	**23.3**	8.6
VLMs Chain-of-Thought										
Qwen2.5VL [7B]	40.7	40.7	22.9	34.8	36.5	29.9	23.5	30.0	32.4	4.8

Table 6: Results on multi-image of our ablation: LLMs+OCR vs VLMs on MMCricBench-3K.

overall performer shows a substantial drop of 12.2 points (single-image) and 7.0 points (multi-image) on average when evaluated on Hindi scorecards. Similarly, Qwen2.5VL-7B experiences a 6.5 to 6.9 point decrease across both subsets. These degradations indicate that even state-of-the-art models with strong English capabilities are not robust to script variation in visually embedded text. Our findings suggest that effective VQA on cricket scorecards requires a combination of table structure understanding, OCR, and visual text groundingcapabilities that current models struggle to achieve in non-Latin scripts and low-resource visual text languages like Hindi.

3.1.1 Ablations

LLMs + OCR: To isolate the role of visual perception in scorecard-based VQA, we evaluate a baseline that combines OCR with text-only large language models (LLMs). Specifically, we extract text from scorecard images using the Tesseract OCR engine (Smith, 2007) and feed the output into two LLMs: LLaMA-3.2-3B (Dubey et al., 2024) and Qwen2.5-3B (Yang et al., 2024). This setting evaluates whether textual cues alone are sufficient to reason over cricket scorecards. As shown in Tables 4 and 6, both models perform significantly worse than vision-language models. On average across MMCricBench-3K, LLaMA-3.2-3B exhibits a performance drop of 4.7%, while Qwen2.5-3B shows a much larger drop of 16.5% compared to their vision counterparts. These results highlight the limitations of OCR+LLM pipelines: despite having access to textual input, these models struggle to capture structural cues such as column alignment and row grouping that are essential for tabular reasoning. The English-Hindi gap remains wide, showing that OCR-based pipelines struggle in cross-lingual, visually complex settings.

CoT Prompting vs. Regular Prompting: Applying Chain-of-Thought (CoT) prompting to Qwen2.5VL-7B improves overall performance in the single-image setting, with accuracy increasing from 45.8% to 49.7%. This gain is especially notable in reasoning-heavy categories such as arithmetic (C2) and multi-step (C3), indicating that CoT helps the model decompose complex queries into interpretable steps. However, in the multi-image setting, overall performance drops slightly from 34.6% to 32.4%, suggesting that CoT may not transfer well when reasoning must span multiple visual contexts. While CoT improves reason-

ing behaviour, the cross-lingual gap still remains.

4 Comparison with Related Work

LVLMs for VQA over text images: Recent advancements of large vision-language models (LVLMs) have transformed visual question-answering (VQA) tasks into gaining impressive zero-shot performance across diverse scenarios (OpenAI, 2024; Yang et al., 2024; Chen et al., 2024; Liu et al., 2024a) including text-centric VQA. On these lines, DocPedia (Feng et al., 2024) processes high-resolution inputs without increasing token sequence length. mPLUG-DocOwl(Ye et al., 2024), Qwen2-VL (Wang et al., 2024), and TextMonkey (Liu et al., 2024d) further leverage publicly available document VQA datasets to boost text performance. Extensions of the LLaVA (Liu et al., 2024b) framework such as LLaVAR (Zhang et al., 2023), InternVL (Chen et al., 2024), KaLMA (Penamakuri and Mishra, 2024b) and UniDoc (Feng et al., 2023) have broadened LVLM capabilities in visual text by leveraging both textual content and visual content, thereby setting a new benchmark for text-centric VQA including their knowledge-aware counterparts (e.g. TextKVQA (Singh et al., 2019a)). Despite these significant strides, LVLMs fall short in complex tasks like MMCricBench-3K as discussed in Section 3.

Text-centric VQA: The existing text VQA datasets TextVQA (Singh et al., 2019b), ST-VQA (Biten et al., 2019), DocVQA (Mathew et al., 2021), and VisualMRC (Tanaka et al., 2021) solely focus on the English language. While EST-VQA (Wang et al., 2020) and MTVQA (Tang et al., 2024) are multilingual, they do not cover low-resource visual languages e.g. Hindi. Further, existing datasets either primarily focus on single-image QA or lack questions that require structure-aware mathematical reasoning (summarized in Table 1). We aim to address this gap.

Models and Datasets for Table VQA: While benchmark datasets like TableVQA-Bench (Kim et al., 2024), TabComp (Gautam et al., 2025), and ComTQA (Zhao et al., 2024) exist for VQA over table images, they are all English-focused with answers directly in the images. However, table image datasets to evaluate the cross-lingual mathematical reasoning capabilities of LVLMs remain underexplored.

