

Evaluating Multimodal Large Language Models on Video Captioning via Monte Carlo Tree Search

Lin hao Yu¹, Xinguang Ji², Yahui Liu², Fanheng Kong², Chenxi Sun²,
Jingyuan Zhang², Hongzhi Zhang², Victoria W.², Fuzheng Zhang², Deyi Xiong^{1*}
¹TJUNLP Lab, College of Intelligence and Computing, Tianjin University, Tianjin, China
²Independent Researcher
linhaoyu@tju.edu.cn, dyxiong@tju.edu.cn

Abstract

Video captioning can be used to assess the video understanding capabilities of Multimodal Large Language Models (MLLMs). However, existing benchmarks and evaluation protocols suffer from crucial issues, such as inadequate or homogeneous creation of key points, exorbitant cost of data creation, and limited evaluation scopes. To address these issues, we propose an automatic framework, named AutoCaption, which leverages Monte Carlo Tree Search (MCTS) to construct numerous and diverse descriptive sentences (*i.e.*, key points) that thoroughly represent video content in an iterative way. This iterative captioning strategy enables the continuous enhancement of video details such as actions, objects' attributes, environment details, etc. We apply AutoCaption to curate MCTS-VCB, a fine-grained video caption benchmark covering video details, thereby enabling a comprehensive evaluation of MLLMs on the video captioning task. We evaluate more than 20 open- and closed-source MLLMs of varying sizes on MCTS-VCB. Results show that MCTS-VCB can effectively and comprehensively evaluate the video captioning capability, with Gemini-1.5-Pro achieving the highest F1 score of 71.2. Interestingly, we fine-tune InternVL2.5-8B with the AutoCaption-generated data, which helps the model achieve an overall improvement of 25.0% on MCTS-VCB and 16.3% on DREAM-1K, further demonstrating the effectiveness of AutoCaption. The code and data are available at <https://github.com/tjunlp-lab/MCTS-VCB>.

1 Introduction

Video captioning is a crucial task that expects to obtain a complete, accurate, and low-hallucination textual description for a given video. Recently, this task is widely applied to showcase the multidimensional and comprehensive understanding

*Corresponding author.

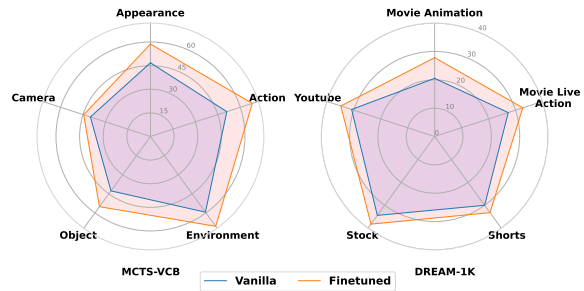


Figure 1: Comparison of MCTS-VCB against DREAM-1K with InternVL2.5-8B (Chen et al., 2024b) after fine-tuning on caption data generated by AutoCaption.

capabilities of MLLMs across various video categories (Wang et al., 2024d; Li et al., 2024b; Hong et al., 2025).

Previous video captioning benchmarks (Wang et al., 2024a; Chen and Dolan, 2011; Xu et al., 2016; Wang et al., 2019) commonly involve creating key points (*i.e.*, descriptive sentences) from video clips and comparing them to the captions generated by MLLMs. There are *three* main challenges by using the above strategy: 1) Once the created key points are inadequate or homogeneous, the evaluation can be overstated or inaccurate. For example, if the created key points are tend to general or action-oriented descriptions, they would not be suitable for evaluating the perception of environment aspects. 2) Owing to the potential incompleteness of annotations, it is challenging to perform a full-scope evaluation of the video captioning capability of MLLMs. 3) Either manually annotated captions or extracted from manually constructed captions are time-consuming and expensive.

To alleviate the above-mentioned challenges, we introduce AutoCaption, an automated framework that utilizes Monte Carlo Tree Search (MCTS) to iteratively construct plenty of diverse key points, corresponding to the given video content. AutoCaption can enrich the description of video details (*e.g.*, color, quantity, position, material, and other

attributes of objects) through a continuous procedure of predefined actions using a tree search approach. In this manner, AutoCaption can eliminate the need for labor-intensive manual annotations, and address the challenge of creating insufficient or homogeneous key points.

Based on AutoCaption, we have established a video caption benchmark, named MCTS-VCB, which can be applied to assess the video captioning capabilities of MLLMs. Our MCTS-VCB benchmark consists of 1,765 videos that are sourced from 10 distinct video categories. Each video contains an average of 122 key points that are categorized into five dimensions, involving a broad evaluation scope.

After evaluating 20+ open-source and closed-source MLLMs on MCTS-VCB, we observe that the performance of different MLLMs varies a lot, especially on some particular video categories. To further validate the effectiveness of AutoCaption, we construct a dataset of around 10K samples with AutoCaption to fine-tune existing MLLMs (*e.g.*, InternVL2.5-8B (Chen et al., 2024b)). Notably, we find that the synthesized data can significantly boost existing MLLMs (*e.g.*, InternVL2.5-8B +25.0% on MCTS-VCB and +16.3% on DREAM-1K (Wang et al., 2024a)), as shown in Figure 1.

In summary, our contributions are as follows:

1. We propose an automated captioning framework, named AutoCaption, that leverages Monte Carlo Tree Search (MCTS) to iteratively analyze and describe video details, achieving a comprehensive understanding of the video content.
2. We develop a fine-grained video captioning evaluation benchmark MCTS-VCB for MLLMs. To the best of our knowledge, it is the first MCTS based benchmark in this field.
3. We prove that AutoCaption can be applied to generate data for fine-tuning existing MLLMs. Experimental results indicate that it is a more efficient method to boost the captioning capability of MLLMs.

2 Related Work

Data Synthesis and Augmentation Synthetic data can effectively alleviate the substantial demand for large-scale training data for LLM (Zhou et al., 2024). There are two commonly used

approaches improving the capabilities of LLMs: data augmentation and data synthesis (Wang et al., 2024b). The former involves further processing of existing data, including data labeling (Alizadeh et al., 2023; Törnberg, 2023; Zhu et al., 2023), data transformation (Dai et al., 2023; Dunlap et al., 2023; Chen et al., 2023; Shi and Xiong, 2024; Yu et al., 2024), to create fine-grained data or augmented data. The latter typically curates data from scratch with common methods, including model distillation (Xu et al., 2024; Xin et al., 2024) and self-improvement (Wang et al., 2023; Chen et al., 2024c). Self-improvement usually involves multiple refinements on sampled data to obtain high-quality synthetic data. Theoretically, tree-search methods like MCTS (Yao et al., 2023; Hao et al., 2023) can sample and iterate over an infinite number of paths, that can not only ensure high-quality data but also significantly increase data diversity. Owing to this reason, MCTS has been widely applied to complex reasoning tasks (Chen et al., 2024a; Zhang et al., 2024a; Qi et al., 2024; Zhang et al., 2024b; Luo et al., 2024; Wang et al., 2024c) and planning (Li et al., 2025). Inspired by its advantage in reasoning, we leverage MCTS to continuously capture the details of videos, ensuring the maximum coverage of video content.

Video Understanding Benchmarks Video understanding benchmarks can be broadly categorized into two groups. For the former, there are benchmarks that evaluate different video understanding tasks, such as spatial-temporal reasoning (Yu et al., 2019; Jang et al., 2017; Zhang et al., 2020b; Tang et al., 2021), causal reasoning (Xiao et al., 2021), and long video reasoning (Song et al., 2024a,b). For the latter, there are comprehensive evaluation frameworks like MMBench (Liu et al., 2024) and MVBench (Li et al., 2024a), which focus on different video understanding tasks by conducting question answering (QA) on various videos. However, these comprehensive frameworks usually struggle to provide a comprehensive evaluation over video captioning.

Video Captioning Benchmarks As a fundamental task of video understanding, it has been drawing attention to building video caption benchmarks. Initially, these benchmarks are specific to certain domains, such as social media (Gella et al., 2018), movies (Rohrbach et al., 2015, 2017), and cooking (Regneri et al., 2013; Zhou et al., 2018). There are also open-domain benchmarks, such as MSR-

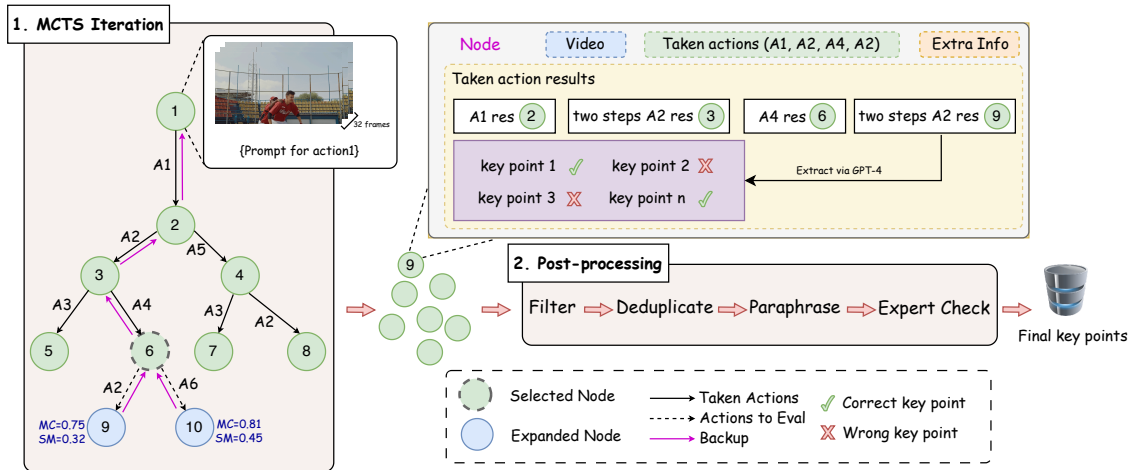


Figure 2: The overall workflow of AutoCaption, illustrating the key points generation process from a video. The “A1”-“A6” labels on the edges of search tree T denote six different actions. The MC value and SM value besides two new nodes expanded from the selected node 6 are calculated in the “Evaluation” phase.

VTT (Xu et al., 2016) and MSCV (Chen and Dolan, 2011), which annotate a video with multiple similar single-sentence descriptions. Vatex (Wang et al., 2019) is similar to these benchmarks but provides bilingual single-sentence descriptions with both Chinese and English. However, such descriptions cannot fully, accurately, and comprehensively convey the content of a video. The emergence of DREAM-1K (Wang et al., 2024a) provides a more comprehensive evaluation of video captioning through a fine-grained approach, but the key points in a video are manually annotated as “events”, with an average of only 6.3 key points per video, which may miss details of videos. Our proposed AutoCaption method, can automatically annotate 122.3 key points per video, thus providing a more comprehensive evaluation of the video captioning capability of MLLMs.

3 Preliminary

MCTS is a heuristic search algorithm for some kinds of decision processes, such as game AI, planning, and optimization problems. MCTS evaluates the potential value of different decisions by simulating multiple possible future paths in a decision tree. The core idea is to estimate the value of each node through random simulations and gradually expand and optimize the search tree.

There are four main steps in MCTS (Browne et al., 2012): 1) **Selection** Starting from the root node, child nodes are selected based on a certain strategy (e.g., UCB (Chang et al., 2005)) until a leaf node is reached. 2) **Expansion** If the selected

node is not in a terminal state, expand it by adding one or more child nodes to the selected node. 3) **Simulation** From the newly expanded node, a random simulation is performed until the end of the game or a predefined depth is reached. The result of the simulation is used to estimate the value of the node. 4) **Backpropagation** The simulation result is propagated back to all nodes along the path, while the visit count and value estimate of these nodes are updated.

