When Reasoning Meets Information Aggregation: A Case Study with Sports Narratives

Yebowen Hu,[1*] Kaiqiang Song,[2*] Sangwoo Cho,[2*] Xiaoyang Wang,[2] Wenlin Yao,[2] Hassan Foroosh,[1] Dong Yu,[2] Fei Liu[3]

[1]University of Central Florida [2]Tencent AI Lab, Seattle [3]Emory University

[1]{yebowen.hu, hassan.foroosh}@ucf.edu [3]fei.liu@emory.edu

[2]{riversong, swcho, shawnxywang, wenlinyao, dyu}@global.tencent.com

Abstract

Reasoning is most powerful when an LLM accurately aggregates relevant information. We examine the critical role of information aggregation in reasoning by requiring the LLM to analyze sports narratives. To succeed at this task, an LLM must infer points from actions, identify related entities, attribute points accurately to players and teams, and compile key statistics to draw conclusions. We conduct comprehensive experiments with real NBA basketball data and present SPORTSGEN, a new method to synthesize game narratives. By synthesizing data, we can rigorously evaluate LLMs' reasoning capabilities under complex scenarios with varying narrative lengths and density of information. Our findings show that most models, including GPT-4o, often fail to accurately aggregate basketball scores due to frequent scoring patterns. Open-source models like Llama-3 further suffer from significant score hallucinations. Finally, the effectiveness of reasoning is influenced by narrative complexity, information density, and domain-specific terms, highlighting the challenges in analytical reasoning tasks.[1]

1 Introduction

Accurately aggregating information is essential for reasoning (Sprague et al., 2024; Wang et al., 2023a; Yasunaga et al., 2024; Hu et al., 2024c). This is especially true when reasoning over longitudinal data (Łukasz Kidziński and Hastie, 2021). For example, when assessing patient outcomes over years to determine the effectiveness of new medications, models must accurately track changes to provide insights into trends, patterns, and potential causal relationships. While LLMs hold significant potential in reasoning, understanding how they aggregate information to support reasoning is key to ensuring their robustness in decision-making.

Sports data have recently emerged as a popular testing ground for LLMs (Srivastava et al., 2023; Wang et al., 2023b; Li et al., 2024b; Yang et al., 2024c). Unlike health data, sports narratives are readily accessible from diverse sources, e.g., ESPN, CBS Sports, BBC Sport, and Sports Illustrated.[2] These narratives provide detailed event descriptions over extended game periods, making them an excellent case study to explore how LLMs reason with longitudinal data. Figure 1 presents a snippet from an NBA basketball game. The play-by-plays capture key actions of the game, such as passes, shots, and fouls, along with timestamps and team-player affiliations. *Our research involves using LLMs to track scoring actions within these detailed game descriptions and to link entities for comprehensive game analysis.* These games include a mix of expected and unexpected events, presenting unique challenges for reasoning tasks.

LLMs's reasoning abilities, including multi-hop, deductive, inductive, and abductive reasoning, have garnered increasing attention (Hu and Shu, 2023; Shi et al., 2023; Zhao et al., 2023a; Yang et al., 2024b,a; Li et al., 2024a; Duan et al., 2024; Chu-Carroll et al., 2024). For example, Sprague et al. (2023) introduce a dataset designed to evaluate language models on multi-step reasoning with 1,000-word murder mysteries and object placement. Yang et al. (2024b) explore latent multi-hop reasoning and found evidence of its use in fact composition. Berglund et al. (2024) study the reversal curse that LLMs trained on 'A is B' fail to learn that 'B is A.' Distinguishing from previous studies, our approach introduces a quantitative aspect to reasoning with game narratives, where the LLM must summarize key statistics to derive conclusions.

We focus on *analytical reasoning* in this work, which involves identifying relationships between elements, drawing inferences and synthesizing infor-

[*]Work done during Yebowen Hu's internship; Kaiqiang Song and Sangwoo Cho were full-time researchers at Tencent AI Lab, Seattle, USA at the time of this work.

[1]`https://github.com/YebowenHu/SportsGen`

[2]`espn.com cbssports.com bbc.com/sport si.com`

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 4293–4308
November 12-16, 2024 ©2024 Association for Computational Linguistics

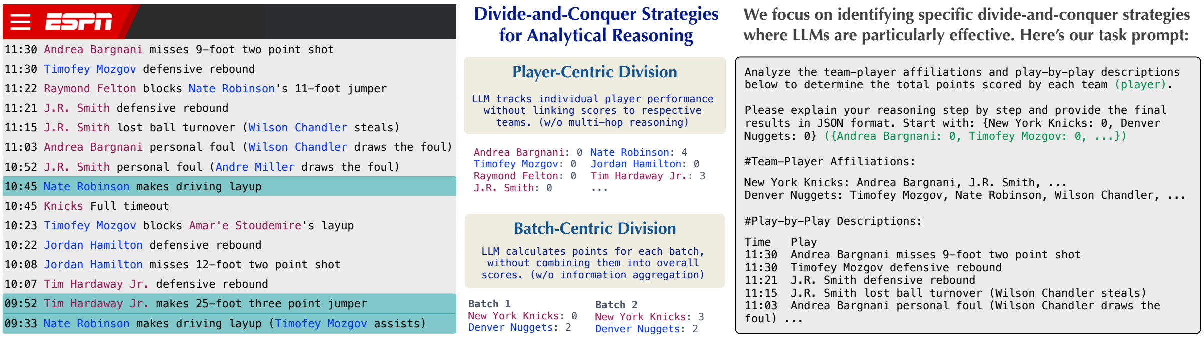

Figure 1: ESPN's NBA game play-by-play descriptions. We are particularly interested in exploring whether LLMs can perform analytical reasoning in a more focused and manageable context using divide-and-conquer strategies.

mation (Savage et al., 2023). Notably, we evaluate if LLMs perform better in a more focused and manageable context when using a *divide-and-conquer* strategy. Sports narratives, for instance, can be segmented along multiple dimensions. We explore division methods that *simplify multi-hop reasoning* or *facilitate information aggregation*, thereby easing the load on LLMs. We further introduce a new Discounted Cumulative Accuracy (DCA) metric. DCA differs from traditional accuracy by allowing a small margin of error. This helps us identify scenarios where LLMs significantly hallucinate, rather than grounding predictions on the narratives. Our experiments reveal multiple limitations of state-of-the-art LLMs in using divide-and-conquer strategies for analytical reasoning.

With data synthesis, we further stress test LLMs' reasoning abilities in complex and novel scenarios. LLMs are required to handle a range of narratives that differ in length, complexity, and information density, and may also involve a shift from natural language to symbolic reasoning. Our approach, SPORTSGEN, enhances the control over narrative complexity when compared to human-written game narratives and few-shot prompting. It contributes to ongoing research in data synthesis (Veselovsky et al., 2023; Arora et al., 2023; Kang et al., 2024; Divekar and Durrett, 2024) by creating synthetic sports narratives that help evaluate the analytical capabilities of LLMs. Our research contributions are summarized as follows.

- We investigate LLMs' reasoning abilities by analyzing play-by-play sports narratives. This involves tracking scoring actions, inferring points from actions, recognizing players and teams, and summarizing key statistics to draw conclusions. By exploring how LLMs aggregate information, we gain insights into their potential in longitudinal studies, such as patient health management, where the LLM must identify and interpret recurring patterns to make critical decisions.
- We observe that LLMs struggle more with accurately aggregating points than assigning points to players and teams. Basketball's frequent scoring poses a challenge for tracking actions. Moreover, when input narratives are short and instructions are lengthy, LLMs may overlook the narrative and hallucinate game points. Our SPORTSGEN approach has significant implications for simulating complex reasoning scenarios that involves synthesizing both text and numerical data.