Multi-image VQA: Several benchmarks (Talmor et al., 2021; Mathew et al., 2021; Bansal et al., 2020; Chang et al., 2022; Penamakuri et al., 2023; Wu et al., 2025) explore reasoning across multiple image. However, these tasks largely overlook structure-aware tabular understanding, numerical reasoning, and cross-lingual robustness, which are central to our setting. In contrast, we include a dedicated multi-image subset within MMCricBench-3K, where answering a question requires aggregating statistics across multiple images representing different innings of a match, thereby combining tabular, numerical, and cross-lingual reasoning.

Table Reasoning Ability of LLMs: LLMs and multimodal LLMs (MLLMs) are evaluated in (Deng et al., 2024) using tables presented as either text or images, finding that text-based representations yield better results, while image-based table reasoning remains weak for current models. To enhance reasoning, (Lu et al., 2024) introduced TART, a tool-augmented framework that enables step-by-step table question answering by integrating LLMs with symbolic tools. Similarly, (Nahid and Rafiei, 2024) proposed TabSQLify, which improves efficiency by decomposing large tables into smaller, relevant segments using text-to-SQL conversion. Furthermore, (Zhao et al., 2022) proposed ReasTAP, a pretraining strategy using synthetic table reasoning examples to inject structured reasoning ability into LLMs. Despite these advances, most research focuses on structured tables in English. A critical gap remains in (i) table reasoning in low-resource languages (such as Hindi in our dataset), particularly visually complex, domain-specific formats like cricket scorecards, (ii) evaluating multi-step reasoning and conditional logic in understanding the tabular content. Our work addresses this need by introducing MMCricBench-H-1.5K, a benchmark designed to push the limits of visual-text reasoning in Hindi.

5 Conclusion

We presented MMCricBench-3K, a novel benchmark for VQA on cricket scorecards that addresses critical gaps in existing datasets by incorporating cross-lingual understanding, multi-image reasoning, and domain-specific numerical analysis. Our evaluation across MMCricBench-E-1.5K (English) and MMCricBench-H-1.5K (Hindi) scorecards reveals a significant perfor-

mance disparity among state-of-the-art LVLMs. While these models show reasonable proficiency with English scorecards, they struggle substantially with Hindi variants despite identical information content. Even advanced prompting strategies like CoT fail to bridge this performance gap. These findings highlight a critical weakness in cross-lingual visual reasoning capabilities, highlighting the need for more robust models that can effectively process structured numerical data across language boundaries. As AI applications expand globally, addressing these limitations becomes increasingly crucial. MMCricBench-3K provides researchers with a challenging testbed for advancing LVLM capabilities beyond English-centric contexts, particularly in domains requiring precise analysis of semi-structured information.

6 Limitations

Despite the strengths of MMCricBench-3K in evaluating structure-aware and cross-lingual visual question answering, several limitations persist. First, the dataset's linguistic scope is limited to English and Hindi, leaving out other regional scripts and languages prevalent in cricket contexts. Second, the use of synthetically generated scorecards, while visually coherent, may not fully capture the complexity and noise present in real-world documents.

Ethical Considerations

Our benchmark, MMCricBench-3K, is synthetically generated using publicly available cricket statistics, with no private or sensitive personal information involved. All scorecard data is derived from open datasets (e.g., Kaggle) and only includes publicly known player names and match events. We translate content using automated tools (e.g., Google Translate), and manually verify for correctness to minimize cultural or linguistic bias.

While our dataset uses Hindi as a representative low-resource script for cross-lingual evaluation, we acknowledge the limitations of focusing only on English-Hindi and encourage future extensions to other regional languages and scripts. Additionally, though we simulate realistic scorecards, real-world images may include noise, varied layouts, or OCR artifacts that are not fully captured in our synthetic setup. Our work aims to support fair and inclusive evaluation of vision-language models in global contexts. No human annotators were subjected to sensitive or harmful content during data creation, and no demographic or identity information is used or inferred in this study.

Acknowledgments

Abhirama is deeply grateful to his PhD advisor, Dr. Anand Mishra, for his unwavering guidance and mentorship throughout his PhD journey, which has profoundly shaped his research outlook and provided the foundation that enabled this work. Abhirama is supported by the Prime Ministers Research Fellowship (PMRF), Ministry of Education, Government of India.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Ankan Bansal, Yuting Zhang, and Rama Chellappa. 2020. Visual question answering on image sets. In *ECCV*, pages 51–67.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. 2019. Scene text visual question answering. In *Proceedings of the ICCV*, pages 4291–4301.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. In *CVPR*, pages 16495–16504.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*.