AutoCaption is an automated framework aimed at generating detailed key points that fully cover the content of videos using MCTS. Utilizing the tree search feature of MCTS enables the progressive enhancement of video details such as appearance and comprehensive descriptions of the environment by taking different actions iteratively. Inspired by Qi et al. (2024), we can use a small MLLM as the generator to dig detailed information in our proposed AutoCaption framework.

4 AutoCaption

We divide AutoCaption into two essential procedures: MCTS iteration and data post-processing, as shown in Figure 2. We provide the pseudocode of AutoCaption in Algorithm 1 in appendix. In this section, we explain these two procedures in detail.

Given a video v and a small MLLM for generating video captions, we can construct a search tree T through MCTS. As shown in Figure 2, we assign the video v to the root node, each edge signifies an executing action a , and a child node s embodies a state resulting from a . The trajectory $t(s)$ from the

root node to any child node s symbolizes the process of increasingly concentrating on imperceptible video content to produce new detailed descriptions.

4.1 MCTS Iteration

In the MCTS iteration, each root node contains a video v that needs to be described, a list of actions taken from the root node to the current node s , and a list of action results (video descriptions) generated by the corresponding actions. Additionally, each node s contains statistical information: $N(s)$ denotes the number of times that the node is visited, $MC(s)$ is the Monte Carlo value of the node, which measures the correctness of the action result obtained after executing the corresponding action at node s , and $SM(s)$ is the similarity score of the node to all other nodes in trajectory $t(s)$. Then, we follow the standard MCTS steps to conduct the iteration procedure.

Selection In the i -th iteration of MCTS, we first select a node s with the highest value (calculated in “Evaluation” phase) from all the leaf nodes $L(T)$ in the search tree T based on a variant of the PUCT algorithm (Rosin, 2011) formulated as:

$$s_i = \arg \max_{s \in L(T)} \left[Q(s, a) + c \frac{\sqrt{N_{\text{parent}}(s)}}{1 + N(s)} \right] \quad (1)$$

where $Q(s, a)$ is the state value function after taking action a , with a higher value indicating that the selected node is more likely to generate a correct detailed description when executing corresponding actions. Refer to Luo et al. (2024), c is a constant set to 0.125 that controls the exploration degree, $N_{\text{parent}}(s)$ denotes the current number of visit times to the parent node of s .

Expansion Unlike traditional MCTS, we merge “simulation” phase into the “Expansion” phase since executing an action by MMLM can be treated as “simulation”. In the i -th iteration, we expand the possible actions at node s_i selected in the previous phase. Specifically, we have defined six types of actions: *Overall Description* (A1), *Detail Description* (A2), *Temporal Perspective Description* (A3), *Spatial Perspective Description* (A4), *Background Description* (A5), and *Camera Movement Description* (A6). These actions follow a certain execution order. The *Overall Description* action is executed only once and only after the root node. To give MCTS iteration a good initialization, the *Overall*

Description action is executed by GPT-4o (OpenAI, 2024) and Gemini-1.5-Pro (Team et al., 2024). These six actions comprehensively cover the description of video content from various perspectives. Meanwhile, these actions can ensure the breadth of descriptions while progressively increasing the depth of detail through continuous iteration. In the i -th iteration, for the selected child node s_i , since we expect the generator to focus on different details when executing the corresponding action a_j , we always include the correct key points extracted from the results of previously executed actions in a_j in trajectory $t(s_i)$ into the action prompt to enforce the generator paying attention to new details. Therefore, each node s in $t(s)$ needs to store both the historical actions and action results.

For the selected node s_i , we do not expand all actions but randomly expand only two of them under the condition that the probability of sampling the *Detailed Description* action is twice that of sampling other actions. This is because, for video captioning, it is not necessary to generate multiple paths to achieve a higher completion rate as in reasoning tasks. Instead, video captioning requires a more comprehensive description of video details during executing *Detail Description* action. Additionally, for the rest of the actions, we observe that the improvement in detail brought by multiple samplings is very limited, so it is unnecessary to expand all actions at each node s_i .

In addition, to enable the generator to continuously discover new details, we have adopted a two-stage process for the *Detail Description* action: 1) First, prompting the generator to identify a new detail besides those previously found. Assuming response in this stage to be Ans_{s1} . 2) Then, we utilize Qwen2.5-7B-Instruct (Team, 2024b) to extract the details that need further observation from Ans_{s1} , along with the category of the detail and its specific attributes, integrating them into a prompt to guide the generator to continue describing new details. The two-stage process effectively reduces repetitive descriptions. The prompts corresponding to different actions are shown in Appendix A.2.

Evaluation We evaluate all the new nodes expanded during the “Expansion” phase here. The state value of each node is calculated using the following formula adopted in Luo et al. (2024):

$$Q(s, a) = \alpha^{1-MC(s)} \cdot \beta^{SM(s)} \quad (2)$$

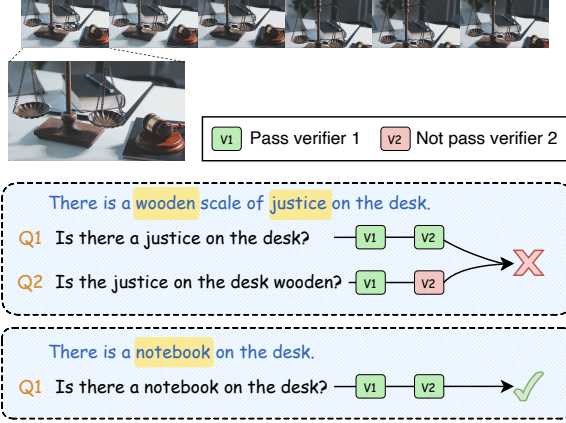


Figure 3: Illustration of the key point verification process. Texts in yellow shade are key information need to be verified. Each key information needs a verification question. “V1” and “V2” denote GPT-4o and Qwen2-VL-72B, respectively.

where α and β are values $\in (0, 1)$, used to balance the value and similarity. In Luo et al. (2024), $\alpha = 0.5$ and $\beta = 0.9$, while both α and β are set to 0.5 in our experiment since $MC(s)$ and $SM(s)$ are same important. Here, $MC(s)$ represents the Monte Carlo simulation value of the node s , and $SM(s)$ represents the similarity of the node s . The node with higher $MC(s)$ and lower $SM(s)$ will have a higher state value $Q(s, a)$. The calculations for $MC(s)$ and $SM(s)$ are as follows:

$$SM(s) = \frac{1}{|t(s)| - 1} \sum_{\substack{u \in t(s) \\ u \neq s}} \text{sim}(u, s) \quad (3)$$

$$MC(s) = \frac{N_{\text{correct}}(s)}{N_{\text{total}}(s)} \quad (4)$$

where $N_{\text{correct}}(s)$ is the number of key points passed the verification and $N_{\text{total}}(s)$ is the total number of key points extracted from node s .

The key points are fine-grained key information extracted from a node s by Qwen2-VL-72B (Team, 2025) under the prompt displayed in Figure 13 in the Appendix, ensuring that each key point contains only atomic-level subjects and indivisible predicates. We further categorize each extracted atomic-level key point into five categories: *Appearance Description*, *Action Description*, *Environment Description*, *Object Description*, and *Camera Movement and Composition*. We provide an example video with these five key point types in Figure 10 in the Appendix.

To verify each key point, we first determine the key information that needs to be verified, such as

actions, appearance, attributes of an object, etc. Then, we utilize GPT-4 (Team, 2024a) to construct a “yes/no” question for each key information needed to be verified and then use two verifiers (*i.e.*, GPT-4o and Qwen2-VL-72B) to answer them. If all yes/no questions derived from a key point are shown to be “yes” for both verifiers, we consider the key point passing the verification. Figure 3 shows the verification process of a key point. Some more explored alternative verification methods, which are not effective, are provided in Appendix A.3.

We claim that the $SM(s)$ in $Q(s, a)$ is necessary because node s with high similarity to nodes in $t(s)$ is more likely to produce repetitive results. This not only undermines data diversity but also increases unnecessary validation.

Backpropagation At the end of the i -th iteration, starting from the selected node s_i , we update the visit times $N(s)$ and state value $Q(s, a)$ of all nodes in $t(s_i)$ except node s_i according to the following backpropagation rules:

$$\begin{aligned} N(s) &\leftarrow N(s) + 1 \\ Q(s, a) &\leftarrow \frac{1}{N(s)} \sum_{u \in s.\text{children}} Q(u, a_u) \end{aligned} \quad (5)$$

where a_u is the action that leads to node u .

4.2 Data Post-processing

After the MCTS iterations over a video, we construct a search tree T for the video. We then obtain all nodes in T , where each node, except the root node, contains a description generated by the corresponding actions. These descriptions need to be post-processed (*e.g.*, filtered, deduplicated, paraphrased and human checked) to form final key points of the video. Note that the post-processing is lightweight that requires low computation resources. The post-processing detail is described in Appendix A.4.

5 MCTS-VCB

MCTS-VCB is a video caption benchmark curated with AutoCaption described in Section 4. We use Qwen2-VL-7B (Wang et al., 2024d) as the generator in MCTS iteration. We uniformly sample 64 frames from each video, and the iteration number is set to 25 since we found that more duplicated contents are produced with more iteration.

Model Name	Appearance Description	Action Description	Environment Description	Object Description	Camera Movement and Composition	Overall
Open-Source MLLMs						
InternVL2_5-8B	65.1/36.5/46.8	65.0/42.0/51.0	76.0/48.7/59.4	72.1/30.3/42.7	54.7/31.7/40.1	69.6/40.1/50.8
InternVL2_5-26B	65.2/37.1/47.3	67.6/43.4/52.8	75.6/49.1/59.5	73.6/31.5/44.1	54.4/32.0/40.3	70.1/40.9/51.7
InternVL2_5-38B	65.0/37.0/47.1	68.2/44.0/53.5	75.7/48.9/59.5	73.8/31.7/44.4	56.2/31.7/40.6	70.6/41.0/51.8
InternVL2_5-78B	65.3/38.2/48.2	67.7/44.7/53.8	76.4/49.8/60.3	73.5/31.8/44.4	55.6/32.0/40.6	70.6/41.6/52.4
LLaVa_OV_Qwen2_7B	75.8/45.1/56.5	77.3/58.9/66.8	84.5/59.1/69.6	77.1/42.2/54.5	73.2/42.4/53.7	79.6/51.8/62.8
LLaVa_OV_Qwen2_72B	75.8/45.5/56.9	78.9/60.3/68.3	86.1/60.2/70.9	78.7/43.1/55.7	79.6/45.3/57.7	81.3/52.9/64.1
LLaVa_Video_Qwen2_7B	78.0/41.3/54.0	79.7/50.7/61.9	82.7/58.4/68.5	76.4/41.0/53.4	76.1/40.8/53.1	79.7/48.9/60.6
LLaVa_Video_Qwen2_72B	79.8/45.8/58.2	80.9/57.9/67.5	85.2/59.1/69.8	80.2/41.5/54.7	77.9/44.1/56.3	82.1/51.7/63.4
MiniCPM-V-2_6	75.3/40.8/52.9	73.0/57.8/64.5	82.7/58.2/68.3	78.3/38.9/52.0	75.6/ 53.9/62.9	77.7/50.4/61.2
PLLaVA-7B	47.4/28.3/35.5	47.6/37.6/42.0	69.0/37.7/48.7	61.9/17.6/27.4	37.6/20.3/26.4	55.1/30.4/39.2
PLLaVA-13B	69.9/35.9/47.4	69.5/44.0/53.9	78.0/46.2/58.0	66.7/27.3/38.8	69.8/29.5/41.4	72.5/38.7/50.4
PLLaVA-34B	72.3/33.5/45.8	73.2/44.6/55.4	78.3/46.6/58.4	73.6/27.9/40.4	65.6/26.5/37.7	74.9/38.5/50.9
Qwen2-VL-7B	74.4/43.5/54.9	75.8/57.7/65.5	80.4/60.2/68.8	77.6/40.7/53.4	71.8/34.9/46.9	77.8/50.9/61.5
Qwen2-VL-72B	71.4/45.8/55.8	75.9/57.9/65.7	80.9/60.8/69.4	78.7/42.5/55.2	73.2/37.7/49.7	77.7/52.1/62.4
Tarsier-7B	76.3/34.0/47.1	69.7/49.7/58.0	85.7/56.4/68.0	72.9/35.5/47.8	58.7/36.0/44.6	76.7/45.2/56.9
Tarsier-34B	79.8/34.3/48.0	75.5/51.5/61.2	86.8/55.9/68.0	76.8/36.4/49.4	62.1/35.8/45.5	79.6/45.7/58.0
Llama-3-VILA1.5-8B	71.7/39.2/50.7	73.8/46.6/57.1	80.9/52.3/63.5	73.5/37.1/49.3	66.9/31.6/42.9	75.6/44.2/55.8
VILA1.5-13B	69.4/39.0/50.0	70.5/50.3/58.7	79.3/51.6/62.5	74.0/34.6/47.2	66.8/35.2/46.1	73.9/44.4/55.4
VILA1.5-40B	74.5/42.3/54.0	76.1/55.8/64.4	82.1/53.4/64.7	79.0/37.9/51.2	73.1/39.8/51.6	78.3/47.6/59.2
Closed-Source MLLMs						
Gemini-1.5-Pro	78.2/ 59.9/67.9	82.4/64.8/72.5	81.0/ 70.7/75.5	82.0/56.0/66.5	79.9/59.0/67.9	80.9/ 63.6/71.2
GPT-4o	84.9/47.2/60.6	86.4/62.7/72.7	80.9/ 67.1/73.4	87.3/50.6/64.1	81.9/40.8/54.5	83.8/57.5/68.2