2 Related Work

Success in reasoning hinges on effective content selection. Recent studies have placed a growing emphasis on LLM's reasoning capabilities, exploring various types such as deductive, inductive, abductive, and multi-hop reasoning (Bostrom et al., 2021; Huang and Chang, 2022; Hu and Shu, 2023; Zhao et al., 2023a; Holliday and Mandelkern, 2024). Efforts to enhance LLMs' reasoning abilities include prompting, supervised fine-tuning, and adjustments to decoding (Wei et al., 2023; Chen et al., 2023; Zhao et al., 2023a; Burnell et al., 2023; Zheng et al., 2023; Cheng et al., 2024; Ahn et al., 2024; Wang and Zhou, 2024). In our paper, we propose a quantitative approach to reasoning by requiring the LLM to identify and consolidate relevant facts. Similar to how frequency indicates salient content in multi-document summarization (Fabbri et al., 2019; Mao et al., 2020; Gholipour Ghalandari et al., 2020; Lebanoff et al., 2021; Thomson et al., 2023), we conjecture that how LLMs aggregate information is critical in addressing complex reasoning challenges.

Numerical reasoning is applied in financial question answering and solving mathematical word problems, and numerous datasets have been developed for these purposes (Amini et al., 2019; Cobbe et al., 2021; Patel et al., 2021; Zhu et al., 2021; Chen et al., 2022; Liu et al., 2023; Zhao et al., 2023b; Lu et al., 2023). Our study explores a new angle—investigating LLMs' ability to analytically solve problems using divide-and-conquer strategies (Lee and Kim, 2023; You et al., 2023). Sports data are complex and multifaceted. Our focus is on pinpointing specific divide-and-conquer strategies where LLMs are particularly effective.

Sports data plays a pivotal role in various language tasks, such as real-time summarization of games, data-to-text generation, and analysis of commentator bias (Wiseman et al., 2017; Edouard et al., 2017; van der Lee et al., 2017; Puduppully et al., 2019; Merullo et al., 2019; Huang et al., 2020; Hu et al., 2024a,b). It also influences the analysis of domains like game reviews and gameplay logs, enhancing our understanding of gameplay commentary dynamics (Lukin, 2020; Kicikoglu et al., 2020; Gu et al., 2022; Furman et al., 2022). Building on research by Hu et al. (2024b), our study explores how well LLMs can track scoring actions across lengthy narratives and accurately link entities for in-depth game analysis. With SportsGen for data synthesis (Veselovsky et al., 2023; Arora et al., 2023; Divekar and Durrett, 2024), we take a step further by creating diverse narratives that vary in style, complexity, and level of detail to better assess LLM effectiveness.

3 Analytical Reasoning with Divide-and-Conquer Strategies

Our research uses analytical reasoning to accurately calculate team points from sports narratives. We specifically focus on ESPN's NBA game play-by-play descriptions (Figure 1). On average, an NBA game consists of 466 plays and 6,229 tokens, with the longest narrative containing up to 7,322 tokens. To succeed at this task, an LLM must understand the basic rules of the game, distinguishing between scoring plays, such as three-pointers and field goals, and non-scoring actions like passing. Further, the model must correctly attribute each scoring action to the appropriate player and team, which requires nuanced reasoning. Finally, the LLM must aggregate all the scored points to determine the final scores for the teams. This process tests the LLM in three critical areas: knowledge acquisition, point referencing, and effective score aggregation. Errors in any of these areas can lead to incorrect results.

Player-Centric Division. We are particularly interested in exploring whether LLMs can perform analytical reasoning in a more focused and manageable context using a divide-and-conquer strategy. For instance, sports narratives can be segmented along multiple dimensions. In our ***player-centric division***, the LLM assesses player performance based on play-by-play descriptions. Here, the LLM receives a list of players, without their team affiliations, and is tasked with tracking and summarizing each player's performance throughout the game. The final results are presented in JSON format. Following this, we link the individual players' scores with their respective teams to calculate the overall team scores. This player-centric approach employs LLMs to track individual player performances, allowing for a direct calculation of team scores from player data. It lessens the burden on LLMs to use multi-hop reasoning for inferring team scores from player scores.

Batch-Centric Division. Instead of processing the play-by-play narrative at once, we break it down into smaller, non-overlapping batches, each containing K plays. In our ***batch-centric division*** approach, each batch is analyzed independently by the LLM to determine the team points for that segment. This approach allows the LLM to narrow its focus, helping it easily identify where errors might be occurring. It also eases the burden on LLMs, as they are not required to aggregate points to determine total scores from batch scores; instead, this is handled through direct calculation. We compare this approach to ***monolithic processing***, where an LLM processes the entire game narrative in one attempt. Interestingly, one would expect that when LLMs are provided with data in smaller batches, they would perform better in this more granular and manageable setting. In §5, we report notable findings that highlight the limitations of state-of-the-art LLMs when using these divide-and-conquer strategies.

We focus on identifying specific data divisions where LLMs are particularly effective. Real-world data can be complex and multifaceted. By focusing on sports narratives, we aim to understand the limitations of LLMs in reasoning and information aggregation, since complex reasoning often require LLMs to aggregate relevant information effectively. Our research does not introduce new methods of reasoning. Instead, it uses chain-of-thought prompting, which enables Transformers to tackle inher-

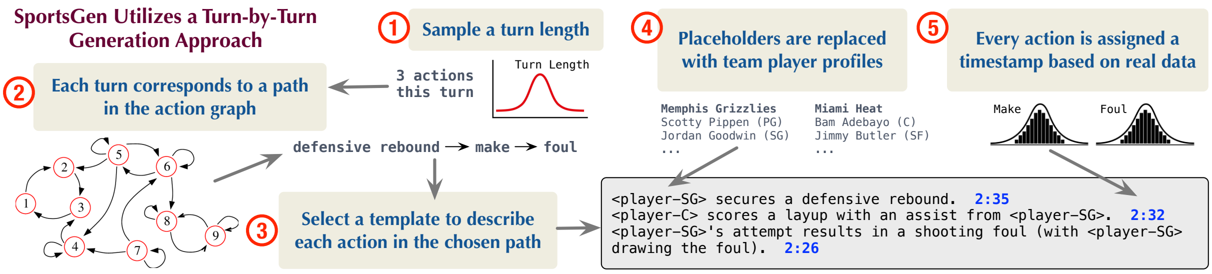

Figure 2: SPORTSGEN, a new method that synthesizes sports narratives by modeling game dynamics.

ently serial problems (Li et al., 2024c). In the following section, we propose synthesizing sports narratives, which will allow us to challenge LLMs in varying and unseen scenarios.

4 Creating Data with SPORTSGEN

We introduce SPORTSGEN, a new method that synthesizes sports narratives by modeling game dynamics. SPORTSGEN offers a platform to assess LLMs' reasoning capabilities in novel scenarios that may not exist in real data. Since synthesized narratives have not been encountered by LLMs during pretraining, they also serve as a valuable benchmark for future LLM assessments. Previously, data synthesis has been utilized in fine-tuning, distilling, and evaluating LLMs for tasks such as hallucination detection, topic classification, sarcasm and humor detection (Veselovsky et al., 2023; Tang et al., 2023; Arora et al., 2023; Kang et al., 2024; Zhang et al., 2024; Divekar and Durrett, 2024). Our work expands on these efforts by focusing on the specific challenges in generating game narratives.[3]

Setting Up Team Profiles. We create profiles for two basketball teams, Team $\mathcal{T}_1$ and Team $\mathcal{T}_2$, for a specific game. Each team has five players, filling the roles of point guard (PG), shooting guard (SG), small forward (SF), power forward (PF), and center (C). Each team has an efficiency score $\mathcal{E}$ from 0 to 100, which reflects the points they score per 100 possessions.[4] We start with real NBA teams for these profiles, and later users can select either real or synthetic players to create fantasy teams.

Turn-by-Turn Game Generation. Our SPORTSGEN renders realistic sports narratives using a turn-by-turn approach, which mirrors the back-and-forth dynamics seen in live sports games. In each turn, a team controls the ball and performs a series of plays[5] such as passes, shots, and fouls, which are key actions during the game. After each turn, possession generally shifts to the opposing team. Each quarter consists of alternating turns between the two teams, with each turn being given an allotted time frame. For example, a turn might unfold as follows: "Lakers take possession, make a quick pass, and take a shot that misses," all within about 15 seconds. Each basketball game lasts 48 minutes. Our approach continues to add turns until nearing the end of a quarter's time limit. If the ending turn exceeds this limit, it will be truncated.