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. 2024. Tables as texts or images: Evaluating the table reasoning ability of llms and mllms. *arXiv preprint arXiv:2402.12424*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hao Feng, Qi Liu, Hao Liu, Jingqun Tang, Wengang Zhou, Houqiang Li, and Can Huang. 2024. Docpedia: Unleashing the power of large multimodal model in the frequency domain for versatile document understanding. *Science China Information Sciences*, 67(12):1–14.

Hao Feng, Zijian Wang, Jingqun Tang, Jinghui Lu, Wengang Zhou, Houqiang Li, and Can Huang. 2023. Unidoc: A universal large multimodal model for simultaneous text detection, recognition, spotting and understanding. *arXiv preprint arXiv:2308.11592*.

Somraj Gautam, Abhishek Bhandari, and Gaurav Harit. 2025. Tabcomp: A dataset for visual table reading comprehension. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5773–5780.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. 2024. mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. *arXiv preprint arXiv:2409.03420*.

Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. 2024. Tablevqa-bench: A visual question answering benchmark on multiple table domains. *arXiv preprint arXiv:2404.19205*.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

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

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. 2024c. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102.

Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. 2024d. Textmonkey: An ocr-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*.

Xinyuan Lu, Liangming Pan, Yubo Ma, Preslav Nakov, and Min-Yen Kan. 2024. Tart: An open-source tool-augmented framework for explainable table-based reasoning. *arXiv preprint arXiv:2409.11724*.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, et al. 2025. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the WACV*, pages 2200–2209.

Md Mahadi Hasan Nahid and Davood Rafiei. 2024. Normtab: Improving symbolic reasoning in llms through tabular data normalization. *arXiv preprint arXiv:2406.17961*.

OpenAI. 2024. Gpt-4 api documentation. OpenAI API Documentation. Accessed: 2024-02-16.

Abhirama Subramanyam Penamakuri, Manish Gupta, Mithun Das Gupta, and Anand Mishra. 2023. Answer mining from a pool of images: towards retrieval-based visual question answering. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 1312–1321.

Abhirama Subramanyam Penamakuri and Anand Mishra. 2024a. Visual text matters: Improving text-KVQA with visual text entity knowledge-aware large multimodal assistant. In *EMNLP*.

Abhirama Subramanyam Penamakuri and Anand Mishra. 2024b. Visual text matters: Improving text-KVQA with visual text entity knowledge-aware large multimodal assistant. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20675–20688. Association for Computational Linguistics.

Ajeet Kumar Singh, Anand Mishra, Shashank Shekhar, and Anirban Chakraborty. 2019a. From strings to things: Knowledge-enabled VQA model that can read and reason. In *ICCV*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019b. Towards vqa models that can read. In *Proceedings of the CVPR*, pages 8317–8326.

Ray Smith. 2007. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. Multimodalqa: complex question answering over text, tables and images. In *ICLR (Poster)*.

Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. Visualmrc: Machine reading comprehension on document images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13878–13888.

Jingqun Tang, Qi Liu, Yongjie Ye, Jinghui Lu, Shu Wei, Chunhui Lin, Wanqing Li, Mohamad Fitri Faiz Bin Mahmood, Hao Feng, Zhen Zhao, et al. 2024. Mtvqa: Benchmarking multilingual text-centric visual question answering. *arXiv preprint arXiv:2405.11985*.

TensorDock Inc. 2024. Tensordock: Gpu cloud computing platform. Cloud Computing Service.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. 2020. On the general value of evidence, and bilingual scene-text visual question answering. In *Proceedings of the CVPR*, pages 10126–10135.

Tsung-Han Wu, Giscard Biamby, Jerome Quenum, Ritwik Gupta, Joseph E Gonzalez, Trevor Darrell, and David Chan. 2025. Visual haystacks: A vision-centric needle-in-a-haystack benchmark. In *The Thirteenth International Conference on Learning Representations*.

Haiying Xia, Richeng Lan, Haisheng Li, and Shuxiang Song. 2023. St-vqa: shrinkage transformer with accurate alignment for visual question answering. *Applied Intelligence*, 53(18):20967–20978.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. 2024. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *CVPR*.

Yanzhe Zhang, Ruiyi Zhang, Jiuxiang Gu, Yufan Zhou, Nedim Lipka, Diyi Yang, and Tong Sun. 2023. Llavar: Enhanced visual instruction tuning for text-rich image understanding. *arXiv preprint arXiv:2306.17107*.

Weichao Zhao, Hao Feng, Qi Liu, Jingqun Tang, Binghong Wu, Lei Liao, Shu Wei, Yongjie Ye, Hao Liu, Wengang Zhou, et al. 2024. Tabpedia: Towards comprehensive visual table understanding with concept synergy. *Advances in Neural Information Processing Systems*, 37:7185–7212.