Table 1: Performance on MCTS-VCB across different key point categories. Each cell contains “Precision / Recall / F1 Score”. The best, second-best, third best results are **bold underlined**, **bold** and underlined respectively.

Data Source Initially, we randomly select 2,000 video clips from open-source data to form the video candidates for MCTS-VCB. Among these videos, we extract visual and textual information to automatically exclude clips lacking distinct actions or subjects. The majority of excluded clips feature extensive special effects, an absence of clear subjects, or indeterminate objectives. The filtering process is detailed in Appendix A.1.

MCTS-VCB Statistics The average duration of these clips is 12.39 seconds, which ensures a balanced amount of information is presented in each, making them suitable for evaluating captioning tasks. The distribution of video durations is shown in Figure 7 in Appendix B.1. MCTS-VCB encompasses 10 categories of videos, 5 categories of video content key points, enabling a thorough evaluation of MLLMs’ capability to describe various video genres. The detailed explanation and statistics of different video categories and key point categories are illustrated in Appendix B.1.

6 Experiments

We evaluated 21 open- and closed-source MLLMs, with sizes ranging from 7B to 78B. We conducted a comprehensive evaluation of these MLLMs across different dimensions. The model cards of the 21 evaluated MLLMs and corresponding experimental settings (such as the number of frames and used prompts) can be found in Appendix B.2. To further validate the effectiveness of AutoCaption, we

conducted a series of augmented experiments.

Evaluation Metrics We used settings similar to those adopted in the evaluation method proposed by DREAM-1K (Wang et al., 2024a). First, for each evaluated model, we used the prompt shown in Figure 13 to guide Qwen2-VL-72B in extracting key points KP_{model} from the model-generated description D_{model} . For each video, MCTS-VCB contains the verified video key points KP_{ref} . Then, we calculated the ratio of key points in KP_{model} that are entailed in KP_{ref} as **Precision**, and the ratio of key points in KP_{ref} that are entailed in D_{model} as **Recall**. Finally, we calculated **F1** by $\frac{2*P*R}{P+R}$. This evaluation method allows us to assess both the accuracy and completeness of the responses from all evaluated MLLMs. We used Qwen2.5-72B-Instruct to be the judge model and the prompt is displayed in Figure 14.

6.1 Evaluation Results

Performance across different key point types.

Table 1 presents the performance of different MLLMs on MCTS-VCB. Overall, Gemini-1.5-Pro excels in multiple categories, achieving 4 wins out of 5 categories and overall best performance, demonstrating its robust capability in handling video caption tasks. GPT-4o follows closely, performing well in the action description category, highlighting its strengths in these specific tasks. Though the gap between open- and closed-source MLLMs still exists, LLaVa_OV_Qwen2_72B performs best among the open-source MLLMs with

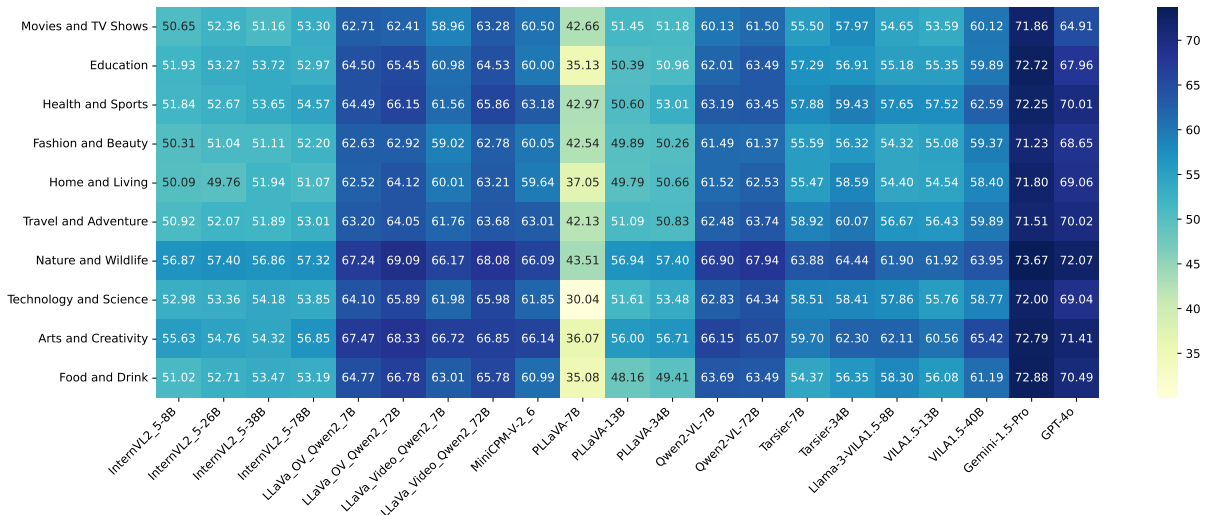


Figure 4: F1 score across different video categories.

an overall F1 score of 64.1. Furthermore, we can observe that performance generally improves along with increasing the model size. Surprisingly, the InternVL series MLLMs demonstrate merely average performance; this will be further investigated Section 6.2.

Performance across different video categories.

Figure 4 shows the performance of various MLLMs across different video categories. It is evident that MLLMs achieve better results in the “nature and Wildlife” and “Arts and Creativity” categories, possibly due to these videos’ real-life resemblance, facilitating model learning. In contrast, performance in the remaining categories varies slightly. Although videos in “Movie and TV Shows” offer easily accessible training data with clear subjects and rich content, these types of videos often contain more detailed information. In our MCTS-VCB benchmark, these detailed information is captured thoroughly. However, existing MLLMs struggle to process the detailed information presented in these videos, leading to less impressive performance.

Performance on different frame numbers. As shown in Figure 3, we evaluated four MLLMs with a similar model size to investigate the performance when varying numbers of input frames. During the training process, LLaVa_OV_Qwen2_7B is trained with a maximum frame number of 32 while LLaVa_Video_Qwen2_7B and InternVL2_5-8B are trained with a maximum frame number of 64. Consequently, we observe that the performance of these three models increases as the number of frames increases, until the frame number

exceeds the maximum frame number used during training. MiniCPM-V-2_6 exhibits the strongest performance at 16 frames, which we attribute to its lack of training on sufficiently long contexts. This suggests that, as more input frames are provided, models can acquire more information and detect finer details, but this requires prior training on long context length.

Performance across different similarity threshold.

We explored the impact of using different similarity thresholds for merging data from each node on the evaluation results. We conducted experiments by uniformly sampling 11 similarity thresholds in [0.7, 0.9]. The results indicate that as the similarity threshold increases, the MLLMs’ performance also improves. However, this is an expected phenomenon, and the relative ranking remains unchanged, demonstrating the accuracy of the evaluation metrics. Results of this group of experiments are provided in Appendix B.3.

Human Consistency

We selected 100 videos from the 1,765 videos in MCTS-VCB and used 3 MLLMs to check their consistency with human evaluations across various methods. For each MLLM and video, annotators reviewed key points from AutoCaption and the MLLM’s output to assess relationship between these two sets of key points (entailment, neutral, contradiction) and calculated the F1 score.

We employed four different evaluation metrics, two of which are based on n-gram overlap (BertScore (Zhang et al., 2020a) and ROUGE (Lin, 2004)), and the other two are based on judgment

InternVL2_5-8B			
Method	Kendall's τ	Spearman's ρ	Pearson r
Bert-Score	18.8	27.9	27.1
ROUGE	17.3	23.5	24.7
CLAIR	41.8	53.2	58.2
AutoDQ	32.4	47.4	43.6
MCTS-VCB (ours)	68.4	87.6	88.5
LLaVa_OV_Qwen2_72B			
Method	Kendall's τ	Spearman's ρ	Pearson r
Bert-Score	18.9	28.5	31.3
ROUGE	20.5	30.3	33.6
CLAIR	25.6	32.1	39.2
AutoDQ	20.8	28.7	24.5
MCTS-VCB (ours)	56.8	75.0	80.5
PLLaVA-7B			
Method	Kendall's τ	Spearman's ρ	Pearson r
Bert-Score	36.3	52.0	61.8
ROUGE	36.2	51.8	57.7
CLAIR	53.0	67.7	70.4
AutoDQ	33.2	45.1	48.1
MCTS-VCB (ours)	77.8	92.1	94.8

Table 2: Correlation results (Kendall's τ , Spearman's ρ , and Pearson r) of various evaluation metrics on three models.

Models	#Frames			
	8	16	32	64
LLaVa_OV_Qwen2_7B	62.1	62.8	62.9	62.6
LLaVa_Video_Qwen2_7B	59.2	60.6	61.7	63.5
MiniCPM-V-2_6	60.3	61.2	60.2	58.8
InternVL2_5-8B	50.4	50.8	51.2	51.5

Table 3: Performance on MCTS-VCB across different input frame numbers.

models (CLAIR (Chan et al., 2023) and AutoDQ (Wang et al., 2024a)). We calculated Kendall's τ , Spearman's ρ , and Pearson r correlation scores. Below are the consistencies of the selected 3 models with human evaluation across the four assessment methods. All P-values < 0.05 . As shown in Table 2, our evaluation method ensures the highest consistency, demonstrating MCTS-VCB's superiority.

6.2 Ablation Study

AutoCaption v.s. Human Annotation We extracted 50 videos and their corresponding human-annotated results (KP_{human}) from DREAM-1K and obtained the automatic annotations (KP_{ac}) for these videos using AutoCaption. We then used GPT-4 to determine the proportion of mutual inclusion between KP_{human} and KP_{ac} . We find that 73.29% of KP_{human} appear in KP_{ac} , while only 14.98% of KP_{ac} appear in KP_{human} , demonstrating the comprehensiveness of AutoCaption in capturing video content.