Building an Action Graph. When a team has the ball, they are on offense and use actions such as passing, dribbling, shooting to score points. The defense tries to stop them by blocking, stealing, and rebounding. We refer to each sequence of these actions as a **turn**, and characterize it using a Markov graph. For example, a turn might start with a *defensive rebound*, followed by a *missed shot*, and then conclude. In the graph, each significant action is represented as a node, and transitions between nodes show how team members cooperate and execute their tactics. The graph begins and ends with special nodes that mark the start and end of a turn. Our model includes 16 key actions; an example is provided in Appendix 5. This action graph helps to visualize the game's flow within each turn.

Action Templates. Each action is linked to specific templates that describe it. E.g., '*<player-PF> misses a 20-foot jumper.*' and '*<player-PF> fails to convert a 14-foot pull-up jump shot.*' correspond to the action 'miss'. To gather these templates, we first search NBA data using keywords selected by experts to identify plays that describe each action. We then provide these initial templates to GPT-4o to increase their lexical diversity and enhance the

[3]Spatial information, such as player positions on the field, is not included in our synthesized narratives, as it falls outside our current scope.

[4]Efficiency scores for basketball teams are obtained from ESPN. We aim to model team effectiveness in goal attempts; future research may consider individual player metrics.

[5]In our paper, 'plays' and 'actions' are used interchangeably to describe significant game actions.

Action - "make"
• <player-SG> successfully makes a 2-foot two point shot with the assistance of <player-SF>. • <player-SF> executes a delicate finger roll layup. • <player-PG> hit a 26-foot three pointer. • <player-PG> executes a 21-foot step back jumpshot. • <player-SG> sinks a 26-foot three-point shot with an assist from <player-C>. • <player-SF> executes a tip shot...
Action - "miss"
• <player-PG> fails to make a 25-foot standard jump shot. • <player-C> fails to convert a 20-foot jumper. • <player-SG> unsuccessfully attempts a three point jumper. • <player-PF> misses a 20-foot jumper. • <player-PG> misses a 4-foot driving layup...
Action - "block"
• <player-PG> individual defensive play resulted in a personal block (<player-SG> draws the foul). • <player-PF> executes a personal block as <player-PG> draws the foul. • <player-PG> successfully executes a personal block against the opponent, drawing the foul in the process. • <player-PG> commits a shooting block foul and <player-G> is fouled during the play...

Table 1: Each action is linked to specific templates that describe it. The templates are enhanced by GPT-4o to increase their lexical diversity and narrative quality.

narrative quality. An example of actions and their templates is shown in Table 1. During generation, each template linked to a sampled action has an equal probability of being used. The transitions between actions are determined by analyzing real game narratives to calculate maximum likelihood estimates.

Controlling Narrative Complexity. We adjust the complexity of the game by changing the number of actions per turn. Longer turns can extend the overall narrative, whereas shorter turns create more frequent scoring opportunities, as each turn often results in an attempt to score. In NBA narratives, a turn might have anywhere from 1 to 10 actions, with an average of 1.65 actions. This results in a scoring to non-scoring action ratio $\mathcal{R}$ = 1:3. To model turn length, we employ a Gaussian distribution $\mathcal{G}$ parameterized by the mean (μ) and standard deviation (σ), which we calibrate using NBA data. During synthesis, we vary μ to generate narratives with different turn lengths while keeping σ stable. This approach allows us to produce narratives with realistic scoring ratios such as $\mathcal{R}$ = 1:2, 1:3, 1:4, or 1:5, effectively capturing the dynamics of actual basketball games.

Rendering the Narrative. We begin by selecting turn lengths from the Gaussian function $\mathcal{G}$ (Figure 2). Each turn is then rendered by tracing a path through the action graph that matches the designated length. We evaluate these paths for quality and reject any that are unlikely in a basketball game according to a set of expert rules. The probability of a turn leading to a score is determined using the Efficiency Score ($\mathcal{E}$) for the involved team. From this, we sample a '1' to indicate scoring and '0' otherwise. A scoring turn must include a 'make' action to indicate a successful score, and we record the points earned by the team in the box score.

We repeatedly sample from the action graph to find a path that meets our length and scoring criteria. We then select a template to describe each action in the chosen path. Next, placeholders are replaced with relevant team player profiles, e.g., 'player-PG' is replaced with the actual point guard's name. Each action is tagged with a timestamp indicating the time elapsed since the previous action. We calculate these time intervals using real data and create a Gaussian $\mathcal{G}_a$ for each action type. We alternate between the profiles of the two teams involved in the game. This process is repeated until we have created enough turns to complete one game quarter, after which we proceed to generate the remaining quarters to complete the full game narrative.

5 Experiments

Sports data are accessible from a variety of sources such as ESPN, BBC Sport, CBS Sports, and Sports Illustrated. This work utilizes the basketball play-by-plays provided by Hu et al. (2024b) consisting of 28,492 NBA games from 2002 to 2023, gathered from ESPN. Due to resource constraints, we subsampled the original dataset to $D_{\mathcal{H}}$, corresponding to the number of human-written narrative quarters. Further, our proposed method, SportsGen, generates synthesized game narratives with varied scoring to non-scoring (S:NS) ratios including 1:2, 1:3, 1:4, and 1:5. $D_{\mathcal{S}}$ represents the number of game quarters synthesized by SportsGen.

Accuracy. Our goal is to enable LLMs' to track and predict the total points scored by each team at the end of a game quarter. We choose quarter-level evaluation because current LLMs do not perform well at tracking points for full games. The ***accuracy*** metric strictly evaluates whether the model's predictions matches with the actual scores for each quarter and team. Here, each team is considered as a separate data point. For human-written narratives, our dataset includes detailed box scores sourced from ESPN. For our SportsGen generated narratives, ground-truth scores are obtained during the generation process.

Discounted Cumulative Accuracy (DCA). "Exact match" accuracy is not an ideal metric for track-

	Model	Human-Written Sports Narratives						Synthetic Narratives		
		DnC-1	DnC-3	DnC-10	DnC-30	DnC-P	Mono	DnC-10	DnC-P	Mono
Accuracy	Llama3-8B-Inst	7.92	33.79	33.42	38.61	4.70	4.43	11.67	5.86	3.78
	Gemini-Pro-1.5	48.27	67.20	64.36	32.43	8.79	4.58	15.00	11.67	3.75
	GPT-3.5-Turbo	14.60	56.68	44.43	21.41	6.68	2.00	5.00	8.33	1.25
	Llama3-70B-Inst	22.40	63.12	80.45	84.53	21.29	17.45	**35.00**	24.17	9.17
	GPT-4o	**72.77**	**84.28**	**88.61**	**84.90**	**68.19**	45.54	28.33	**52.08**	37.92
	Claude-3-Opus	60.64	75.99	84.16	81.31	61.39	**67.20**	31.67	42.50	**54.58**
DCA	Llama3-8B-Inst	39.18	87.96	86.18	86.79	36.11	41.5	71.21	55.94	38.08
	Gemini-Pro-1.5	91.58	95.83	94.94	81.75	46.21	18.56	81.52	69.09	24.58
	GPT-3.5-Turbo	70.66	93.27	89.60	74.31	55.64	29.00	62.27	60.80	25.19
	Llama3-70B-Inst	55.54	92.20	**96.12**	97.52	63.81	74.18	87.73	77.01	68.90
	GPT-4o	93.06	**97.52**	**98.41**	**97.66**	**92.38**	86.70	82.42	78.79	82.69
	Claude-3-Opus	**94.76**	97.07	98.28	97.60	91.39	**93.56**	**90.00**	**85.00**	**89.32**

Table 2: Results of top LLMs in calculating team points from sports narratives, divided by batches of 1, 3, 10, or 30 plays (DnC-{1,3,10,30}), by individual players (DnC-P), and monolithic processing (Mono).

ing numerical values, as it is strict and can unfairly penalize LLMs. To address this, we introduce the ***discounted cumulative accuracy*** metric, which differs from traditional accuracy by allowing a small margin of error. DCA rewards predictions that are close to the true value, with diminishing gains as predictions deviate further. This approach draws inspiration from the DCG (Järvelin and Kekäläinen, 2002), a highly effective metric in information retrieval. It provides a balanced evaluation by cumulatively adjusting rewards, allowing for a fairer assessment of a system's performance.