Yilun Zhao, Linyong Nan, Zhenting Qi, Rui Zhang, and Dragomir Radev. 2022. Reastap: Injecting table reasoning skills during pre-training via synthetic reasoning examples. *arXiv preprint arXiv:2210.12374*.

A Appendix

A.1 Implementation Details

We conduct all of our experiments on the baseline VLMs in a zero-shot setting, with their default setting provided in their respective implementations. When prompted with these methods, we faced two challenges: (i) verbose answers and (ii) digits written in text, e.g. Fifth in place of 5. To overcome these challenges and generate precise and concise answers, we added a brief instruction to the prompt: 'Answer precisely in 1-2 words, answer in digits when required' before the main question. We conducted all our experiments on a cloud machine with 3 A6000 Nvidia GPUs (48 GB each) rented from online cloud GPU provider TensorDock (TensorDock Inc., 2024).

A.2 Cricket Specific Terms and Their Calculations

In cricket, performance metrics help quantify a player's efficiency in both batting and bowling. Two key metrics are the Strike Rate and the Economy Rate. The following explanations and formulas provide a detailed understanding of these terms.

A.2.1 Strike Rate

The Strike Rate is primarily used to measure a batsman's scoring efficiency. It represents the average number of runs scored per 100 balls faced, indicating how quickly a batsman can accumulate runs.

Calculation: The basic formula for Strike Rate is:

$$\text{Strike Rate (SR)} = \frac{\text{Total Runs Scored}}{\text{Total Balls Faced}} \times 100$$

Example: For instance, if a batsman scores 50 runs from 40 balls, the Strike Rate is calculated as:

$$\text{SR} = \frac{50}{40} \times 100 = 125$$

This means that, on average, the batsman scores 125 runs for every 100 balls faced. A higher strike rate reflects a more aggressive and effective scoring approach.

A.2.2 Economy Rate

The Economy Rate measures a bowlers efficiency by calculating the average number of runs conceded per over. An over in cricket typically consists of 6 legal deliveries.

Calculation: The basic formula for Economy Rate is:

$$\text{Economy Rate (Econ)} = \frac{\text{Total Runs Conceded}}{\text{Total Overs Bowled}}$$

If the data is provided in terms of balls bowled rather than overs, the formula is adjusted by converting balls to overs:

$$\text{Economy Rate (Econ)} = \frac{\text{Total Runs Conceded}}{\text{Total Balls Bowled}} \times 6$$

Example: Consider a bowler who concedes 30 runs in 10 overs. The Economy Rate is:

$$\text{Econ} = \frac{30}{10} = 3.0 \text{ runs per over}$$

A lower economy rate suggests that the bowler is effective at limiting the opposing team's scoring.

A.2.3 Summary

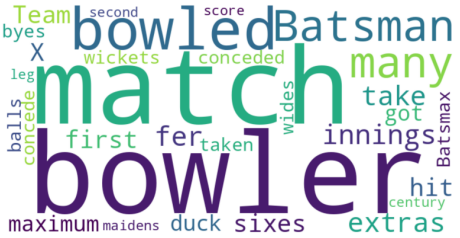
Understanding these calculations is fundamental for analyzing cricket performance:

- The Strike Rate provides insight into a batsman's ability to score quickly, which is especially valuable in limited-overs formats.
- The Economy Rate evaluates a bowler's performance by highlighting how few runs they allow per over, thus reflecting their effectiveness in containing the opposition's scoring.

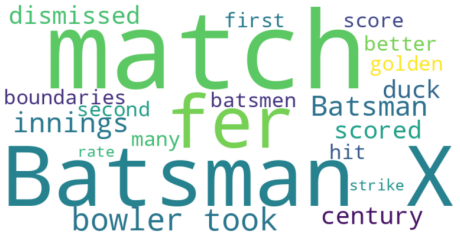
These metrics are essential for comparing player performances across different matches and cricket formats, offering a standardized way to assess and discuss efficiency in both batting and bowling.

A.2.4 Rationale for Multi-Images

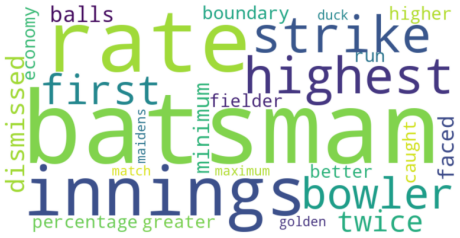
Cricket matches played over multiple innings often contain statistics that span beyond a single table or image. For instance, Test matches commonly have four innings across five days, with runs, wickets, and partnerships distributed across these innings. A single static image may not encapsulate the full statistical narrative, necessitating a shift toward a multi-image structure. LVLMs must then establish logical connections across these images to accurately answer questions involving cumulative statistics or cross-inning performance comparisons.