MCTS v.s. Beam Search In AutoCaption, we use MCTS to iteratively obtain detailed information

about the videos. We hence investigated whether using Beam Search could construct key points with the same completeness as the AutoCaption method. We randomly selected 50 videos from MCTS-VCB. We set the beam size to 2, selecting the leaf node with the highest $Q(s, a)$ for expansion each time, with 25 rounds of expansion. After obtaining these nodes, we perform the data post-processing described in Section 4.2 to obtain the key points (KP_{bs}). We again use GPT-4 to determine the proportion of mutual inclusion between KP_{bs} and the corresponding key points of the videos in MCTS-VCB (KP_{mcts}). We find that 91.62% of KP_{bs} appear in KP_{mcts} , while only 83.43% of KP_{mcts} appear in KP_{bs} . This is because Beam Search cannot better expand the breadth of the search. Additionally, since Beam Search only selects the optimal node from the newly expanded nodes for further expansion each time, we find that it has a higher probability of re-executing previously performed actions with identical prompts, leading to redundant information. In contrast, AutoCaption can consider different paths, thereby expanding the search for different details of the video. Furthermore, we find that as the number of Beam Search layers increases, the proportion of repeated content generated by the generator increases, further limiting the acquisition of information.

AutoCaption for Fine-tuning MLLMs We use AutoCaption to construct additional training data based on open-source video data. Before synthesizing with AutoCaption, the videos were filtered according to the method described in Appendix A.1, resulting in a final set of 9,419 videos.

For each video, key points were generated with AutoCaption. Next, GPT-4 was used to integrate these key points into a detailed thought process, simulating the human process of continuously exploring new details. Subsequently, GPT-4o was used to generate new captions based on the video (uniformly sampled at 32 frames) and the thought process, which were served as fine-tuning data. Detailed construction processes and data examples can be found in Appendix C.

Based on the evaluation results, we selected InternVL2_5-8B, which had average performance on MCTS-VCB, as the MLLM for fine-tuning. The model was fine-tuned for one epoch. In the experiments, we not only evaluate the results of fine-tuning using data generated by AutoCaption but also compare these results with those obtained by

Metric	Vanilla	+ GPT-4o caption	+AutoCaption (Ours)
MCTS-VCB Precision	69.6	77.6	78.0
MCTS-VCB Recall	40.1	52.0	53.6
MCTS-VCB F1	50.8	62.3	63.5
DREAM-1K Precision	34.4	33.7	36.8
DREAM-1K Recall	24.7	29.0	30.7
DREAM-1K F1	28.8	31.2	33.5
Video-MME	57.8	58.5	59.9
MMBench-Video	1.37	1.44	1.59

Table 4: Performance improvement of fine-tuned model on out-of-domain and in-domain benchmarks.

directly fine-tuning on responses generated by GPT-4o. This comparison is made to determine the effectiveness of our method, considering that GPT-4o is used in the process of generating fine-tuning data.

Finally, we take MCTS-VCB as an in-domain benchmark and DREAM-1K as the out-of-domain benchmark. Experiment results are shown in Table 4. The performance improvement of different dimensions on these two benchmarks is shown in Figure 1. Specific data can be found in Appendix D. We observe that the training data constructed by our method achieves the highest performance on both benchmarks. According to the official leaderboard¹ of DREAM-1K, the fine-tuning results by our methods (*i.e.*, using MCTS and Qwen2-VL-7B as generator) for constructing training data surpassed Qwen2-VL-72B and closely approached GPT-4o-mini. On the MCTS-VCB leaderboard, it ranks second among the open-source MLLMs. Additionally, the fine-tuning results based on data generated by AutoCaption also exceed the results of fine-tuning directly using GPT-4o’s responses, validating the effectiveness of AutoCaption.

7 Conclusion

We have presented AutoCaption, an automated framework leveraging MCTS to iteratively construct plenty of diverse and descriptive sentences. Additionally, we have constructed a video caption benchmark, MCTS-VCB, with AutoCaption. We evaluate 21 open- and closed-source MLLMs and conduct a comprehensive analysis over various dimensions. Furthermore, we reveal that the MLLM fine-tuned by AutoCaption-generated data achieves a great performance improvement on both in-domain and out-of-domain benchmarks, indicating the effectiveness of AutoCaption.

¹<https://tarsier-vlm.github.io/>

Acknowledgments

The present research was supported by the National Key Research and Development Program of China (Grant No. 2024YFE0203000). We would like to thank the anonymous reviewers for their insightful comments.

Limitations

While AutoCaption constructs a large number of key points, there are inevitably similar key points. Although our analysis shows that different similarity thresholds do not affect the evaluation ranking, it is still necessary to find an appropriate threshold that can balance the resource consumption and accuracy of the evaluation.

Ethics Policy

During the process of filtering video clips, we conducted a rigorous review to ensure the safety of the video content and its alignment with human values.

References

- Meysam Alizadeh, Maël Kubli, Zeynab Samei, Shirin Dehghani, Juan Diego Bermeo, Maria Kobeynikova, and Fabrizio Gilardi. 2023. Open-source large language models outperform crowd workers and approach chatgpt in text-annotation tasks. *arXiv preprint abs:2307.02179*.
- Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.
- David M. Chan, Suzanne Petryk, Joseph E. Gonzalez, Trevor Darrell, and John F. Canny. 2023. **CLAIR: evaluating image captions with large language models**. *CoRR*, abs/2310.12971.
- Hyeong Soo Chang, Michael C Fu, Jiaqiao Hu, and Steven I Marcus. 2005. An adaptive sampling algorithm for solving markov decision processes. *Operations Research*, 53(1):126–139.
- David L. Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *The Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024a. Alphamath almost zero: process supervision without process. *arXiv preprint arXiv:2405.03553*.

- Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. 2023. DISCO: distilling counterfactuals with large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024c. Self-play fine-tuning converts weak language models to strong language models. In *The International Conference on Machine Learning, (ICML)*.
- Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. Chataug: Leveraging chatgpt for text data augmentation. *arXiv preprint abs:2302.13007*.
- Lisa Dunlap, Alyssa Umno, Han Zhang, Jiezhi Yang, Joseph E. Gonzalez, and Trevor Darrell. 2023. Diversify your vision datasets with automatic diffusion-based augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Spandana Gella, Mike Lewis, and Marcus Rohrbach. 2018. A dataset for telling the stories of social media videos. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.
- Wenyi Hong, Yean Cheng, Zhuoyi Yang, Weihang Wang, Lefan Wang, Xiaotao Gu, Shiyu Huang, Yuxiao Dong, and Jie Tang. 2025. Motionbench: Benchmarking and improving fine-grained video motion understanding for vision language models. *arXiv preprint arXiv:2501.02955*.
- Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. 2024a. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yanwei Li, Chengyao Wang, and Jiaya Jia. 2024b. Llama-vid: An image is worth 2 tokens in large language models.
- Zhigen Li, Jianxiang Peng, Yanmeng Wang, Yong Cao, Tianhao Shen, Minghui Zhang, Linxi Su, Shang Wu, Yihang Wu, Yuqian Wang, Ye Wang, Wei Hu, Jianfeng Li, Shaojun Wang, Jing Xiao, and Deyi Xiong. 2025. Chatsop: An sop-guided mcts planning framework for controllable llm dialogue agents. *Preprint, arXiv:2407.03884*.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2024. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision (ECCV)*.
- Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, et al. 2024. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*.
- OpenAI. 2024. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>.
- Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lina Zhang, Fan Yang, and Mao Yang. 2024. Mutual reasoning makes smaller llms stronger problem-solvers. *arXiv preprint arXiv:2408.06195*.
- Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. 2013. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36.
- Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. 2015. A dataset for movie description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. 2017. Movie description. *International Journal of Computer Vision (IJCV)*, 123:94–120.
- Christopher D Rosin. 2011. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3):203–230.
- Ling Shi and Deyi Xiong. 2024. Crisieval: A chinese multi-level risk evaluation benchmark dataset for large language models. *Preprint, arXiv:2406.04752*.
- Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. 2024a. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Enxin Song, Wenhao Chai, Tian Ye, Jenq-Neng Hwang, Xi Li, and Gaoang Wang. 2024b. Moviechat+: Question-aware sparse memory for long video question answering. *arXiv preprint arXiv:2404.17176*.
- Zongheng Tang, Yue Liao, Si Liu, Guanbin Li, Xiaojie Jin, Hongxu Jiang, Qian Yu, and Dong Xu. 2021. Human-centric spatio-temporal video grounding with visual transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12):8238–8249.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- OpenAI Team. 2024a. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Qwen Team. 2024b. [Qwen2.5: A party of foundation models](#).
- Qwen Team. 2025. [Qwen2.5-vl](#).
- Petter Törnberg. 2023. Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. *arXiv preprint abs:2304.06588*.
- Jiawei Wang, Liping Yuan, and Yuchen Zhang. 2024a. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint abs:2407.00634*.
- Ke Wang, Jiahui Zhu, Minjie Ren, Zeming Liu, Shiwei Li, Zongye Zhang, Chenkai Zhang, Xiaoyu Wu, Qiqi Zhan, Qingjie Liu, et al. 2024b. A survey on data synthesis and augmentation for large language models. *arXiv preprint arXiv:2410.12896*.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024c. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024d. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint abs:2409.12191*.
- Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuanfang Wang, and William Yang Wang. 2019. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *IEEE/CVF International Conference on Computer Vision ICCV*.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Huajian Xin, Daya Guo, Zhihong Shao, Zhizhou Ren, Qihao Zhu, Bo Liu, Chong Ruan, Wenda Li, and Xiaodan Liang. 2024. Deepseek-prover: Advancing theorem proving in llms through large-scale synthetic data. *arXiv preprint abs:2405.14333*.
- Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The International Conference on Learning Representations (ICLR)*.
- Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems (NeurIPS)*.
- Linhao Yu, Yongqi Leng, Yufei Huang, Shang Wu, Haixin Liu, Xinmeng Ji, Jiahui Zhao, Jinwang Song, Tingting Cui, Xiaoqing Cheng, Liutao Liutao, and Deyi Xiong. 2024. [Cmoraleval: A moral evaluation benchmark for chinese large language models](#). In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 11817–11837. Association for Computational Linguistics.
- Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. 2019. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *The AAAI Conference on Artificial Intelligence (AAAI)*.
- Di Zhang, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. 2024a. Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b. *arXiv preprint arXiv:2406.07394*.
- Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, et al. 2024b. Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning. *arXiv preprint arXiv:2410.02884*.

- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. [Bertscore: Evaluating text generation with BERT](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. 2020b. Where does it exist: Spatio-temporal video grounding for multi-form sentences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Luowei Zhou, Chenliang Xu, and Jason Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
- Yue Zhou, Chenlu Guo, Xu Wang, Yi Chang, and Yuan Wu. 2024. A survey on data augmentation in large model era. *arXiv preprint arXiv:2401.15422*.
- Yiming Zhu, Peixian Zhang, Ehsan ul Haq, Pan Hui, and Gareth Tyson. 2023. Can chatgpt reproduce human-generated labels? A study of social computing tasks. *arXiv preprint abs:2304.10145*.

A AutoCaption

A.1 Video Screening Details

Some video clips, like those lacking a clear subject, featuring prolonged static scenes, or being overly complex and chaotic, are not suitable for captioning tasks. We employ both visual and textual information for filtering purposes. Specifically, we first use GPT-4o to describe all videos in detail, with the prompt “Please describe the video in detail.”. Subsequently, GPT-4o evaluates each video clip along with its corresponding caption to determine if it is appropriate for captioning, using the prompt as referenced in Figure 12.