Concretely, we denote the tolerance level as $\mathtt{T}$ (Eq.(1)). At zero tolerance ($\mathtt{T} = 0$), the DCA metric becomes standard accuracy. As the tolerance level increases, performance differences between models become less pronounced, making this a more forgiving evaluation metric. During each time step t, we apply a discounting factor of $1 - \frac{t}{\mathtt{T}}$. With a tolerance of 10 points ($\mathtt{T} = 10$), this factor decreases gradually from 1 down to 0 in steps of 0.1.

We use p_t to denote the proportion of instances where the prediction error is exactly t points from the true value (Eq.(2)). Let N represent the total number of instances evaluated. For each instance, $|s_n - s_n^*|$ measures the absolute difference between the predicted and actual values. If this error surpasses $\mathtt{T}$, the instance does not contribute to the score. Higher accuracy and DCA scores indicate better-performing systems.

$$\mathtt{DCA} = \sum_{t=0}^{\mathtt{T}} p_t (1 - \frac{t}{\mathtt{T}}) \tag{1}$$

$$p_t = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_{\{|s_n - s_n^*| = t\}} \tag{2}$$

In the sections below, we evaluate leading LLMs on their analytical reasoning with game narratives using various divide-and-conquer strategies. We also compare human-written sports narratives against our SportsGen narratives to assess LLM effectiveness.

5.1 Divide-and-Conquer Results

In Table 2, we compare the analytical reasoning capabilities of leading LLMs.[6] Our evaluation shows that both GPT-4o and Claude-3-Opus are effective for this task, with Claude-3-Opus outperforming especially in monolithic processing 'Mono' scenarios, where the entire game narrative is processed as a whole. Here, Claude-3-Opus achieves 67.20% accuracy and a DCA score of 93.56%. This indicates that the model provides exactly accurate results in over 60% of cases and is within a minimal error margin ($\mathtt{T} = 0$) in >90% of the scenarios. This superior performance may stem from Claude-3-Opus's advanced logical reasoning and computational abilities. Of the three variants in the Claude-3 model family (Haiku, Sonnet, and Opus), Opus is the most sophisticated and costly. The expense for using this API is \$15 / million tokens for input and \$75 / million tokens for output.

On the other hand, GPT-4o excels when using the divide-and-conquer (DnC) strategy on human-written sports narratives. It achieves the highest accuracy at 88.61% and a DCA score of 98.41% with a batch size of 10 (DnC-10). This suggests that

[6] The lineup includes both proprietary and open-source models: Llama3-8B-Instruct, Llama3-70B-Instruct (MetaAI, 2024), OpenAI's GPT-4o and GPT-3.5-Turbo (OpenAI, 2024), Gemini-Pro-1.5 (GeminiTeam, 2023), and Anthropic's Claude-3-Opus (Anthropic, 2024). These LLMs are available to the research community from March to May 2024. The models can perform complex analyses, handle multi-step procedures, and tackle higher-order math and reasoning tasks.

Model	Accu. T=0	Disc. Cumu. Accu. T=1	T=3	T=5	T=10
Llama3-8B-Inst	4.43	8.53	17.22	25.48	41.50
Gemini-Pro-1.5	4.58	6.44	10.15	12.89	18.56
GPT-3.5-Turbo	2.00	3.55	8.17	13.69	29.00
Llama3-70B-Inst	17.45	29.58	47.65	59.59	74.18
GPT-4o	45.54	52.72	68.94	77.27	86.70
Claude-3-Opus	**67.20**	**72.22**	**83.08**	**88.41**	**93.56**

Table 3: We experiment with varying tolerance levels (T = 0, 1, 3, 5, 10) to assess the impact of this parameter. Adjusting the tolerance threshold generally does not alter the relative rankings of the models.

incorporating innovative strategies like DnC can potentially narrow the performance gaps among top-performing models. The open-source Llama3-70B-Inst model performs on par with GPT-4o and Claude-3-Opus. However, lesser models such as Llama3-8B-Inst, Gemini-Pro-1.5, and GPT-3.5-Turbo lag behind, with Llama3-8B-Inst delivering the lowest accuracy and DCA scores.

Batch-Centric vs. Player-Centric Division. We test the performance of LLMs using different batch sizes by dividing the game's play-by-plays into batches of 1, 3, 10, or 30 plays, denoted as DnC-{1,3,10,30} in Table 2. Our findings revealed two key insights. First, the optimal batch size varies between models. High-performing models such as Claude-3-Opus, GPT-4o, and Llama3-70B-Inst show peak performance with a batch size of 10. In contrast, less robust models perform best with smaller batches, such as 3 plays per batch. This suggests that finding the right balance between accuracy per batch and the total number of batches is crucial for achieving optimal results.

Notably, using the smallest batch size (1 play) did not yield the best performance. This was primarily because play-by-play data in small batches were often overshadowed by system messages and instructions, resulting in models, particularly the Llama3-8B-Inst, generating hallucinated scores. Increasing the batch size generally led to more scoring errors for all models. This emphasizes the need for a strategic approach to selecting batch sizes as well as enhancing models' instruction following abilities to improve their robustness. The DnC-P method slightly enhances LLM accuracy compared to processing the narrative as a whole ('Mono'), although it is not as effective as segmenting the narrative into smaller batches. Our findings suggest that while LLMs excel in reasoning, their performance in information aggregation is weaker.

	S:NS	#AActs	#SActs	#Turns	#Tokens
Few-Shot	1 : **0.4**	56	39	47	**856**
SportsGen (Ours)	1 : 2.3	89	27	48	1,609
	1 : **3.1**	97	26	45	**1,700**
	1 : 4.0	103	20	34	1,757
	1 : 5.2	102	16	27	1,722
Human Narr.	1 : **3.0**	114	29	65	**1,632**

Table 4: We compare human narratives with those of SportsGen and few-shot prompting. Prompting generates narratives deviating from typical basketball games.

Accuracy vs. DCA. In Table 3, we experiment with varying tolerance levels (T = 0, 1, 3, 5, 10) to assess the impact of this parameter. We note that adjusting the tolerance threshold generally does not alter the relative rankings of the models. However, increasing the tolerance tends to narrow the performance gap among different models. Importantly, accuracy scores alone may not be the most suitable metric for evaluation. Lower-performing models often result in accuracy in the low single digits. For example, the model Gemini-Pro-1.5 reports an accuracy of just 4.58%. Its performance remains the lowest even when tolerance is increased to 10 points (T=10), resulting in the lowest DCA score of 18.56%. Such findings highlight Gemini-Pro-1.5's considerably weaker analytical reasoning capabilities even under tolerance settings.

5.2 Human Versus Synthetic Narratives

In Table 4, we compare human-written sports narratives with those generated by SportsGen and a few-shot prompting approach. Due to resource limits, we set $D_{\mathcal{H}}$=400 for human narratives and $D_{\mathcal{S}}$=480 for each synthetic narrative set across all experiments. Few-shot prompting, detailed in Appendix A, uses existing play-by-plays to guide GPT-4o in creating synthesized game quarter. This method can result in repetitive scoring actions, with a high S:NS Ratio of 1:0.4, deviating from typical basketball games. In contrast, SportsGen offers enhanced controllability and conveniently generates box scores during generation. We adjust the turn length to produce games with varying information densities, featuring S:NS ratios of 1:2, 1:3, 1:4, and 1:5. With an S:NS ratio of 1:3, it mirrors the ratio found in actual human narratives and resulting in an average of 1,700 tokens per narrative. This makes SportsGen a more practical option for creating realistic sports narratives.