(a) C1



(b) C2



(c) C3

Figure 4: Word cloud of category-wise questions.

A.3 Scorecard Image Template Design

The HTML/CSS template used to render each match's batting and bowling scorecards is defined via a Jinja2 template and styled to ensure consistent layout and visual separation of sections. Images are generated by rendering this HTML to PDF via WeasyPrint, converting singlepage PDFs to 300 DPI PNGs, and cropping whitespace. Below, we list all key design parameters in Table 8.

A.3.1 Country Diversity

Cricket is a global sport played across continents, with scorecards reflecting diverse naming conventions, team compositions, and performance statistics. Incorporating scorecards from 13 countries ensures that the dataset captures these variations,

Country
Afghanistan
Australia
Bangladesh
England
India
Ireland
New Zealand
Pakistan
South Africa
Sri Lanka
West Indies
Zimbabwe
Netherlands

Table 7: Distribution of Scorecards Across 13 Countries.

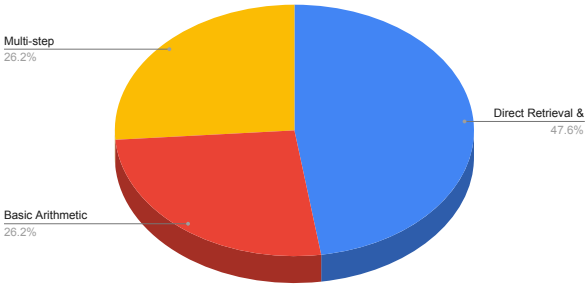


Figure 5: Category Distribution for Batting and Bowling Questions.

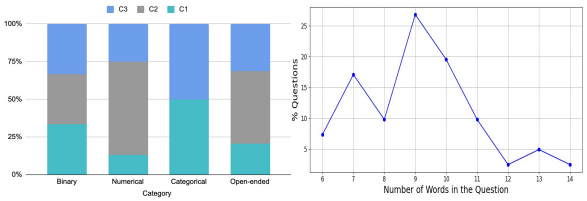


Figure 6: MMCricBench-3K questions and answers analysis: (a) Answer distribution over various question categories, (b) Distribution of the number of words across questions.

providing a comprehensive representation of cricketing data across different regions and formats. The country list is in table 7.

A.4 Remaining set of questions and their category

Table A.4 containing the remaining set of questions and their categories.

A.5 SQL Query for extracting answers

Table 12 contains questions and SQL queries used for getting answers.

Category	Ingredient	Specification
Setup	Page setup & font	20 px margins; white background; Arial, sans-serif font.
	Table layout	Full-width tables; collapsed borders; centered; 20 px vertical margins.
	Cell padding	1 px padding on all header and data cells.
Styling	Header row styling	Custom background color; black text; 14 px font; left-aligned.
	Data cell styling	14 px font; centered text.
	Column widths	First column left-aligned (min-width 120 px); others centered (min-width 60 px).
	Team-name banner	Bold 12 px text on customizable background; 5 px padding and vertical margins.
	Section separation	1 px bottom border + extra spacing between innings.
	Special rows	Bold white rows for Extras and Total.
Color variants	Variant 1	Banner #DA8EE7; header #CCCCFF.
	Variant 2	Banner #E8CCFF; header #CCE7FF.
	Variant 3	Banner #D0CCFF; header #E8CCFF.
	Variant 4	Banner #CCFFE7; header #CCFFCC.

Table 8: Template ingredients for scorecard image generation.

Model		MMCricBench-E-1.5K								MMCricBench-H-1.5K							
		Single				Multi				Single				Multi			
	Cat.	i	ii	iii	iv	i	ii	iii	iv	i	ii	iii	iv	i	ii	iii	iv
GPT4o	C1	84.1	30.6	**50.0**	45.0	**79.6**	28.33	16.6	50.0	81.3	15.9	**50.0**	**30.4**	71.1	15.0	**41.6**	27.7
	C2	67.9	**94.4**	NA	**50.2**	60.0	**75.00**	NA	**56.90**	55.5	**75.0**	NA	27.1	33.3	**80.56**	NA	**48.2**
	C3	57.6	65.8	44.7	35.7	69.23	17.14	NA	31.6	35.7	33.3	36.8	16.3	67.3	25.7	NA	22.78
Qwen2.5VL [7B]	C1	**87.3**	68.1	0.0	25.6	72.8	21.6	**33.3**	27.7	**85.8**	69.9	0.0	18.2	**76.2**	20.0	16.6	11.1
	C2	55.5	69.5	NA	42.5	46.6	69.4	NA	33.3	32.7	72.8	0.0	19.1	46.6	63.8	NA	22.4
	C3	58.9	10.7	5.2	30.1	59.6	2.8	Na	16.4	51.5	10.7	13.1	18.1	67.3	2.8	NA	7.5

Table 9: Answer-type-wise accuracy (%) of GPT-4o and Qwen2.5VL [7B] on MMCricBench-E-1.5K and MMCricBench-H-1.5K across single-image and multi-image settings. The table highlights the models' performance breakdown by question category (C1-C3) and answer type: (i) Binary, (ii) Numerical, (iii) Categorical, and (iv) Open-ended.