A.2 Prompts Utilized for Different Action Types

During the expansion phase, we define six actions, ensuring the extraction of video content details both in terms of breadth and depth. For each action, there is a corresponding prompt to guide the process. The six actions and their respective prompts are described in Table 5.

A.3 MCTS Evaluation Phase Details

In the evaluation phase of MCTS iteration, we need to calculate the MC value for each newly expanded node. The MC value of each node s depends on the key points contained within node s . Therefore, we first need to extract the key points from node s . We use GPT-4 to extract key points from the behavior results represented by the node, and the prompt used for extracting key points can be found in Figure 13.

Next, we need to ask questions about the information of each extracted key point to facilitate subsequent verification work. As shown in Figure 3 in Section 4.1, each key point will generate n ($n \geq 1$) yes/no questions. For each yes/no question, we use two powerful MLLM as verifiers to answer each question. A key point is only considered correct if all its questions pass verification in both verifiers. We use GPT-4 to generate questions, and the corresponding prompt can be found in Figure 16. During verification, both verifiers use the prompt displayed in Figure 17.

The evaluation phase is the most critical step in MCTS because the evaluation of each node influences subsequent backpropagation and the selection in the next MCTS iteration. More specifically, the verification of each key point affects the MC value, which in turn impacts the evaluation results

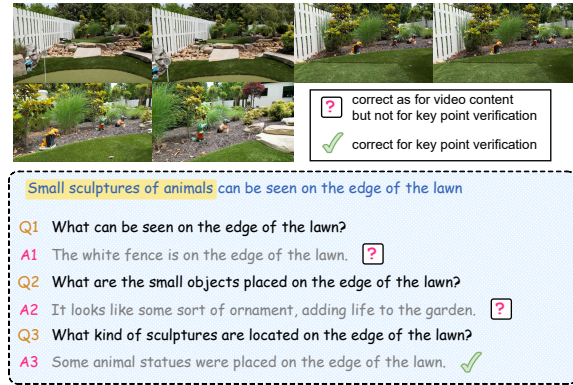


Figure 5: Illustration of “Key-Info as Answer” verification method.

of each node. Below, we introduce other methods we have attempted to verify key points and the reasons why they were ultimately not adopted.

- 1. Paraphrase + Multi-Verify:** We initially create three paraphrased versions of each key point extracted from the video and employed a Verifier to ascertain if each paraphrased point is indeed present in the video. The Verifier we used is Qwen2-VL-72B. However, during our experiments, it becomes evident that the Verifier has issues with misclassification. For instance, it mistakenly recognizes the number 5 on an athlete’s jersey as 6 in all three paraphrased versions of the key point, resulting in incorrect information being included in the verified key points. Furthermore, we note that using paraphrasing as a means of multiple verification was not effective, since the Verifier often provide consistent assessments across the paraphrases of the same key point.
- 2. Key-Info as Answer:** In this approach, we first use GPT-4 to identify key information and then formulate three questions with the key information as the answer. We then use GPT-4o to answer these three questions, considering the key point correct if at least two out of the three answers matched the key information. However, most of the generated questions are open-ended, leading to non-unique answers for the same video. Figure 5 illustrates the drawbacks of this method. As shown in Figure 5, it can be found that the answers to the first two questions are consistent with the video content, but they cannot be used to judge correctness of the key point. Because the key point generated using MCTS are so

Action Type	Prompt to Take Corresponding Action
Overall Description	Please describe the video in detail. Focus on providing a comprehensive overview of the entire video content, including the main subjects, actions, and settings.
Detail Description	<p>Stage 1 Next, I think you should carefully observe the details of the page, I think there are these details to consider, People or Animals, Plants, Food, Buildings, Vehicles, Other Objects and so on. Please output only one detail you want to focus on. [<i>Optional</i> I want you to focus on the detail in the video and describe it in the following detail: relevant_detail_attributes]</p> <p>Stage 2 For MLLMs, generating video content description is a very important task, which is called “video caption” task. At the same time, it is hoped that the model can pay more attention to the details of the video when describing the video. \n To do that, I’ll start by giving the model the video to determine the details that need attention. \n Then I will give you the “MODEL ANSWER” (what the model think itself should focus on) and need you to extract the following information from this “MODEL ANSWER”: \n 1. What video detail does the model need to focus on? \n 2. What category does this detail fall into? \n 3. What describe aspects can model focus on for this detail? \n Here are some categories of details and describe aspects to focus on under the category, if the details extracted from the answer can not be classified into the following categories, you can think of your own: \n 1. People or Animals: Describe their expressions, facial features, postures, clothing, age, and quantity. Please include any notable actions or interactions they are involved in. \n 2. Plants: Describe the quantity, types, size, color, and any notable features such as flowers or fruits. \n 3. Food: Describe the quantity, types, colors, and presentation (e.g., plated, packaged). \n 4. Buildings: Describe the quantity, types, architectural style, appearance, shapes, and any distinctive features (e.g., windows, doors, decorations). \n 5. Vehicles: Describe the types, appearance, quantity, color, and any notable features (e.g., brand, model, condition). \n Please answer in this format: \n Detail: [the detail what the model think itself should focus on in MODEL ANSWER.] \n Category: [What category does this detail fall into. Can either pick from the above categories or think by yourself.] \n Relevant Describe Aspects: [The possible describe aspects when focusing on the detail.] \n Here is the MODEL ANSWER: model_answer \n Output: \n</p>
New Temporal Perspective Description	Please describe the video from a new temporal perspective. Focus on changes that occur before and after a specific camera transition or time point in the video. \n \n Here are the previously mentioned perspectives for reference: \n {previous perspectives}
New Spatial Perspective Description	Please describe the video from a new spatial perspective. Focus on different areas of the frame, such as the left side, right side, foreground, or background. \n \n Here are the previously mentioned perspectives for reference: \n {previous perspectives}
Background Description	Please provide a detailed description of the background in the video. Focus on the setting, environment, and any contextual elements that contribute to the overall scene. \n \n Here are the previously mentioned perspectives for reference: \n {previous descriptions}
Camera Move Description	Please describe the changes in camera shots and movements throughout the video. Focus on different types of shots, camera movements, angles, transitions, and any special effects used. \n \n Here are the previously mentioned perspectives for reference: \n {previous descriptions}

Table 5: The action prompt used in “expand” phase of MCTS.

detailed and subtle, the verifier cannot answer the information we want in such a open-ended question.

Ultimately, our method improves upon Method 2 by modifying the questions to a yes/no format to ensure clear directionality. To enhance the accuracy of the verification, we select two powerful MLLMs (GPT-4o and Qwen2-VL-72B) based on the evaluation results from OpenCompass² by November.

²<https://rank.opencompass.org.cn/home>. An open-source, efficient, and comprehensive large model evaluation system and open platform

A.4 Post Processing Details

During the evaluation phase of MCTS, we have already determined the correctness of the key points contained in each generated description. In the filtering stage, we need to perform two steps: 1. Filter out the correct key points in each node, 2. Use GPT-4 to filter out overly subjective and broad key points. These key points may be similar to the data distribution of the generator’s output (Qwen2-VL-7B), leading to distorted evaluation results. Next, for the filtered key points, we remove duplicates, retaining a similarity of 0.9 to increase the diversity of key points by using “all-MiniLM-L6-v2

”³. To further address the issue of key point data distribution, we used GPT-4 to paraphrase all key points. However, this operation had almost no impact on the evaluation results, possibly because the data distribution correction was already performed when using GPT-4 to extract key points from the nodes. The prompt we used during filtering phase and paraphrasing phase can be found in Figure 18 and Figure 15 respectively. As a benchmark, it is essential to ensure that the key points generated by AutoCaption are entirely reliable. Therefore, we let annotators check the key points generated under a similarity threshold of 0.9, retaining only those that accurately reflect the video content, with a human verification pass rate of 94.7%.

B MCTS-VCB

B.1 MCTS-VCB Statistics

After the refined process of AutoCaption, we obtained an accurate and comprehensive key point data pool for each video, MCTS-VCB. MCTS-VCB comprises 1,765 videos, with an average length of 12.39 seconds per video and an average of 122.3 keypoints per video. The video categories are diverse, encompassing ten categories: 1) Movies and TV Shows, 2) Education, 3) Health and Sports, 4) Fashion and Beauty, 5) Home and Living, 6) Travel and Adventure, 7) Nature and Wildlife, 8) Technology and Science, 9) Arts and Creativity, and 10) Food and Drink. As shown in Figure 6, the *Nature and Wildlife* category has the highest number of videos, reaching 382, while the Arts and Creativity category has the fewest, with 57 videos. Additionally, the average number of key points contained in videos of different categories does not vary significantly, which demonstrates the applicability of AutoCaption across various types of videos.

According to Figure 7, the highest number of videos falls within the 10–12s range, reaching 382. There is a reasonable distribution of videos across different duration ranges. With 224 videos exceeding 24s, this allows for a comprehensive evaluation of MLLMs’ performance across various video lengths.

In AutoCaption, it is verified to obtain detailed, comprehensive, and nuanced key points of a video. However, when merging key points from different nodes s in search tree T , semantically similar key

³<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

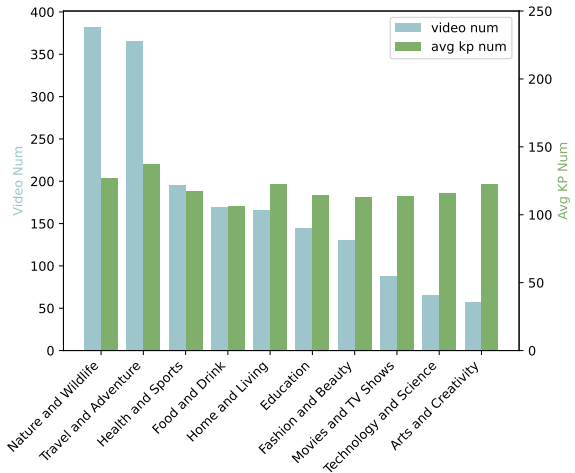


Figure 6: Video counts and average key point counts across different video categories.

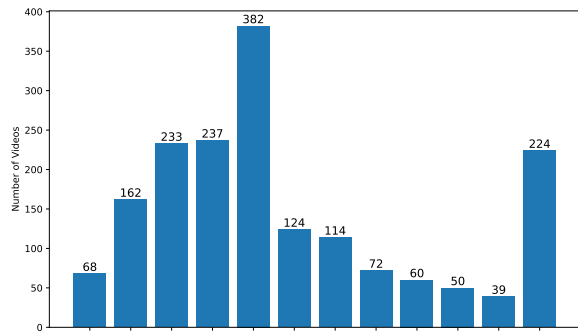


Figure 7: Video length distribution in seconds.

points may exist, and the subtle differences contained within these semantically similar key points could be a weakness of MLLMs. Therefore, when forming MCTS-VCB, we also analyzed data with different similarity thresholds. We evaluated ten thresholds ranging from 0.7 to 0.9, and as shown in Figure 8, the average number of key points per video gradually increases with higher similarity thresholds. We also discuss the impact of different similarity thresholds on evaluation results in the experimental section 6.1.

We categorized the key points into five types: 1) Appearance Description, 2) Action Description, 3) Environment Description, 5) Object Description, and 6) Camera Movement and Composition. The definition and examples of each category is described in Figure 10.