In Table 2, we report accuracy and DCA scores for SportsGen narratives with a 1:3 S:NS ratio, utilizing three strategies: DnC-10 (dividing the nar-

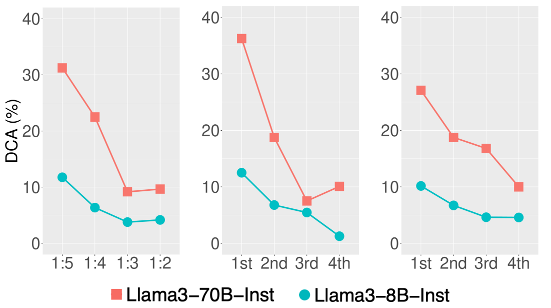

Figure 3: (Left) Synthesized narratives with varying S:NS ratios. (Middle) Narratives grouped by # of scoring actions. (Right) Narratives grouped by # of tokens.

rative into batches of 10 actions), DnC-P (division by individual players), and monolithic processing (Mono). We observe that accuracy for synthetic narratives tends to be lower than for NBA narratives. This may be due to the models' greater familiarity with NBA narratives, which may be included in the pretraining data, compared to our novel synthetic narratives. Further, our use of GPT-4o enhanced action templates introduces more lexical variety than the typical NBA narrative, creating a more challenging task for the models in accurately tracking team scores. Overall, Claude-3-Opus achieves the highest performance across all settings when measured by DCA scores, suggesting its effectiveness where some margin of error is permissible.

Qualitative Evaluation. We conduct a qualitative comparison between SportsGen and real NBA narratives. A human expert or GPT-4 review 100 narrative snippets from each source, with each snippet containing 10 plays. The evaluation focuses on (a) the naturalness of the language and (b) the logical coherence of the narratives. Evaluators choose their preferred narrative or opt for a tie, also providing justifications. We shuffle the order of pairs to reduce bias in this process, i.e., it is not guaranteed that Snippet A $\succ$ Snippet B when (A, B) is presented to the human or system. The results show that our human expert prefers SportsGen narratives 39% of the time, tied 9%, and favors NBA narratives 52% of the time. GPT-4's analysis was quite similar, with 39% preference for SportsGen, 7% ties, and 54% for NBA narratives. These insights suggest that while SportsGen is aligning with human narrative standards, some gaps remain.

5.3 Information Density on Reasoning Performance

Frequent scoring presents a challenge for LLMs in tracking and aggregating team points, especially during intense periods, e.g., the last quarter of a basketball game, where scoring spikes. In this study, we examine how models handle game quarters with varying levels of information density. We apply three settings: (a) four synthesized narrative sets with S:NS ratios of 1:5, 1:4, 1:3, and 1:2, where 1:2 represents a higher proportion of relevant information (see Figure 3, left); (b) we group synthesized narratives into four quartiles based on the total number of scoring actions (middle); and (c) for comparison, we also group narratives into four quartiles based on their length, measured in token count (see Figure 3, right).

In Figure 3, we report DCA scores for Llama3-8B-Inst and Llama3-70B-Inst to see how LLMs perform across various information densities. We note a decrease in performance as the models process inputs from low to high levels of information density, particularly in Llama3-70B-Inst, which exhibits the most noticeable decline at high S:NS ratios, such as 1:2 or 1:3. When analyzing narratives by length, the models exhibit a more gradual performance decrease. Our analysis indicates that *a model's analytical reasoning capabilities are linked to the quantity and density of relevant facts*, echoing Greg Kamradt's needle-in-a-haystack test, where the challenge is to retrieve a single fact from an LLM's extensive context window.[7]

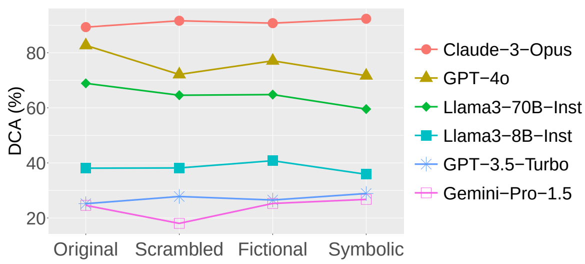

Figure 4: We examine four scenarios that increasingly alter the context to a more 'symbolic' format and evaluate the models' ability to calculate team points.

5.4 From Natural Language to Symbolic Reasoning

Symbolic reasoning uses symbols and logic to form traceable reasoning steps. Research has explored whether LLMs are capable of symbolic reasoning, or if they only infer patterns from data without explicit symbolic manipulation (Gaur and Saunshi, 2023; Pan et al., 2023; Do et al., 2024). In this study, we examine four scenarios that increasingly

[7] github.co/gkamradt/LLMTest_NeedleInAHaystack

alter the context to convert the original narratives to a more 'symbolic' format. '*Original*' are synthetic narratives using NBA team profiles; '*Scrambled*' mismatch NBA players and teams; '*Fictional*' replaces NBA player names with imaginary ones from FIFA; and '*Symbolic*' substitutes players with labels like 'Player 1, 2, 3…'. These changes are not expected to impact the model's ability to understand scoring actions or attributing them to teams. The actions described in play-by-plays remain unchanged as they characterize relationships between symbols, such as 'Player 1 assists Player 2.' This method makes the game input less reliant on natural language and more on symbolic elements.

We examine the models' ability to calculate team points using synthetic narratives and present the DCA scores in Figure 4. Our findings indicate that replacing original players with fictional names or indices can variably impact model performance. Models such as Llama3-8b-Inst and Gemini-Pro-1.5 score too low for us to draw definitive conclusions. Even high-performing models such as GPT-4o and Llama3-70b-Inst experience some decline in performance. Claude-3-Opus has demonstrated greater resilience than GPT-4o when faced with altered contexts. These results suggest that while LLMs have substantial reasoning skills, they rely on natural language context to perform well; thus, symbolic reasoning has room for improvement.

6 Conclusion

We investigate LLMs' analytical reasoning abilities, using divide-and-conquer strategies to determine where they perform best in analyzing sports narratives. With SportsGen, we further generate a variety of narratives that vary in style, complexity, and detail to better assess LLM performance. Our research not only paves the way for future LLM assessments but also sets an understanding of how LLMs may handle complex, longitudinal data.

7 Limitations

This study offers important insights into how LLMs handle sports narratives and there are a few limitations to consider. The use of synthesized narratives from SportsGen, while valuable for controlled experimentation, may not capture all the nuances of natural game commentary. Differences between our synthetic data and real-world narratives could influence how well findings generalize to natural settings. Additionally, our analysis is confined to basketball, a sport with frequent scoring events, which may not be representative of other sports that feature different dynamics and scoring patterns. Therefore, the broader applicability of our results to such diverse contexts is yet to be explored. Our future work may explore qualitative aspects such as strategic analysis or player performance that go beyond numerical scores. While acknowledging these limitations, our study still enriches the discussion about applying advanced reasoning techniques to sports narratives and sets the stage for future research in this exciting area.

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *Preprint*, arXiv:1905.13319.

Anthropic. 2024. Introducing the next generation of claude. `https://www.anthropic.com/news/claude-3-family`. Accessed on: Mar 4, 2024.

Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. 2023. Language models enable simple systems for generating structured views of heterogeneous data lakes. *Preprint*, arXiv:2304.09433.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2024. The reversal curse: Llms trained on "a is b" fail to learn "b is a". *Preprint*, arXiv:2309.12288.

Kaj Bostrom, Xinyu Zhao, Swarat Chaudhuri, and Greg Durrett. 2021. Flexible generation of natural language deductions.

Ryan Burnell, Han Hao, Andrew R. A. Conway, and Jose Hernandez Orallo. 2023. Revealing the structure of language model capabilities.

Jiaao Chen, Xiaoman Pan, Dian Yu, Kaiqiang Song, Xiaoyang Wang, Dong Yu, and Jianshu Chen. 2023. Skills-in-context prompting: Unlocking compositionality in large language models. *Preprint*, arXiv:2308.00304.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2022. Finqa: A dataset of numerical reasoning over financial data. *Preprint*, arXiv:2109.00122.

