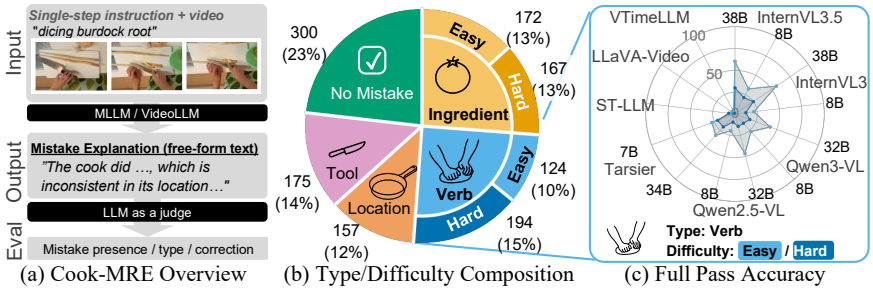
Benchmarking MLLMs on Mistake Recognition and Explanation in Single-Step Components of Cooking

Shun Takashige[2,1], Atsushi Hashimoto[3], and Shin'ichi Satoh[1,2]

[1] National Institute of Informatics, Japan
[2] The University of Tokyo, Japan
[3] OMRON SINIC X Corp., Japan



(a) Cook-MRE Overview (b) Type/Difficulty Composition (c) Full Pass Accuracy

Fig. 1: Overview of Cook-MRE. MLLMs receive a single-step cooking video with its instruction and must recognize and explain mistakes. The task involves four mistake types in videos, each with difficulty levels. Full Pass Accuracy reveals challenges in explaining verb mistakes, especially in hard cases (verb category only).

Abstract. While mistake understanding in procedural video is advancing toward increasingly complex multi-step tasks and diverse mistake types, it remains unclear whether current MLLMs can even recognize and explain mistakes at individual step levels. This study proposes Cooking Mistake Recognition and Explanation (Cook-MRE), a dataset for assessing MLLMs in-depth performance in understanding cooking mistakes within each single-step. The evaluation uses an LLM-based approach along two aspects: recognition (mistake presence and type classification) and explanation (error correction). By focusing on mistakes in the form of deviations from instructions, Cook-MRE reuses an existing cooking video dataset and synthesizes mistake samples through modification of text instructions. It covers four basic operation elements and includes difficulty labels based on visual distinguishability. Evaluation of 13 MLLMs showed that our comprehensive metric, Full Pass Accuracy, was substantially lower than mistake presence accuracy alone, indicating the difficulty of explaining mistakes in detail even for single steps. Analysis by utilizing our type and difficulty labels showed notably low performance on visually challenging problems, and scaling model size yields limited improvement on them. The dataset and code are publicly available at https://github.com/Shun-Takashige/cook-mre-benchmark.

Keywords: procedural video · mistake explanation · multi-modal LLM

1 Introduction

Multimodal large language models (MLLMs) have brought significant advances in vision-and-language understanding, and their application to procedural activities—tasks defined by a sequence of steps toward a goal—is attracting growing interest [22, 30, 31, 35]. Among such activities, cooking is a representative domain [5, 21, 42], and mistake detection is recognized as a particularly challenging problem [1]. With such advances in MLLMs, there is increasing interest not only in detection but also in explaining what went wrong for task assistance [1, 4].

Research on procedural video understanding spans from action recognition and segmentation [5, 8, 23, 26, 36] to temporal localization of mistakes across multi-step sequences [7, 10, 15, 25, 27, 34] and question answering about them [11]. The scope of targeted mistakes has also broadened from visually obvious errors to those more naturally captured by non-visual means, such as temperature or measurement. [25] The field is thus advancing toward longer, more complex videos with increasingly diverse mistake types.