Additionally, Figure 9 shows the distribution of different keypoint categories in an average video. It can be observed that *Environment Description* keypoint type is the most prevalent among the key points. However, there is currently no dataset that

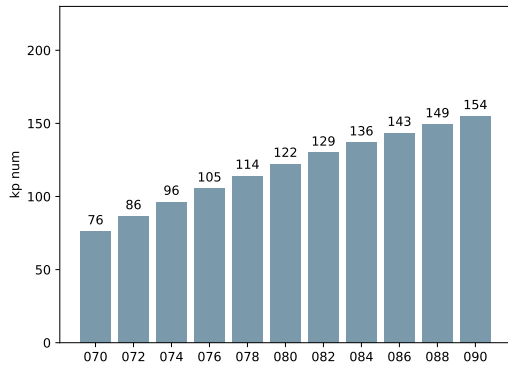


Figure 8: Video counts across different deduplicate similarity threshold.

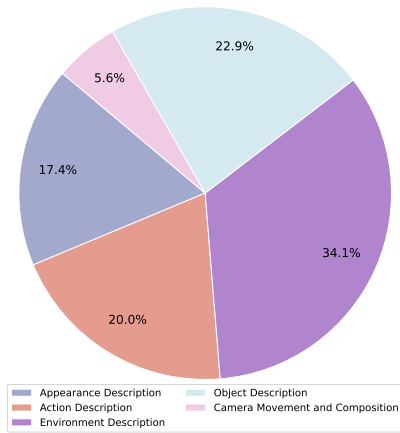


Figure 9: The distribution of key points of a video.

focuses on environment descriptions when evaluating video captioning tasks, which is an advantage of MCTS-VCB. The second most common key point type is *Object Description*, indicating that AutoCaption effectively generates key points related to object details, surpassing other key point types such as *Action Description* that are the focus of other works.

B.2 Evaluation Experiments Setting

We evaluate 21 open- and close-source MLLMs, with varying sizes. The model cards are shown in Table 6. We used different prompts for different series of MLLMs, which can be categorized into the following three classes:

1. For VILA series, we use the prompt displayed

on GitHub ⁴: “Elaborate on the visual and narrative elements of the video in detail.”

2. For Pllava series, we use the prompt shown on official script ⁵: “You are to assist me in accomplishing a task about the input video. Reply to me with a precise yet detailed response. For how you would succeed in the recaptioning task, read the following Instructions section and Then, make your response with a elaborate paragraph.\n # Instructions\n 1. Avoid providing over detailed information such as color, counts of any objects as you are terrible regarding observing these details\n 2. Instead, you should carefully go over the provided video and reason about key information about the overall video\n 3. If you are not sure about something, do not include it in you response.\n # Task\n Describe the background, characters and the actions in the provided video.\n”
3. For all other MLLMs, we use “Please describe the video in detail.”

B.3 Supplementary Experiment Results

As shown in Figure 11, it is interesting to note that as the similarity threshold increases, the evaluation scores of the models also increase. This phenomenon is reasonable because the final key points, after deduplication using higher similarity thresholds, contain more similar key points. These key points are more easily generated by the generator during MCTS generation. Upon observation, we found that these key points and the similar parts of the descriptions for the same video by different models are generally consistent. This consistency leads to better performance in the evaluation sets filtered by higher similarity thresholds. To ensure that MCTS-VCB includes key points with a certain degree of semantic similarity (since semantic similarity does not imply complete consistency in detailed information), all experimental results in this paper are based on a similarity threshold of 0.8.

⁴<https://github.com/NVlabs/VILA?tab=readme-ov-file#video-captioning>

⁵https://github.com/magic-research/PLLaVA/blob/main/tasks/eval/recaption/pllava_recaption.py#L121C1-L129C6

Example Video Frames



KP Category Explanation

KP Category	Defination	Examples
Appearance Description	Appearance description refers to the physical characteristics, attire, or visible features of the person or object in the video.	1. A young boy is wearing a white shirt. 2. This girl possesses curly hair.
Action Description	Action description focuses on the specific behavior or activity that takes place in the video. Task states associated with actions can also fall into this category.	1. The young boy returns the flask to the table. 2. The actions of the young boy are careful and well-measured.
Environment Description	Environment description covers the background and environmental features of the scene in the video.	1. The background is kept tidy and orderly. 2. The room fosters an educational and orderly atmosphere.
Object Description	Describes the features, characteristics, or details of inanimate objects or items present in the scene.	1. The artwork includes various colors like blue, red, and yellow. 2. The glassware items include beakers, test tubes, and flasks.
Camera Movement	Describes the camera angles, movements, framing, or other cinematographic techniques used in the scene.	1. The camera is fixed, providing a consistent and clear view of the activity.

Figure 10: Illustration of key points categories.

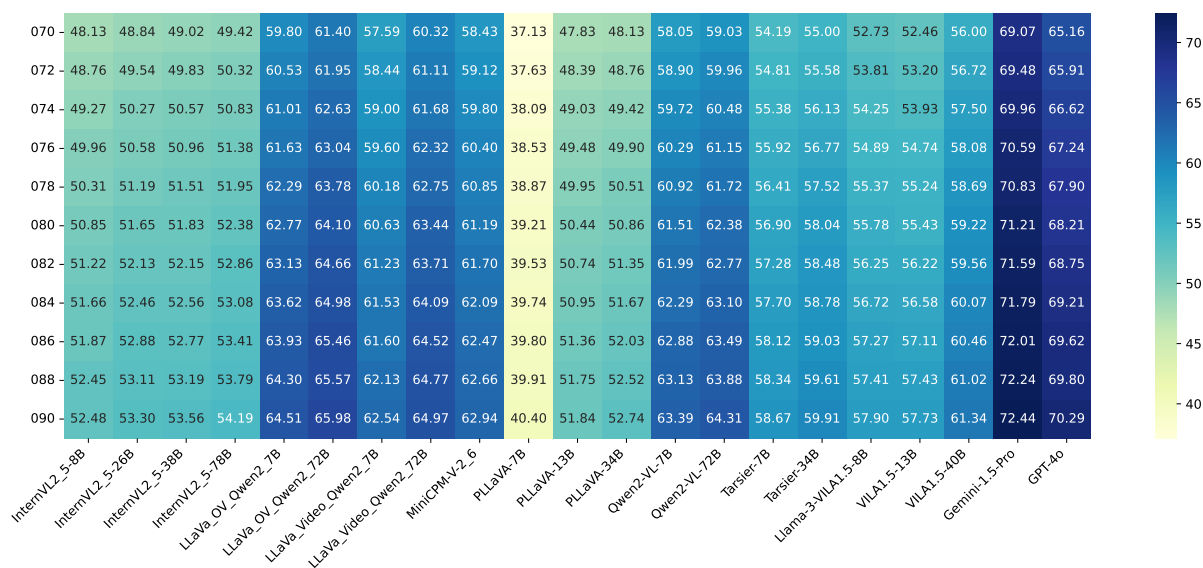


Figure 11: F1 score across different similarity threshold.

C Training Data Curation Detail & Example

The construction of the training data followed AutoCaption but omitted the human check in data post-processing and the “Overall Description” action is also executed by the generator. First, key points were obtained from a video within AutoCaption. Then, we used the prompt shown in Figure 19 to guide GPT-4 in merging the key points into a

thought process according to the generated order. As illustrated in Figure 20, we placed the thought process within `<thought>...</thought>` to guide GPT-4o in generating detail-focused captions based on the video and the thought process. The generated captions will be used as fine-tuning data.

D Complete Experimental Results

The specific result of the fine-tuned MLLM's performance on different dimensions of MCTS-VCB and DREAM-1K are displayed in Table 7.

The complete result of precision, recall and F1 score of MLLMs' performance across different video categories is displayed in Table 8.

The complete result of precision, recall and F1 score of MLLMs' performance across different similarity thresholds used in merging key points from all nodes s of one search tree T is displayed in Table 9.

Model Name	Language Model	Vision Model	#Frames
Open-Source MLLMs			
InternVL2_5-8B	Internlm2_5-7b-chat	InternViT-300M-448px-V2_5	16
InternVL2_5-26B	Internlm2_5-20b-chat	InternViT-6B-448px-V2_5	16
InternVL2_5-38B	Qwen2.5-32B	InternViT-6B-448px-V2_5	16
InternVL2_5-78B	Qwen2.5-72B	InternViT-6B-448px-V2_5	16
LLaVa_OV_Qwen2_7B	Qwen2-7B	SigLIP-SO400M	16
LLaVa_OV_Qwen2_72B	Qwen2-72B	SigLIP-SO400M	16
LLaVa_Video_Qwen2_7B	Qwen2-7B	SigLIP-SO400M	16
LLaVa_Video_Qwen2_72B	Qwen2-72B	SigLIP-SO400M	16
MiniCPM-V-2_6	Qwen2-7B	SigLIP-SO400M	16
PLLaVA-7B	Vicuna-7B-v1.5	CLIP ViT	16
PLLaVA-13B	Vicuna-13B-v1.5	CLIP ViT	16
PLLaVA-34B	Nous-Hermes-2-Yi-34B	CLIP ViT	16
Qwen2-VL-7B-Instruct	Qwen2-7B	ViT-675M	16
Qwen2-VL-72B-Instruct	Qwen2-72B	ViT-675M	16
Tarsier-7B	Vicuna-7B-v1.5	CLIP ViT	16
Tarsier-34B	Nous-Hermes-2-Yi-34B	CLIP ViT	16
Llama-3-VILA1.5-8B	Llama3-8B	SigLIP-SO400M	16
VILA1.5-13B	Vicuna-7B-v1.5	SigLIP-SO400M	16
VILA1.5-40B	vicuna1.5	SigLIP-SO400M	15
Closed-Source MLLMs			
Gemini-1.5-Pro	/	/	16
GPT-4o	/	/	16

Table 6: The model cards of 21 evaluated MLLMs, including both open- and close-source ones. For most MLLMs, we sample 16 frames uniformly to represent videos; however, for VILA1.5-40B, due to context length limits, we sample 15 frames uniformly as input.

Please evaluate the given video according to the given video content and corresponding caption to determine if it meets the following criteria:

1. The video must have a clear subject (e.g., person, animal, object, etc.).
2. The video should not contain a lot of special effects.
3. The video cannot contain long still clips.
4. The video must not be confused, unclear meaning.

If the video meets all the criteria, output [yes]; if it fails to meet any of the criteria, output [no].
The input format is: “Judgment: [yes/no] Reason: [Brief explanation]”

Please provide your judgment and briefly explain the reason.

Figure 12: The prompt used for GPT-4o to filter video clips.

MCTS-VCB	Appearance Description	Action Description	Environment Description	Object Description	Camera Movement and Composition
Vanilla	65.0/42.0/51.0	76.0/48.7/59.4	72.1/30.3/42.7	54.7/31.7/40.1	69.6/40.1/50.8
Fine-tuned	74.2/48.6/58.7	78.6/59.9/68.0	81.0/62.5/70.5	74.1/43.9/55.1	72.0/32.3/44.6
DREAM-1K	Movie Animation	Movie Live Action	Shorts	Stock	Youtube
Vanilla	28.1/16.1/20.5	30.1/25.0/27.3	39.5/24.2/30.0	37.8/31.6/34.4	36.4/26.7/30.8
Fine-tuned	31.1/25.2/27.8	33.9/31.6/32.7	40.7/28.1/33.2	38.6/37.7/38.2	39.5/31.1/34.8

Table 7: The performance improvement across different dimensions of MCTS-VCB and DREAM-1K. Each cell contains “**Precision / Recall / F1 Score**”.

Please break down the following video caption into individual key points. A sentence in the video caption should be split to individual key points if it contains much information.

Requirement:

1. Make sure to substitute pronouns in split individual sentences by the nouns they refer to.
2. If a individual sentence contains uncertain expressions, these expressions should not be included as key points.

For clarity, consider these examples:

Example 1

Video Description:

The video showcases ...

Result:

> {Atomic key point}

> {Atomic key point}

> ...

{two more examples}

With these examples in mind, please help me break down the following video caption into individual key points.