Kewei Cheng, Nesreen K. Ahmed, Theodore Willke, and Yizhou Sun. 2024. Structure guided prompt: Instructing large language model in multi-step reasoning by exploring graph structure of the text.

Jennifer Chu-Carroll, Andrew Beck, Greg Burnham, David OS Melville, David Nachman, A. Erdem Özcan, and David Ferrucci. 2024. Beyond llms: Advancing the landscape of complex reasoning.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Abhishek Divekar and Greg Durrett. 2024. Synthesizrr: Generating diverse datasets with retrieval augmentation. *Preprint*, arXiv:2405.10040.

Quyet V. Do, Tianqing Fang, Shizhe Diao, Zhaowei Wang, and Yangqiu Song. 2024. ConstraintChecker: A plugin for large language models to reason on commonsense knowledge bases. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 714–731, St. Julian's, Malta. Association for Computational Linguistics.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. 2024. Gtbench: Uncovering the strategic reasoning limitations of llms via game-theoretic evaluations.

Amosse Edouard, Elena Cabrio, Sara Tonelli, and Nhan Le-Thanh. 2017. You'll never tweet alone: Building sports match timelines from microblog posts. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 214–221, Varna, Bulgaria. INCOMA Ltd.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Gregory Furman, Edan Toledo, Jonathan Shock, and Jan Buys. 2022. A sequence modelling approach to question answering in text-based games. In *Proceedings of the 3rd Wordplay: When Language Meets Games Workshop (Wordplay 2022)*, pages 44–58, Seattle, United States. Association for Computational Linguistics.

Vedant Gaur and Nikunj Saunshi. 2023. Reasoning in large language models through symbolic math word problems. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5889–5903, Toronto, Canada. Association for Computational Linguistics.

GeminiTeam. 2023. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the Wikipedia current events portal. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1302–1308, Online. Association for Computational Linguistics.

Yi Gu, Shunyu Yao, Chuang Gan, Josh Tenenbaum, and Mo Yu. 2022. Revisiting the roles of "text" in text games. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6867–6876, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wesley H. Holliday and Matthew Mandelkern. 2024. Conditional and modal reasoning in large language models.

Yebowen Hu, Kaiqiang Song, Sangwoo Cho, Xiaoyang Wang, Hassan Foroosh, Dong Yu, and Fei Liu. 2024a. Can large language models do analytical reasoning? *Preprint*, arXiv:2403.04031.

Yebowen Hu, Kaiqiang Song, Sangwoo Cho, Xiaoyang Wang, Hassan Foroosh, Dong Yu, and Fei Liu. 2024b. Sportsmetrics: Blending text and numerical data to understand information fusion in llms. *Preprint*, arXiv:2402.10979.

Yebowen Hu, Xiaoyang Wang, Wenlin Yao, Yiming Lu, Daoan Zhang, Hassan Foroosh, Dong Yu, and Fei Liu. 2024c. Define: Enhancing llm decision-making with factor profiles and analogical reasoning. *Preprint*, arXiv:2410.01772.

Zhiting Hu and Tianmin Shu. 2023. Language models, agent models, and world models: The law for machine reasoning and planning.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey.

Kuan-Hao Huang, Chen Li, and Kai-Wei Chang. 2020. Generating sports news from live commentary: A Chinese dataset for sports game summarization. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 609–615, Suzhou, China. Association for Computational Linguistics.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446.

Hong Jin Kang, Fabrice Harel-Canada, Muhammad Ali Gulzar, Violet Peng, and Miryung Kim. 2024. Human-in-the-loop synthetic text data inspection with provenance tracking. *Preprint*, arXiv:2404.18881.

Osman Doruk Kicikoglu, Richard Bartle, Jon Chamberlain, Silviu Paun, and Massimo Poesio. 2020. Aggregation driven progression system for GWAPs. In *Workshop on Games and Natural Language Processing*, pages 79–84, Marseille, France. European Language Resources Association.

Logan Lebanoff, Bingqing Wang, Zhe Feng, and Fei Liu. 2021. Modeling endorsement for multi-document abstractive summarization. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 119–130, Online and in Dominican Republic. Association for Computational Linguistics.

Soochan Lee and Gunhee Kim. 2023. Recursion of thought: A divide-and-conquer approach to multi-context reasoning with language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 623–658, Toronto, Canada. Association for Computational Linguistics.

Bryan Li, Tamer Alkhouli, Daniele Bonadiman, Nikolaos Pappas, and Saab Mansour. 2024a. Eliciting better multilingual structured reasoning from llms through code.

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

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. 2024c. Chain of thought empowers transformers to solve inherently serial problems. *Preprint*, arXiv:2402.12875.

Yixin Liu, Avi Singh, C. Daniel Freeman, John D. Co-Reyes, and Peter J. Liu. 2023. Improving large language model fine-tuning for solving math problems. *Preprint*, arXiv:2310.10047.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *Preprint*, arXiv:2209.14610.

Stephanie M. Lukin, editor. 2020. *Workshop on Games and Natural Language Processing*. European Language Resources Association, Marseille, France.

Yuning Mao, Yanru Qu, Yiqing Xie, Xiang Ren, and Jiawei Han. 2020. Multi-document summarization with maximal marginal relevance-guided reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1737–1751, Online. Association for Computational Linguistics.

Jack Merullo, Luke Yeh, Abram Handler, Alvin Grissom II, Brendan O'Connor, and Mohit Iyyer. 2019. Investigating sports commentator bias within a large corpus of American football broadcasts. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6355–6361, Hong Kong, China. Association for Computational Linguistics.

MetaAI. 2024. Introducing meta llama 3: The most capable openly available llm to date. `https://ai.meta.com/blog/meta-llama-3/`. Large language model.

OpenAI. 2024. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`. Large language model.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *Preprint*, arXiv:2103.07191.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with entity modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.

Thomas Savage, Ashwin Nayak, Robert Gallo, Ekanath Rangan, and Jonathan H Chen. 2023. Diagnostic reasoning prompts reveal the potential for large language model interpretability in medicine. *Preprint*, arXiv:2308.06834.

Xiaoming Shi, Siqiao Xue, Kangrui Wang, Fan Zhou, James Y. Zhang, Jun Zhou, Chenhao Tan, and Hongyuan Mei. 2023. Language models can improve event prediction by few-shot abductive reasoning.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2023. Musr: Testing the limits of chain-of-thought with multistep soft reasoning.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *Preprint*, arXiv:2310.16049.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, and Andrew Dai. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Preprint*, arXiv:2206.04615.

Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. 2023. Does synthetic data generation of llms help clinical text mining? *Preprint*, arXiv:2303.04360.

Craig Thomson, Ehud Reiter, and Barkavi Sundararajan. 2023. Evaluating factual accuracy in complex data-to-text. *Computer Speech & Language*, 80:101482.

Chris van der Lee, Emiel Krahmer, and Sander Wubben. 2017. PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 95–104, Santiago de Compostela, Spain. Association for Computational Linguistics.

Veniamin Veselovsky, Manoel Horta Ribeiro, Akhil Arora, Martin Josifoski, Ashton Anderson, and Robert West. 2023. Generating faithful synthetic data with large language models: A case study in computational social science. *Preprint*, arXiv:2305.15041.

Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D. Goodman. 2023a. Hypothesis search: Inductive reasoning with language models. *Preprint*, arXiv:2309.05660.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting.

Zhe Wang, Petar Veličković, Daniel Hennes, Nenad Tomašev, Laurel Prince, Michael Kaisers, Yoram Bachrach, Romuald Elie, Li Kevin Wenliang, Federico Piccinini, William Spearman, Ian Graham, Jerome Connor, Yi Yang, Adrià Recasens, Mina Khan, Nathalie Beauguerlange, Pablo Sprechmann, Pol Moreno, Nicolas Heess, Michael Bowling, Demis Hassabis, and Karl Tuyls. 2023b. Tacticai: an ai assistant for football tactics. *Preprint*, arXiv:2310.10553.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Siwei Yang, Bingchen Zhao, and Cihang Xie. 2024a. Aqa-bench: An interactive benchmark for evaluating llms' sequential reasoning ability.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024b. Do large language models latently perform multi-hop reasoning?