However, recent benchmarks have revealed that MLLMs often fail on visual tasks that are straightforward for humans [6, 12, 16, 17, 20, 38, 41]. Procedural activities involve fine-grained details such as hand movements and object states, making them particularly demanding for visual understanding. Even within basic operation types, mistakes can manifest in many different ways, and it remains unclear whether current models can reliably recognize and explain such basic mistakes within a single step. Establishing this foundation would provide a solid basis for tackling more complex, multi-step mistakes. A systematic benchmark for single-step mistake recognition and explanation is therefore needed.

Evaluating this capability requires a benchmark that satisfies three properties: coverage of mistake types in single step, control over difficulty, and scalable assessment of free-form explanations (Fig. 1(a)). First, understanding four basic operation elements—*verb*, *ingredient*, *tool*, and *location*—is fundamental, and mistakes can occur in any of these elements. Second, even within the same element, visual distinguishability varies widely: recognizing different cutting techniques demands greater precision than distinguishing cutting from boiling. Third, recent advances in LLMs have made it possible to evaluate free-form explanations when ground truth is available at scale [40].

Conventional mistake datasets in procedural activities have been constructed to include deviations from instructions and anomalous actions as a single set, requiring dedicated video collection for each mistake condition [7, 10, 15, 25, 27, 34]. Because the mistake types are intentionally designed at the video acquisition stage, type labels can be obtained without additional annotation cost. However, difficulty levels and detailed explanations require substantial manual effort, and datasets containing these elements at a meaningful scale do not yet exist.

From this observation, we propose a novel dataset, Cook-MRE[4], which consists of 1,289 samples with 989 mistakes across four elements and two difficulty levels, with explanation ground truth (Fig. 1(b)). By excluding anomalous actions from

[4] Cook-MRE stands for Cooking Mistake Recognition and Explanation.

Table 1: Comparison of datasets for mistake detection in procedural videos. Single Elem. Dev. shows samples with deviations in one of four operation elements (verb, ingredient, tool, location). CaptainCook4D and ProMQA counts are from reclassifying their annotated error types into these four categories.

Dataset	Domain	Video Input	Temp. Loc.	Mistake Scope	# Mistakes	# Single Elem. Dev.	Difficulty	Output Expl.
EgoPER [15]	Cooking	Multi-step	✓	Proc. + Exec.	∼963	N/A	✗	✗
EgoOops [10]	Diverse	Multi-step	✓	Proc. + Exec.	233	95	✗	✗
CaptainCook4D [25]	Cooking	Multi-step	✗	Proc. + Exec.	2,566	597	✗	✗
ProMQA [11]	Cooking	Multi-step	✗	Proc. + Exec.	401	261	✗	✓
Cook-MRE (Ours)	Cooking	Single-step	✗	Exec.	989	**989**	✓	✓

consideration, we developed a pipeline to virtually generate this dataset; for each single step segment in an existing procedural video dataset, we modified its corresponding instructional text with LLM, treating inconsistencies between single-step video segment and instruction as virtual mistakes. Systematically replacing a part of the instruction allows us to control mistake types and visual distinguishability. Ground-truth explanations can be generated from the edited text.

The contributions of this study are threefold:

1. Focusing on the capability to understand mistakes in single procedural steps, we design a MLLM benchmark task that evaluates this capability along two aspects: *recognition* (error presence and type classification) and *explanation* (free-form description for error correction).
2. We propose a low-cost pipeline to generate mistake samples from normal procedural videos, creating the Cook-MRE dataset at a meaningful scale.
3. We evaluate 13 MLLMs and reveal that even these basic single-step mistakes pose a significant challenge: the best model achieves only approximately 57% on our comprehensive metric. Substantial performance gaps exist between easy and hard problems, and model size scaling yields limited improvement, with consistent gains observed only in the InternVL family (Fig. 1(c)).