Country	India	Australia	Pakistan
League Teams	Mumbai Indians	Sydney Thunder	Islamabad United
	Kolkata Knight Riders	Adelaide Strikers	Lahore Qalandars
	Kings XI Punjab	Melbourne Renegades	Karachi Kings
	Royal Challengers Bangalore	Sydney Sixers	Peshawar Zalmi
	Gujarat Lions	Perth Scorchers	Multan Sultans
	Delhi Daredevils	Hobart Hurricanes	Quetta Gladiators
	Sunrisers Hyderabad	Brisbane Heat	
	Rising Pune Supergiants	Melbourne Stars	
	Chennai Super Kings		
	Rajasthan Royals		
	Delhi Capitals		

Table 10: List of franchise Cricket teams by Country and League included in the dataset.

Cat.	Category Name	Example Question	#sQ's	#mQ's	#Total	%
C1	Direct Retrieval & Simple Inference	Which bowler has bowled the most no-balls in the match?	9	5	14	0.93
		Who got out for a duck in the second innings?	16	12	28	1.87
		Did any batsman score a century in the match?	25	11	36	2.4
		Which bowler has bowled the maximum maidens?	4	15	19	1.27
		Did any bowler take a 3-fer in the match?	24	9	33	2.2
		Did any bowler take a 5-fer in the match?	18	7	25	1.67
		Did any bowler take a 6-fer in the match?	23	8	31	2.07
		How many wides were bowled by Team 1?	27	13	40	2.67
		How many no balls were bowled by Team 1?	24	9	33	2.2
		How many leg byes did Team 1 concede?	23	6	29	1.93
		How many byes did Team 1 concede?	49	24	73	4.87
		How many extras are bowled in match?	35	11	46	3.07
		Which bowler has bowled the most wides in the match?	22	8	30	2
		Who got out for a duck in the first innings?	6	5	11	0.73
		Did any bowler take a 4-fer in the match?	15	7	22	1.47
		Has [Batsman X] taken more wickets than [Batsman Y]?	30	19	49	3.27
		Which bowler has conceded the most extras?	57	16	73	4.87
		Who has hit the maximum sixes?	24	8	32	2.13
		Does [Batsmax X] hit more sixes than [Batsman Y]?	16	15	31	2.07
		How many extras were bowled in the first innings?	46	18	64	4.27
C2	Basic Arithmetic Reasoning & Conditional Logic	How many batsmen have scored a century?	19	6	25	1.67
		How many batsmen have been dismissed for a duck?	20	16	36	2.4
		Which bowler took a 3-fer in the match?	27	21	48	3.2
		Which bowler took a 5-fer in the match?	-	9	9	0.6
		Which bowler took a 6-fer in the match?	-	2	2	0.13
		What is [Batsman X] strike rate?	46	18	64	4.27
		Did [Batsman X] score better in the first innings or the second innings?	-	7	7	0.47
		Which batsman scored a century in the match?	11	15	26	1.73
		Which bowler took a 4-fer in the match?	6	11	17	1.13
		Has [Batsman X] hit more boundaries than [Batsman X]?	13	2	15	1
		Which batsman was dismissed for a golden duck in the match?	24	15	39	2.6
C3	Multi-step Reasoning & Quantitative Analysis	How many batsmen had a strike rate greater than 70 in the first innings?	21	11	32	2.13
		Which innings had the maximum maidens?	4	12	16	1.07
		Has any batsman been dismissed for a golden duck in the match?	54	15	69	4.6
		Which batsman had the highest strike rate (minimum 10 balls faced)?	37	17	54	3.6
		Which batsman had the highest boundary percentage?	35	18	53	3.53
		Which bowler had the better economy rate in the first innings?	38	18	56	3.73
		Which innings had the higher run rate?	38	15	53	3.53
		Which batsman had a strike rate greater than 70 in the first innings?	49	13	62	4.13
		Has the same fielder caught any batsman twice?	37	14	51	3.4
		Has any batsman been dismissed twice by the same bowler?	28	19	47	3.13
Total			1000	500	1500	100

Table 11: Statistics of single-image and multi-image questions.