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen tau Yih, and Xin Luna Dong. 2024c. Crag – comprehensive rag benchmark. *Preprint*, arXiv:2406.04744.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H. Chi, and Denny Zhou. 2024. Large language models as analogical reasoners. *Preprint*, arXiv:2310.01714.

Haoxuan You, Rui Sun, Zhecan Wang, Long Chen, Gengyu Wang, Hammad Ayyubi, Kai-Wei Chang, and Shih-Fu Chang. 2023. IdealGPT: Iteratively decomposing vision and language reasoning via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11289–11303, Singapore. Association for Computational Linguistics.

Dongxu Zhang, Varun Prashant Gangal, Barrett Martin Lattimer, and Yi Yang. 2024. Enhancing hallucination detection through perturbation-based synthetic data generation in system responses. In *Proceedings of the 62th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Hongyu Zhao, Kangrui Wang, Mo Yu, and Hongyuan Mei. 2023a. Explicit planning helps language models in logical reasoning.

Yilun Zhao, Yitao Long, Hongjun Liu, Linyong Nan, Lyuhao Chen, Ryo Kamoi, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. 2023b. Docmath-eval: Evaluating numerical reasoning capabilities of llms in understanding long documents with tabular data. *Preprint*, arXiv:2311.09805.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2023. Take a step back: Evoking reasoning via abstraction in large language models. *Preprint*, arXiv:2310.06117.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. *Preprint*, arXiv:2105.07624.

Łukasz Kidziński and Trevor Hastie. 2021. Modeling longitudinal data using matrix completion. *Preprint*, arXiv:1809.08771.

A Example Prompts

In this section, we present sample prompts used in our experiments. Particularly, few-shot prompting uses existing play-by-plays to guide GPT-4o in creating synthesized game quarters. This method tends to result in repetitive scoring actions.

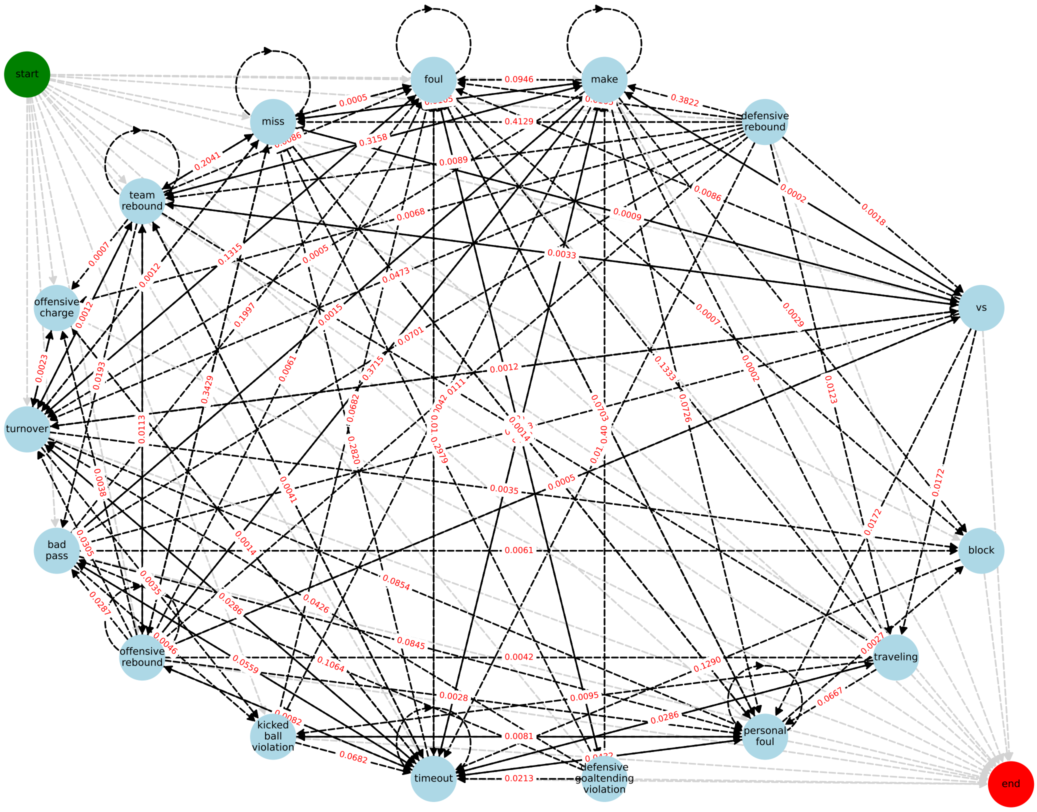

Figure 5: An example action graph created from NBA narratives. When a team has the ball, they are on offense and use actions such as passing, dribbling, shooting to score points. The defense tries to stop them by blocking, stealing, and rebounding. We refer to each sequence of these actions as a **turn**, and characterize it using a Markov graph. In the graph, each significant action is represented as a node, and transitions between nodes show how team members cooperate and execute their tactics. The node 'vs' denotes a matchup between two players, such as 'Nikola Jokic vs. Anthony Davis,' indicating their direct competition in key moments of the game. The graph begins and ends with special nodes that mark the start and end of a turn.

SportsGen Narratives (S:NS Ratio = 1:3)

<Example 1>
3:20 Zach LaVine secured the defensive rebound.
3:20 Nikola Vucevic performs a dunk with an assist from Lonzo Ball.
2:59 LaMelo Ball fails to make a 17-foot jump shot.
2:56 Cody Martin secures an offensive rebound.
2:38 Mark Williams shoots and scores a flagrant 1 free throw.
2:34 Mark Williams committed a personal foul.
2:33 Lonzo Ball secures a defensive rebound.
2:19 DeMar DeRozan misses a 20-foot jumper.
2:17 Miles Bridges secures a defensive rebound.
2:14 Mark Williams fails to make a 17-foot pullup jump shot.

<Example 2>
3:00 Jonathan Kuminga executes a tip shot.
2:57 Franz Wagner secured the defensive rebound.
2:46 Cole Anthony hit a 26-foot three pointer.
2:39 Stephen Curry fails to make a 29-foot three-point shot.
2:38 Golden State Warriors offensive team rebounded the ball.
2:37 Franz Wagner secured the defensive rebound.
2:22 Cole Anthony fails to make a 10-foot two-point shot.
2:19 Stephen Curry secures the defensive rebound.
2:00 Stephen Curry fails to connect on a 27-foot three-point shot.
1:57 Wendell Carter Jr. fails to make a 17-foot pullup jump shot.

Figure 6: SportsGen offers a practical option for creating realistic sports narratives.

Few-Shot Prompting Narratives

```
<Example 1>
2:30   Clint Capela makes 2-foot layup (Trae Young assists)
2:15   Jaylen Brown makes 3-point jumper (Derrick White assists)
2:00   Dejounte Murray makes 20-foot jumper
1:45   Jayson Tatum makes 18-foot jumper
1:30   Trae Young makes 3-point jumper
1:15   Kristaps Porzingis makes 10-foot jumper
1:00   Dejounte Murray makes 18-foot jumper
0:45   Jaylen Brown makes 2-foot layup (Jayson Tatum assists)
0:30   Trae Young makes 3-point jumper
0:15   Jayson Tatum makes 20-foot jumper

<Example 2>
8:13   Rui Hachimura defensive rebound
8:00   D'Angelo Russell makes 20-foot jumper
7:45   Devin Vassell makes 25-foot three point jumper (Tre Jones assists)
7:30   LeBron James makes driving layup
7:15   Keldon Johnson makes 10-foot jumper
7:00   Anthony Davis makes 15-foot jumper
6:45   Victor Wembanyama makes 12-foot jumper
6:30   Austin Reaves makes 25-foot three point jumper (D'Angelo Russell assists)
6:15   Devin Vassell misses 20-foot jumper
6:13   Anthony Davis defensive rebound
```

Figure 7: Few-shot prompting uses existing play-by-plays to guide GPT-4o in creating synthesized game quarter. This method tends to result in repetitive scoring actions.