2 Related Work

Mistake detection in procedural activity videos has been studied across diverse domains [10, 15, 25, 27]. Early work focused on detecting anomalous actions from video alone [7, 27, 28]. However, while anomaly detection captures visually obvious irregularities, it cannot distinguish actions that appear normal yet deviate from the intended instructions. With advances in vision-language models, mistake detection tasks have emerged that take both video and text as input, verifying whether seemingly normal actions are consistent with the given instructions [10, 11, 15, 25]. Recent methods further improve detection via multiple action prototype modeling [14] or action effect representations [9]. Detecting such deviations

from instructions is a setting specific to procedural activities, where performers are expected to follow step instructions. Understanding to what extent their actions conform to these instructions is essential for educational support and task assistance. From this perspective, we focus on deviations from textual instructions. These mistakes have been categorized into Execution Mistakes, such as insufficient heating time or incorrect use of a tool within a single step, and Procedural Mistakes, such as skipping a step or performing steps in the wrong order [1]. Fine-grained mistake attribution across semantic, temporal, and spatial dimensions in egocentric video has also been explored [18]. Recent studies have proposed various mistake categories, and benchmarks incorporating these categories have been developed [10, 11, 15, 25].

However, these benchmarks take long multi-step videos as input. Consequently, no recent work targets single-step videos alone. Existing research has moved to more complex multi-step settings, where the mainstream task is *detection*—determining the presence of a mistake together with its temporal localization within a multi-step sequence. Yet, detection only judges whether a mistake exists. It does not measure how well models understand and explain the deviation from the instruction. Although recent work has begun using MLLMs to generate explanations for anomalous actions [11, 24, 30], whether models can produce accurate explanations even for deviations from instructions in single-step videos remains underexplored. If models cannot recognize the presence of a mistake and verify their explanations even in the simpler case of a single step, it is fundamentally difficult to solve the harder multi-step task. Therefore, verification of mistake explanation at the single-step level is pivotal.

Although single-step mistakes are partly covered by existing datasets [10, 11, 15, 25], their mistake categories involve many compounding factors. For instance, elements such as temperature and measurement are better captured with dedicated sensors than with visual recognition alone. In contrast, deviations in the four elements of verb, ingredient, tool, and location inherently require visual recognition and can serve as a primitive test of a model's ability to understand mistakes. Yet existing datasets provide limited coverage of such deviations. As Tab. 1 shows, the number of single element deviations is at most around 600 samples, and ProMQA [11], the only dataset that requires explanations, contains only 261 such samples. Moreover, even among these four elements, visual distinguishability ranges from easy to hard, yet no existing dataset distinguishes difficulty levels. At present, it remains unexplored how models' ability to explain single element deviations varies across element types and difficulty levels—an analysis essential before tackling more complex tasks such as temporal localization. This study constructs a benchmark that enables such fine-grained analysis.

3 Cook-MRE Dataset

This section describes the pipeline of the Cook-MRE dataset and its statistics. Section 3.1 introduces the source data used for dataset construction. Section 3.2

defines mistake types and difficulty levels. Section 3.3 details the annotation pipeline, including preprocessing, LLM-based generation, and human review. Section 3.4 summarizes the dataset statistics.

3.1 Collecting Source Data

We built the Cook-MRE dataset upon the COM Kitchens dataset [21], a collection of cooking videos with precise annotations. In this collection, each cooking step execution is annotated with start and end frames, as well as a textual description specifying a verb, naturally with ingredients, tools, and location. The dataset is recorded with a fixed third-person view from a simple smartphone setup. This setup enabled coverage over 70 kitchen environments, making it the most comprehensive choice for offering diverse recipes, chefs, and backgrounds. This fixed overhead viewpoint differs from egocentric capture, and generalization across viewpoints remains an open question.

3.2 Mistake Definition

We define a single element deviation as a mistake within a single cooking step where exactly one operation element clearly differs between an instruction and the video content. We focus on four primitive elements that inherently require visual recognition—verb, ingredient, tool, and location—and represent a step description T by $(w^{\text{verb}}, w^{\text{ingr}}, w^{\text{tool}}, w^{\text{loc}})$. The verb element encompasses the action itself as well as adverbial modifiers such as cutting style. For example, given $T =$ "Slice burdock root on the diagonal," where $w^{\text{verb}} =$ *slice* and $w^{\text{ingr}} =$ *burdock root*. In this case, an instruction $I =$ "Dice burdock root" constitutes a verb mistake, as only the verb element differs. Since each mistake is restricted to a single element, the mistake type $c \in \{\text{verb}, \text{ingr}, \text{tool}, \text{loc}\}$ is uniquely determined.