Question	SQL Query
Which bowler has bowled the most wides in the match?	SELECT Bowler_Name, SUM(WD) AS Total_Wides FROM bowling GROUP BY Bowler_Name ORDER BY Total_Wides DESC LIMIT 1;
Who got out for a duck in the first innings?	SELECT Batsman_Name FROM batting WHERE Runs = 0 AND Innings = 1 AND ‘Bowler/Catcher‘ NOT LIKE ’not out%’;
Did any bowler take a 4-fer in the match?	SELECT Bowler_Name, Wicket, Innings FROM bowling WHERE Wicket >= 4;
Has Batsman X taken more wickets than Batsman Y?	SELECT CASE WHEN SUM(CASE WHEN Bowler_Name = ’Bowler X’ THEN Wicket ELSE 0 END) >SUM(CASE WHEN Bowler_Name = ’Bowler Y’ THEN Wicket ELSE 0 END) THEN ’Yes’ ELSE ’No’ END AS Result FROM bowling WHERE Bowler_Name IN (’Bowler X’, ’Bowler Y’);
Which bowler has conceded the most extras?	SELECT Bowler_Name, SUM(WD + NB) AS total_extras FROM bowling_data GROUP BY Bowler_Name ORDER BY total_extras DESC LIMIT 1;
Who has hit the maximum sixes?	SELECT Batsman_Name, MAX("6s") AS max_sixes FROM batting_data;
Does Batsmax X hit more sixes than Batsman Y?	SELECT Batsman_Name, SUM(‘6s‘) AS Total_Sixes FROM batting WHERE Batsman_Name IN (’Batsman X’, ’Batsman Y’) GROUP BY Batsman_Name;
How many extras were bowled in the first innings?	SELECT SUM(WD + NB) AS Total_Extras FROM bowling WHERE Innings = 1; for leg bye and bye we calculated manually
Which bowler has bowled the most no-balls in the match?	SELECT Bowler_Name, SUM(NB) AS Total_Wides FROM bowling GROUP BY Bowler_Name ORDER BY Total_NB DESC LIMIT 1;
Who got out for a duck in the second innings?	SELECT Batsman_Name FROM batting WHERE Runs = 0 AND Innings = 2 AND ‘Bowler/Catcher‘ NOT LIKE ’not out%’;
Did any batsman score a century in the match?	SELECT Batsman_Name, Runs, Innings FROM batting WHERE Runs >= 100;
Which bowler has bowled the maximum maidens?	SELECT Bowler_Name, SUM(Maiden) AS Total_Maidens FROM bowling GROUP BY Bowler_Name HAVING Total_Maidens >1 ORDER BY Total_Maidens DESC;
Did any bowler take a 3-fer in the match?	SELECT Bowler_Name, Wicket, Innings FROM bowling WHERE Wicket >= 3;
Did any bowler take a 5-fer in the match?	SELECT Bowler_Name, Wicket, Innings FROM bowling WHERE Wicket >= 5;
Did any bowler take a 6-fer in the match?	SELECT Bowler_Name, Wicket, Innings FROM bowling WHERE Wicket >= 6;
How many wides were bowled by Team 1?	SELECT SUM(WD) AS Total_Wides_By_Team1 FROM bowling WHERE Innings IN (1, 3);
How many no balls were bowled by Team 1?	SELECT SUM(NB) AS Total_Wides_By_Team1 FROM bowling WHERE Innings IN (1, 3);
How many leg byes did Team 1 concede?	SELECT SUM(byes) AS Total_Wides_By_Team1 FROM bowling WHERE Innings IN (1, 3);
How many byes did Team 1 concede?	SELECT SUM(legbyes) AS Total_Wides_By_Team1 FROM bowling WHERE Innings IN (1, 3);
How many extras are bowled in match?	SELECT SUM(WD + NB) AS Total_Extras_In_Match FROM bowling;
How many batsmen have scored a century?	SELECT COUNT(*) AS Century_Count FROM batting WHERE Runs >= 100 AND "Bowler/Catcher" NOT LIKE ’%not out%’;
How many batsmen have been dismissed for a duck?	SELECT CASE WHEN COUNT(*) = 0 THEN ’None’ ELSE CAST(COUNT(*) AS TEXT) END AS Duck_Result FROM batting WHERE Runs = 0 AND "Bowler/Catcher" NOT LIKE ’%not out%’;
Which bowler took a 3-fer in the match?	SELECT Bowler_Name, Wicket, Innings FROM bowling WHERE Wicket >= 3;
Which bowler took a 5-fer in the match?	SELECT Bowler_Name, Wicket, Innings FROM bowling WHERE Wicket >= 5;
Which bowler took a 6-fer in the match?	SELECT Bowler_Name, Wicket, Innings FROM bowling WHERE Wicket >= 6;
What is Batsman X strike rate?	SELECT ROUND((SUM(Runs) * 100.0 / SUM(Balls)), 2) AS Strike_Rate FROM batting WHERE Batsman_Name = ’batsman X’ AND Innings = 1;
Did Batsman X score better in the first innings or the second innings?	SELECT CASE WHEN SUM(CASE WHEN Innings = 0 THEN Runs ELSE 0 END) >SUM(CASE WHEN Innings = 2 THEN Runs ELSE 0 END) THEN ’1st Innings’ WHEN SUM(CASE WHEN Innings = 2 THEN Runs ELSE 0 END) >SUM(CASE WHEN Innings = 0 THEN Runs ELSE 0 END) THEN ’2nd Innings’ ELSE ’None’ END AS Better_Innings FROM batting WHERE Batsman_Name = ’batsman X’;
Which batsman scored a century in the match?	SELECT Batsman_Name, Runs, Innings FROM batting WHERE Runs >= 100;
Which bowler took a 4-fer in the match?	SELECT Bowler_Name, Wicket, Innings FROM bowling WHERE Wicket = 4;