Video Description:

description

Result:

Figure 13: The prompt used for Qwen2-VL-72B to extract key points

Algorithm 1 AutoCaption Workflow

- 1: **Input:** 64 video frames uniformly sampled from video v .
 - 2: **Output:** Final video key points
 - 3: Initialize search tree T with root node containing video v
 - 4: **for** iteration $i = 1$ to 25 **do**
 - 5: **Selection:** Select node s_i with highest value using PUCT
 - 6: **Expansion:** Expand node s_i with possible actions
 - 7: **Evaluation:** Calculate state value $Q(s, a)$ for new nodes
 - 8: **Backpropagation:** Update $N(s)$ and $Q(s, a)$ for nodes in $t(s_i)$
 - 9: **end for**
 - 10: **Data Post-processing:**
 - 11: **Filter:** Remove irrelevant descriptions.
 - 12: **Deduplicate:** Eliminate duplicate descriptions.
 - 13: **Paraphrase:** Rephrase descriptions to alleviate data distribution of the generator.
 - 14: **Human Check:** Double check final key points.
 - 15: **Return** Final key points of the video
-

Please objectively classify the relationship between each video caption breakdown and provided human-generated key points.

Analyze each breakdown point individually to determine its relationship with the human-generated key points.

For each breakdown point, classify the relationship into one of the following categories:

1. “entailment ” means that the breakdown point is accurately reflected within one or more of the human-generated key points.
2. “contradiction ” means that breakdown point some detail in the breakdown point contradicts with the infomation mentioned human-generated key points.
3. “neutral ” means that the relationship is neither “entailment ” nor “contradiction ”.

For each breakdown point, provide a brief analysis explaining the reasoning behind your judgment.

Please present the result in a JSON dict format: {“ breakdown_point_1”: {“ judgement”: judgement_1, “ analysis”: analysis_1}, ..., “ breakdown_point_n”: {“ judgement”: judgement_n, “ analysis”: analysis_n}}.

For clarity, consider these examples:

{two examples }

With these examples in mind, please help me evaluate whether each breakdown point is accurately reflected in the provided human-generated key points.

Human-Generated Video Key Points:

key_point

LMM Caption Breakdown Point to Evaluate:

caption

Result:

Figure 14: The prompt used for Qwen2.5-72B-Instruct to determine the relationship between KP_{model} and KP_{ref} . When to determine the relationship between KP_{ref} and D_{mdoel} , we provide D_{mdoel} . instead of “LMM Caption Breakdown Point to Evaluate ” in this prompt.

You are an experienced linguist, please help me rewrite the sentence num sentences, requiring the semantics to remain unchanged, and not adding or removing any details.
All rewritten sentences should be put in a list. Please only output the list!

Sentences:

["1. sentence1", "2. sentence2", "3. sentence3"]

Output:

["1. rewritten sentence1", "2. rewritten sentence2", "3. rewritten sentence3"]

Sentences:

sentences

Output:

Figure 15: The prompt used for GPT-4 to paraphrase key points.

You are given a key point extracted from a video description.
Your task is to identify the key information within the given key point and construct yes/no questions according to the identified key information.
If the key point has more than one key information, you need to generate more than one question.

For clarity, consider these examples:

Example 1

Key Point

key point 1

Output

> question 1

> ...

{two more examples}

With these examples in mind, please help me break down the following video caption into individual key points.

Key Point

key point

Output

Figure 16: The prompt used for Qwen2-VL-72B to generate verification questions of a key point.

You are provided with several frames extracted from a video and ques_num yes/no questions related to the content of the video.
Your task is to analysis the frames carefully and provide a clear and concise judgement (yes/no) to each question based on the video frames.
For, each question, please provide a single sentence answer (format like “[Judgement], [Reason]”) to the question based on the video frames’ content.
All answers should be put in a list.
Please only output the list!

Example 1

Questions

{1. question 1}

{2. question 2}

Output

[“1. [Judgement 1], [Reason 1]”, “2. [Judgement 2], [Reason 2]”]

{one more example}

Questions

{question list}

Output

Figure 17: The prompt used for both GPT-4o and Qwen2-VL-72B to answer verification questions of a key point.

You are given a list of key points describing a video.
Your task is to filter out key points that are too subjective, too minor, too general, or express speculation.

Additionally, any key points related to the historical background or cultural context of the video should also be filtered out, as they are not directly relevant to the video content.

For each key point, output the judgement “yes” if it should be kept, or “no” if it should be filtered out. Also output a reason for your judgement.

Criteria for filtering:

1. Subjective: Key points that rely heavily on personal feelings or interpretations.
2. Minor: Key points that are overly detailed and do not add significant value to the overall description.
3. General: Key points that are too broad and do not provide specific information about the video.
4. Speculative: Key points that express guesses or assumptions rather than concrete information.
5. Historical/Cultural: Key points that relate to the historical background or cultural context of the video.

For clarity, consider these examples:

Example 1

Key Points

{1. These people could be tourists or travelers visiting the site.}

{...}

Output

1. “[No] Speculative.”

...

{two more examples}

With these examples in mind, please help me filter out key points that are too subjective, too minor, too general, or express speculation.

Key Points

{key points}

Output

Figure 18: The prompt used for GPT-4 to filter too subjective and broad key points.

You are tasked with generating a structured internal thought process that explains how you analyze and describe a video. The goal is to demonstrate a step-by-step reasoning process, gradually refining the observations based on provided key points.

Formatting Requirements:

1. Your thought process should be enclosed within `<thought> . . . </thought>` tags.
2. Your thought should contain the provided “Overall Description” and all (Observation,Key Point) pairs.
3. The reasoning should be sequential and structured, mimicking a natural process of observation and refinement.

Example Input and Output:

Input:

Overall Description: {overall_description}

Observation: Vehicles

Key Point: There is no existence of any vehicles in the video.

Observation: Buildings

Key Point: None

Observation: Snow-covered mountains and a sea of clouds

Key Points:

1. The mountain peaks stand tall against a backdrop of a sea of clouds.
2. The mountains appear in various shapes and sizes.
3. The mountains create a dynamic and picturesque landscape.

...

{more Observation-Key Point pairs}

Expected Output:

`<thought>{thought process}</thought>`

Remember the above example and requirements. Now, please help me integrate the following 'Overall Description' and multiple 'Observation - Key Point' pairs into a complete internal thought process.

Input:

Overall Description:

{overall_description}

{observation_kp_pairs}

Expected Output:

Figure 19: The prompt used for GPT-4 to generate thought process according to video key points.

According to the reasoning thought, describe the video in detail.

<thought> As I begin to analyze the video, I first take in the overall description to set the context. The scene is festive and joyful, featuring a young child sitting on Santa's lap in front of a Christmas tree. Santa is reading from an old-fashioned typewriter, and behind them, elves are busy preparing presents. This setting captures the essence of the holiday spirit, emphasizing the joy of giving and receiving gifts.

Moving on to specific observations, I notice the child's interaction with the typewriter. This typewriter is not just any typewriter; it's vintage, with a mechanical design that includes visible gears and levers. The body of the typewriter shows signs of wear and age, suggesting it has been cherished over many years. The child, pressing the keys and focused on their task, adds a personal touch to the scene. The surrounding environment complements this interaction, with a table filled with wrapped gifts and papers, and decorative elements in the background enhancing the festive atmosphere.

Next, I expand my focus to include other key elements in the scene. Central to the image is Santa Claus, seated at a table adorned with neatly wrapped presents. Santa, in his traditional red and white costume with a lush white beard, creates a classic holiday image. Beside him, the child in a bright yellow outfit adds a pop of color. The elves, dressed in green uniforms with red hats, are a testament to the busy and cheerful environment. The table not only holds the typewriter but also various other items that contribute to the festive setting.

The background elements also play a significant role in enriching the scene. A beautifully decorated Christmas tree stands tall, its branches laden with twinkling lights and adorned with ornaments, creating a magical backdrop. Additionally, a large window to the side of the tree allows natural light to flood the room, enhancing the warmth and inviting nature of the setting.

By examining these details, from the child's interaction with the typewriter to the broader festive environment, I gain a comprehensive understanding of the video. Each element, whether it's the vintage typewriter, Santa's traditional attire, or the elves' busy preparations, contributes to the overall holiday spirit captured in this scene. The video beautifully portrays the magic of the holiday season through these meticulously arranged and detailed festive elements.

</thought>

Figure 20: An example of prompting GPT-4o to generate more detailed caption based on thought process.