3.3 Synthesizing Mistake Samples

Evaluation samples are constructed by modifying the step description while keeping the video unchanged. For each step, the original description serves as the basis for a *correct sample*, while replacing one element yields a *mistake sample*.

The annotation pipeline consists of three stages: preprocessing, LLM-based generation, and human review (Fig. 2). In preprocessing, we filter source steps to ensure that each problem is visually assessable: steps involving liquid seasonings, powdered condiments, or mixture of ingredients are excluded due to their extremely low visual distinguishability. We also limit each step to approximately 120 seconds and discard those with only one operation element in their text descriptions, so that multiple mistake types can be generated per step. Approximately 350 steps per element category are selected, balancing diversity of verbs and ingredients while avoiding excessive overlap of source videos.

For each retained step j with original description T_j, Qwen2.5-7B [37] generates a set of candidate instructions Q'_j by replacing one element at a time.

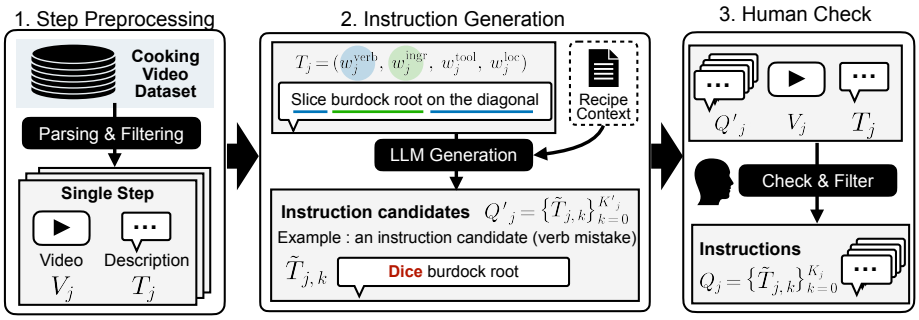


Fig. 2: Annotation pipeline. Each source step (V_j, T_j) is parsed into four elements, an LLM generates candidate instructions Q'_j by replacing one element at a time, and a human annotator filters them into the final set Q_j.

The surrounding recipe steps are provided as context, so that each candidate represents a realistic instruction that could plausibly occur in practice. The LLM also adjusts surrounding words for grammatical naturalness (*e.g.*, removing "on the diagonal" when changing *slice* to *wash*). All candidates are constructed to avoid language-biased samples—instructions that would be identifiable as incorrect from text alone without seeing the video.

Among the generated candidates, we define two difficulty levels—**easy** and **hard**—based on the semantic proximity between the original and replacement word. For tool and location, semantically distant replacements (which would constitute easy samples) are already excluded as language-biased, and the remaining candidates do not exhibit sufficient range to warrant a difficulty split. Difficulty levels are therefore defined only for verb and ingredient. For verbs, hard samples involve semantically close actions such as confusion within the cut-family (*e.g.*, *slice* vs. *dice*) or fine manual operations (*e.g.*, *tear* vs. *fold*); easy samples involve clearly distinct actions. For ingredients, hard samples involve visually similar items (*e.g.*, cabbage vs. lettuce) or items from the same food category; easy samples involve obviously different ingredients. Detailed definitions of difficulty groups are provided in the supplementary material. This process yields multiple candidate instructions per step across type and difficulty combinations.