Table 12: Question and its SQL query to extract answer from CSV.

Question	SQL Query
Has Batsman X hit more boundaries than Batsman X?	SELECT CASE WHEN SUM(CASE WHEN Batsman_Name = 'batsman X' THEN '4s' + '6s' ELSE 0 END) >SUM(CASE WHEN Batsman_Name = 'batsman Y' THEN '4s' + '6s' ELSE 0 END) THEN 'Yes' ELSE 'No' END AS Result FROM batting WHERE Batsman_Name IN ('batsman X', 'batsman Y');
Which batsman was dismissed for a golden duck in the match?	SELECT Batsman_Name FROM batting WHERE Runs = 0 AND 'Bowler/Catcher' NOT LIKE 'not out%';
How many batsmen had a strike rate greater than 70 in the first innings?	SELECT COUNT(DISTINCT Batsman_Name) AS Count FROM batting WHERE Innings = 1 AND Strike_Rate >70;
Which innings had the maximum maidens?	SELECT Innings, SUM(Maiden) AS Total_Maidens FROM bowling GROUP BY Innings HAVING Total_Maidens >1 ORDER BY Total_Maidens DESC LIMIT 1;
Has any batsman been dismissed for a golden duck in the match?	SELECT Batsman_Name, Innings FROM batting WHERE Runs = 0 AND Balls = 1;
Which batsman had the highest strike rate (minimum 10 balls faced)?	SELECT Batsman_Name FROM batting WHERE Innings = 1 AND Balls >= 10 ORDER BY Strike_Rate DESC LIMIT 1;
Which batsman had the highest boundary percentage?	SELECT Batsman_Name FROM batting WHERE Innings = 1 AND Balls >0 ORDER BY ((([4s]*4 + [6s]*6) * 100.0 / Runs)) DESC LIMIT 1;
Which bowler had the better economy rate in the first innings?	SELECT Bowler_Name, ROUND((SUM(Runs) * 1.0 / SUM(Over)), 2) AS Economy_Rate FROM bowling WHERE Innings = 1 GROUP BY Bowler_Name ORDER BY Economy_Rate ASC LIMIT 1;
Which innings had the higher run rate?	SELECT Innings FROM batting GROUP BY Innings ORDER BY SUM(Runs)*1.0/COUNT(DISTINCT Batsman_Name) DESC LIMIT 1;
Which batsman had a strike rate greater than 70 in the first innings?	SELECT GROUP_CONCAT(Batsman_Name) AS Aggressive_Batsmen FROM batting WHERE Innings = 1 AND Strike_Rate >70 AND Balls >= 10 GROUP BY Batsman_Name HAVING Strike_Rate >70;
Has the same fielder caught any batsman twice?	SELECT TRIM(SUBSTR('Bowler/Catcher', 3, INSTR('Bowler/Catcher', 'b') - 3)) AS Fielder, COUNT(*) AS Catches FROM batting WHERE 'Bowler/Catcher' LIKE 'c %b %' GROUP BY Fielder HAVING Catches >1;
Has any batsman been dismissed twice by the same bowler?	SELECT Batsman_Name, SUBSTR('Bowler/Catcher', INSTR('Bowler/Catcher', 'b ') + 2) AS Bowler, COUNT(*) AS Dismissals FROM batting WHERE 'Bowler/Catcher' LIKE '%b %' GROUP BY Batsman_Name, Bowler HAVING Dismissals >1;

Table 13: Question and its SQL query to extract answer from CSV continued.