Model Name	Movies and TV Shows	Education	Health and Sports	Fashion and Beauty	Home and Living	Travel and Adventure	Nature and Wildlife	Technology and Science	Arts and Creativity	Food and Drink
Open-Source MLLMs										
InternVL2_5-8B	69.8/39.7/50.6	71.0/40.9/51.9	66.9/42.3/51.8	68.6/39.7/50.3	69.7/39.1/50.1	68.7/40.5/50.9	71.5/47.2/56.9	69.8/42.7/53.0	73.2/44.8/55.6	67.7/40.9/51.0
InternVL2_5-26B	71.3/41.4/52.4	72.1/42.2/53.3	67.7/43.1/52.7	69.1/40.5/51.0	69.2/38.8/49.8	69.8/41.5/52.1	71.5/47.9/57.4	67.5/44.1/53.4	72.7/43.9/54.8	70.0/42.3/52.7
InternVL2_5-38B	70.5/40.1/51.2	72.1/42.8/53.7	69.5/43.7/53.6	70.5/40.1/51.1	70.8/41.0/51.9	70.5/41.1/51.9	71.7/47.1/56.9	68.8/44.7/54.2	69.5/44.6/54.3	68.9/43.7/53.5
InternVL2_5-78B	72.7/42.1/53.3	72.3/41.8/53.0	69.9/44.7/54.6	71.0/41.3/52.2	69.6/40.3/51.1	70.5/42.5/53.0	70.9/48.1/57.3	68.6/44.3/53.9	73.0/46.6/56.8	69.2/43.2/53.2
LLaVa_OV_Qwen2_7B	77.3/52.8/62.7	77.7/55.1/64.5	78.0/55.0/64.5	77.9/52.4/62.6	77.2/52.5/62.5	80.0/52.2/63.2	83.2/56.4/67.2	79.7/53.6/64.1	82.6/57.0/67.5	78.4/55.2/64.8
LLaVa_OV_Qwen2_72B	78.8/51.7/62.4	77.9/56.4/65.4	78.4/57.2/66.1	79.5/52.0/62.9	79.1/53.9/64.1	82.1/52.5/64.0	85.1/58.1/69.1	84.7/53.9/65.9	82.3/58.4/68.3	80.2/57.2/66.8
LLaVa_Video_Qwen2_7B	78.4/47.2/59.0	78.9/49.7/61.0	78.7/50.6/61.6	78.7/47.2/59.0	78.1/48.7/60.0	79.9/50.3/61.8	81.7/55.6/66.2	79.6/50.7/62.0	82.0/56.3/66.7	79.1/52.4/63.0
LLaVa_Video_Qwen2_72B	80.6/52.1/63.3	80.3/53.9/64.5	81.4/55.3/65.9	79.9/51.7/62.8	79.4/52.5/63.2	83.1/51.6/63.7	85.0/56.8/68.1	82.4/55.0/66.0	80.9/57.0/66.8	81.2/55.3/65.8
MiniCPM-V-2_6	74.9/50.8/60.5	76.4/49.4/60.0	78.1/53.0/63.2	76.0/49.6/60.1	75.6/49.2/59.6	79.3/52.3/63.0	79.2/56.7/66.1	78.0/51.2/61.8	80.3/56.2/66.1	75.9/51.0/61.0
PLLaVA-7B	58.4/33.6/42.7	50.8/26.8/35.1	59.7/33.6/43.0	57.6/33.7/42.5	53.2/28.4/37.0	58.7/32.9/42.1	54.5/36.2/43.5	43.9/22.8/30.0	49.7/28.3/36.1	50.3/26.9/35.1
PLLaVA-13B	73.9/39.7/51.2	71.5/38.9/50.4	70.1/39.6/50.6	70.8/38.5/49.9	71.7/38.1/49.8	73.0/39.3/51.1	75.1/45.9/56.9	73.3/39.8/51.6	75.7/44.4/56.0	68.3/37.2/48.2
PLLaVA-34B	72.0/39.7/51.2	75.3/38.5/51.0	74.9/41.0/53.0	71.5/38.7/50.3	74.7/38.3/50.7	74.5/38.6/50.8	78.0/45.4/57.4	76.2/41.2/53.5	78.3/44.4/56.0	71.3/37.8/49.4
Qwen2-VL-7B	76.0/49.8/60.1	75.6/52.5/62.0	76.2/54.0/63.2	75.3/52.0/61.5	76.8/51.3/61.5	78.7/51.8/62.5	81.1/56.9/66.9	75.2/54.0/62.8	79.4/56.7/66.2	75.8/54.9/63.7
Qwen2-VL-72B	75.9/51.7/61.5	76.2/54.4/63.5	73.8/55.6/63.5	74.1/52.4/61.4	76.5/52.9/62.5	80.0/53.0/63.7	81.4/58.3/67.9	75.7/56.0/64.3	78.9/55.4/65.1	74.7/55.2/63.5
Tarsier-7b	72.4/45.0/55.5	73.7/46.9/57.3	74.7/47.2/57.9	73.3/44.8/55.6	73.2/44.7/55.5	80.8/46.4/58.9	81.3/52.6/63.9	75.6/47.7/58.5	76.6/48.9/59.7	70.8/44.1/54.4
Tarsier-34b	74.1/47.6/58.0	75.7/45.6/56.9	78.7/47.7/59.4	76.2/44.7/56.3	77.9/47.0/58.6	84.0/46.8/60.1	83.2/52.6/64.4	77.8/46.8/58.4	82.7/50.0/62.3	73.5/45.7/56.4
Llama-3-VILA1.5-8b	75.2/42.9/54.7	73.0/44.3/55.2	73.5/47.4/57.7	74.7/42.7/54.3	73.2/43.3/54.4	76.4/45.0/56.7	77.2/51.7/61.9	76.8/46.4/57.9	81.2/50.3/62.1	75.5/47.5/58.3
VILA1.5-13b	69.5/43.6/53.6	70.7/45.5/55.3	72.1/47.8/57.5	72.3/44.5/55.1	72.5/43.7/54.5	75.8/45.0/56.4	76.6/51.9/61.9	72.4/45.3/55.8	75.9/50.4/60.6	72.9/45.6/56.1
VILA1.5-40b	78.1/48.9/60.1	76.6/49.1/59.9	77.6/52.5/62.6	76.5/48.5/59.4	75.5/47.6/58.4	80.3/47.7/59.9	79.0/53.7/63.9	75.2/48.2/58.8	84.9/53.2/65.4	77.3/50.6/61.2
Closed-Source MLLMs										
Gemini-1.5-Pro	78.6/66.2/71.9	80.1/66.6/72.7	80.4/65.6/72.2	77.5/65.9/71.2	81.4/64.2/71.8	82.1/63.4/71.5	81.8/67.0/73.7	79.7/65.6/72.0	79.7/67.0/72.8	81.0/66.2/72.9
GPT-4o	83.6/53.0/64.9	84.0/57.1/68.0	86.0/59.1/70.0	85.6/57.3/68.7	84.5/58.4/69.1	83.7/60.2/70.0	82.3/64.1/72.1	82.8/59.2/69.0	85.2/61.4/71.4	83.0/61.2/70.5

Table 8: Performance on MCTS-VCB across different video types. Each cell contains "Precision / Recall / F1 Score"

Model Name	0.70	0.72	0.74	0.76	0.78	0.80	0.82	0.84	0.86	0.88	0.90
Open-Source MLLMs											
InternVL2_5-8B	68.2/37.2/48.1	68.5/37.8/48.8	68.8/38.4/49.3	69.8/38.9/50.0	69.1/39.6/50.3	69.6/40.1/50.8	69.4/40.6/51.2	69.7/41.0/51.7	69.3/41.5/51.9	70.1/41.9/52.4	69.0/42.3/52.5
InternVL2_5-26B	68.7/37.9/48.8	69.1/38.6/49.5	70.1/39.2/50.3	69.5/39.7/50.6	69.9/40.4/51.2	70.1/40.9/51.7	70.4/41.4/52.1	70.3/41.8/52.5	70.6/42.3/52.9	70.2/42.7/53.1	69.7/43.2/53.3
InternVL2_5-38B	68.8/38.1/49.0	69.5/38.8/49.8	70.6/39.4/50.6	70.4/39.9/51.0	70.7/40.5/51.5	70.6/41.0/51.8	70.2/41.5/52.1	70.5/41.9/52.6	70.0/42.4/52.8	70.3/42.8/53.2	70.3/43.2/53.6
InternVL2_5-78B	68.9/38.5/49.4	69.8/39.4/50.3	70.0/39.9/50.8	70.2/40.5/51.4	70.5/41.1/51.9	70.6/41.6/52.4	70.9/42.1/52.9	70.4/42.6/53.1	70.3/43.1/53.4	70.5/43.5/53.8	70.7/43.9/54.2
LLaVa_OV_Qwen2_7B	77.6/48.6/59.8	78.1/49.4/60.5	78.3/50.0/61.0	78.8/50.6/61.6	79.4/51.3/62.3	79.6/51.8/62.8	79.5/52.4/63.1	79.9/52.8/63.6	79.8/53.3/63.9	79.9/53.8/64.3	79.6/54.2/64.5
LLaVa_OV_Qwen2_72B	80.0/49.8/61.4	80.1/50.5/61.9	80.8/51.1/62.6	80.7/51.7/63.0	81.5/52.4/63.8	81.3/52.9/64.1	81.8/53.4/64.7	81.7/54.0/65.0	82.1/54.4/65.5	81.4/54.9/65.6	81.7/55.3/66.0
LLaVa_Video_Qwen2_7B	77.9/45.7/57.6	78.8/46.4/58.4	79.2/47.0/59.0	79.4/47.7/59.6	79.6/48.4/60.2	79.7/48.9/60.6	80.3/49.5/61.2	80.0/50.0/61.5	79.0/50.5/61.6	79.6/50.9/62.1	79.8/51.4/62.5
LLaVa_Video_Qwen2_72B	80.0/48.4/60.3	80.7/49.2/61.1	80.9/49.8/61.7	81.5/50.5/62.3	81.2/51.1/62.7	82.1/51.7/63.4	81.7/52.2/63.7	81.7/52.7/64.1	82.0/53.2/64.5	81.7/53.7/64.8	81.3/54.1/65.0
MiniCPM-V-2_6	76.2/47.4/58.4	76.8/48.1/59.1	77.4/48.7/59.8	77.9/49.3/60.4	77.9/49.9/60.9	77.7/50.4/61.2	78.2/51.0/61.7	78.3/51.4/62.1	78.6/51.8/62.5	78.2/52.3/62.7	78.2/52.6/62.9
PLLaVA-7B	54.2/28.2/37.1	54.4/28.8/37.6	54.7/29.2/38.1	54.9/29.7/38.5	54.8/30.1/38.9	55.1/30.4/39.2	55.3/30.8/39.5	55.0/31.1/39.7	54.4/31.4/39.8	54.0/31.7/39.9	55.0/31.9/40.4
PLLaVA-13B	71.2/36.0/47.8	71.2/36.6/48.4	72.1/37.1/49.0	72.1/37.7/49.5	72.1/38.2/49.9	72.5/38.7/50.4	72.2/39.1/50.7	71.7/39.5/50.9	72.1/39.9/51.4	72.3/40.3/51.8	71.4/40.7/51.8
PLLaVA-34B	73.7/35.7/48.1	73.8/36.4/48.8	74.7/36.9/49.4	74.7/37.5/49.9	75.2/38.0/50.5	74.9/38.5/50.9	75.2/39.0/51.4	75.0/39.4/51.7	75.0/39.8/52.0	75.6/40.2/52.5	74.9/40.7/52.7
Qwen2-VL-7B	75.8/47.0/58.0	76.3/48.0/58.9	77.0/48.8/59.7	77.1/49.5/60.3	77.3/50.2/60.9	77.8/50.9/61.5	77.8/51.5/62.0	77.4/52.1/62.3	78.1/52.6/62.9	77.7/53.2/63.1	77.4/53.7/63.4
Qwen2-VL-72B	75.5/48.4/59.0	76.5/49.3/60.0	76.4/50.1/60.5	76.8/50.8/61.2	77.0/51.5/61.7	77.7/52.1/62.4	77.5/52.8/62.8	77.2/53.3/63.1	77.3/53.8/63.5	77.4/54.4/63.9	77.7/54.9/64.3
Tarsier-7b	75.4/42.3/54.2	75.8/42.9/54.8	76.1/43.5/55.4	76.5/44.1/55.9	76.4/44.7/56.4	76.7/45.2/56.9	76.6/45.7/57.3	76.8/46.2/57.7	77.0/46.7/58.1	76.5/47.1/58.3	76.5/47.6/58.7
Tarsier-34b	77.6/42.6/55.0	77.7/43.3/55.6	77.8/43.9/56.1	78.5/44.5/56.8	79.4/45.1/57.5	79.6/45.7/58.0	79.6/46.2/58.5	79.3/46.7/58.8	78.9/47.2/59.0	79.6/47.6/59.6	79.5/48.1/59.9
Llama-3-VILA1.5-8b	74.1/40.9/52.7	75.8/41.7/53.8	75.6/42.3/54.3	76.1/42.9/54.9	75.8/43.6/55.4	75.6/44.2/55.8	75.9/44.8/56.3	75.9/45.3/56.7	76.4/45.8/57.3	75.6/46.3/57.4	75.9/46.8/57.9
VILA1.5-13b	72.9/41.0/52.5	73.3/41.8/53.2	74.0/42.4/53.9	75.0/43.1/54.7	74.8/43.8/55.2	73.9/44.4/55.4	75.1/44.9/56.2	74.8/45.5/56.6	75.4/46.0/57.1	75.1/46.5/57.4	74.9/47.0/57.7
VILA1.5-40b	76.3/44.2/56.0	76.6/45.0/56.7	77.6/45.7/57.5	77.7/46.4/58.1	78.0/47.0/58.7	78.3/47.6/59.2	77.9/48.2/59.6	78.2/48.7/60.1	78.3/49.2/60.5	78.9/49.8/61.0	78.6/50.3/61.3
Closed-Source MLLMs											
Gemini-1.5-Pro	78.1/61.9/69.1	78.7/62.2/69.5	79.4/62.5/70.0	80.4/62.9/70.6	80.5/63.3/70.8	80.9/63.6/71.2	81.2/64.0/71.6	81.1/64.4/71.8	81.1/64.8/72.0	81.1/65.1/72.2	81.0/65.5/72.4
GPT-4o	81.7/54.2/65.2	82.5/54.9/65.9	83.2/55.5/66.6	83.6/56.2/67.2	84.1/56.9/67.9	83.8/57.5/68.2	84.2/58.1/68.8	84.4/58.7/69.2	84.5/59.2/69.6	84.1/59.7/69.8	84.5/60.2/70.3

Table 9: Performance on MCTS-VCB across different similarity threshold used in merging key points from different nodes s . Each cell contains "Precision / Recall / F1 Score"