An Example Showing a Successful Reasoning Attempt by GPT-4o

```
To determine the total points scored by each team, we need to examine the play-by-play
descriptions and identify which team scored the points in each play. Here's the breakdown:

1. Jakob Poeltl makes layup (Keldon Johnson assists) - San Antonio Spurs score 2 points.
2. Gordon Hayward makes 15-foot pullup jump shot - Charlotte Hornets score 2 points.
3. Gordon Hayward makes free throw 1 of 1 - Charlotte Hornets score 1 point.
4. Terry Rozier makes 25-foot three point jumper (Mason Plumlee assists) - Charlotte Hornets
score 3 points.
5. Jalen McDaniels makes layup (Terry Rozier assists) - Charlotte Hornets score 2 points.
6. Devin Vassell makes free throw 1 of 2 - San Antonio Spurs score 1 point.
7. Devin Vassell makes free throw 2 of 2 - San Antonio Spurs score 1 point.
8. Mason Plumlee makes alley oop dunk shot (James Bouknight assists) - Charlotte Hornets score
2 points.
9. Gordon Hayward makes free throw 1 of 2 - Charlotte Hornets score 1 point.
10. Gordon Hayward makes free throw 2 of 2 - Charlotte Hornets score 1 point.

Now, let's sum up the points for each team:

- Charlotte Hornets:
 2 (Hayward) + 1 (Hayward FT) + 3 (Rozier) + 2 (McDaniels) + 2 (Plumlee) + 1 (Hayward FT) + 1
(Hayward FT) = 12 points

- San Antonio Spurs:
 2 (Poeltl) + 1 (Vassell FT) + 1 (Vassell FT) = 4 points

Here is the final result in JSON format:

    {
      "Charlotte Hornets": 12,
      "San Antonio Spurs": 4
    }
```

Figure 8: An example showing a successful reasoning attempt by GPT-4o.

An Example Showing an Unsuccessful Reasoning Attempt by LLama-3-8B-Instruct

```
To determine the total points scored by each team, I'll analyze the play-by-play descriptions
and identify the scoring plays.

Charlotte Hornets:
1. Gordon Hayward makes 15-foot pullup jump shot (5:53) - 2 points
2. Terry Rozier makes 25-foot three point jumper (5:32) - 3 points
3. Jalen McDaniels makes layup (5:06) - 2 points
4. Mason Plumlee makes alley oop dunk shot (4:43) - 2 points
5. Gordon Hayward makes free throw 1 of 2 (4:24) - 1 point
6. Gordon Hayward makes free throw 2 of 2 (4:24) - 1 point
Total points for Charlotte Hornets: 2 + 3 + 2 + 2 + 1 + 1 = 11 points

San Antonio Spurs:
1. Jakob Poeltl makes layup (6:10) - 2 points
2. Devin Vassell makes free throw 1 of 2 (4:55) - 1 point
3. Devin Vassell makes free throw 2 of 2 (4:55) - 1 point
Total points for San Antonio Spurs: 2 + 1 + 1 = 4 points

Here is the final result in JSON format:

    {
      "Charlotte Hornets": 11,
      "San Antonio Spurs": 4
    }
```

Figure 9: An example showing an unsuccessful reasoning attempt by LLama-3-8B-Instruct.

System Message

```
You are a helpful assistant tasked with analyzing sports games. You have been given a
play-by-play breakdown of an NBA basketball game between two teams.

The "Time" column shows the exact time on the game clock when each play took place. The game
clock counts down, so this column displays times in a descending order.

The "Play" column describes the action that happened at the respective times. It provides
details of specific plays, movements, and outcomes on the court.

Team players are listed in two rows, each row representing one of the two basketball teams
involved in the game.
```

User Message

```
Analyze the team-player affiliations and play-by-play descriptions below to determine the
total points scored by each team.

Please explain your reasoning step by step and provide the final results in the JSON format.
Starting with: {Toronto Raptors: 0, Indiana Pacers: 0}

#Team-Player Affiliations:

Toronto Raptors: Og Anunoby, Fred Vanvleet, Norman Powell, Aron Baynes, Chris Boucher, Deandre
Bembry, Stanley Johnson, Yuta Watanabe, Malachi Flynn, Matt Thomas, Terence Davis

Indiana Pacers: Myles Turner, Justin Holiday, Doug Mcdermott, Jeremy Lamb, Malcolm Brogdon,
Domantas Sabonis, TJ McConnell, Aaron Holiday

#Play-by-Play Descriptions:

Time    Play
12:00   Aron Baynes vs. Myles Turner (Fred VanVleet gains possession)
11:38   Myles Turner blocks Stanley Johnson's two point shot
11:35   Justin Holiday defensive rebound
11:24   Fred VanVleet blocks Malcolm Brogdon's two point shot
11:22   Fred VanVleet defensive rebound
11:20   Aron Baynes makes layup (Fred VanVleet assists)
11:04   Stanley Johnson shooting foul
11:04   Myles Turner misses free throw 1 of 2
11:04   Pacers offensive team rebound
```

Figure 10: Our research uses analytical reasoning to accurately calculate team points from sports narratives.

```
System Message

Your task is to generate detailed play-by-play descriptions for one quarter of an NBA game,
capturing the dynamic flow of the game. This includes player actions (e.g., shooting, blocking,
stealing, etc.), key moments (e.g., turnovers, fouls, timeouts), player substitutions, and more.
Please ensure the narrative is engaging and reflects the pace and intensity of an NBA game.
```

```
User Message

Generate a full quarter of an NBA game using the detailed player-team affiliations and
play-by-play data provided. Ensure the following requirements are met:

1. Typically, a quarter of NBA games feature 70 to 150 plays, with 20 to 40 being scoring
   plays, and a scoring density between 0.25 and 0.35.
2. A team's efficiency score, showing potential points per 100 possessions,
   can impact game outcomes.
3. Adhere to the example play-by-play description format provided.
4. Ensure a natural and logical progression between plays.
5. Separate each column in the play-by-play data using a tab character.

#Team-Player Affiliations:

Team-1: player-1, player-2, player-3, player-4, player-5, player-6, player-7, ...
Team-2: player-11, player-12, player-13, player-14, player-15, player-16, player-17, ...

#Team Efficiency Scores:

Team-1: 93
Team-2: 87

#Example Play-by-Play Descriptions:

Time    Team      Play
12:00   Team-2    player-3 vs. player-14 (player-13 gains possession)
11:46   Team-2    player-12 makes alley oop layup (player-14 assists)
11:26   Team-1    player-2 makes 2-foot layup (player-3 assists)
11:08   Team-2    player-12 makes driving layup
10:43   Team-1    player-6 misses 21-foot step back jumpshot
10:41   Team-1    player-2 offensive rebound
10:37   Team-2    player-13 misses 25-foot three point jumper
10:34   Team-2    player-15 defensive rebound
10:27   Team-2    player-14 shooting foul
10:27   Team-2    player-7 misses free throw 1 of 2
10:27   Team-1    team-1 offensive team rebound
10:27   Team-2    player-12 makes free throw 2 of 2
10:14   Team-2    player-18 misses 23-foot three point jumper
10:11   Team-1    player-6 defensive rebound
…

Based on the following team-player affiliations, generate the play-by-plays for the first
quarter of the game.

#Team-Player Affiliations:

Memphis Grizzlies: Jordan Goodwin, Scotty Pippen Jr., Jake LaRavia,
                   Vince Williams Jr., Santi Aldama
Miami Heat: Bam Adebayo, Jimmy Butler, Jamal Cain, Haywood Highsmith, Jaime Jaques Jr.

#Team Efficiency Scores:

Memphis Grizzlies: 92
Miami Heat: 87

#Play-by-Play Descriptions:
```

Figure 11: Few-shot prompting uses existing play-by-plays to guide GPT-4o in creating synthesized game quarters.