All generated candidates Q'_j are reviewed by a single annotator. First, the annotator inspects the actual video frames and excludes cases where the target operation is occluded or out of frame and cannot be visually verified. Second, samples in which the replacement word is effectively synonymous with the original (*e.g.*, *chop* and *cut*) are removed, as they do not constitute a clear mistake. Third, the annotator verifies the consistency of difficulty labels: whether hard labels are appropriately assigned and whether the same mistake pair (*e.g.*, *slice* $\leftrightarrow$ *dice*) receives a consistent difficulty label across different steps. The approved set $Q_j \subseteq Q'_j$ contains only samples that are visually verifiable, clearly constitute a mistake, and carry consistent difficulty labels. Details of the pipeline, including prompts and filtering criteria, are provided in the supplementary material.

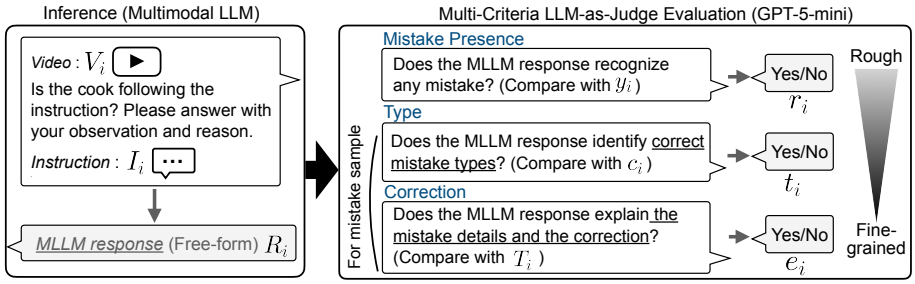


Fig. 3: Benchmark task and evaluation. The model receives (V_i, T_i) and produces a free-form response, which is then evaluated by an LLM judge on three criteria.

Each approved instruction becomes a **synthesized instruction** I_i—the procedure the person in the video should have followed. For a correct sample, I_i is the original description T_i itself; for a mistake sample, I_i is a modified instruction $\tilde{T}_i$ in which one element has been replaced. Each dataset sample pairs a video with its synthesized instruction as input $\mathbf{x}_i = (V_i,\ I_i)$, and is annotated with a ground-truth label $\mathbf{y}_i = (y_i,\ c_i,\ T_i)$ recording the binary judgment $(y_i \in \{0, 1\})$, the mistake type $(c_i \in \{\text{verb}, \text{ingr}, \text{tool}, \text{loc}, \text{none}\})$, and the original description as the correction reference.

3.4 Statistics

The resulting dataset contains 989 mistake samples and 300 correct samples (1,289 total), covering all four operation elements (Fig. 1(b)): verb (318), ingredient (339), tool (175), and location (157). Verb and ingredient account for larger portions as they are the most frequent operation elements in cooking videos. For difficulty, easy and hard levels are defined for verb and ingredient, yielding a balanced distribution of 515 easy and 474 hard samples overall. All video clips are single-step segments of at most 120 seconds, with only two exceptions. Every mistake sample is paired with the original description T_i as a correction reference, enabling large-scale evaluation of free-form explanations. Detailed dataset statistics are provided in the supplementary material.

4 Benchmark Task

4.1 Task Formulation

Given a sample $\mathbf{x}_i = (V_i,\ T_i)$ as defined in Sec. 3.3, the model is asked whether the cook in the video is following the instruction, and to provide its observation and reasoning (Fig. 3, left). The model produces a free-form textual response R_i; no predefined answer choices are provided, and the model is not told which element may be incorrect, so it must always consider all four types of possible mistakes. The response is expected to convey three pieces of information: (1) a

consistency judgment indicating whether the video follows the instruction, (2) an observation describing what actually occurs in the video, and (3) the reasoning that connects the observation to the judgment. The full prompt template is provided in the supplementary material.

4.2 Multi-Criteria LLM-as-Judge Evaluation

We evaluate the model response R_i on three criteria of increasing granularity—mistake presence check, type classification, and correction. Because these criteria require semantic interpretation of free-form text, we employ an LLM-as-judge with a dedicated prompt for each criterion. The judge reads R_i together with the corresponding ground truth and returns a binary score (Fig. 3, right). Specifically, it produces: $r_i \in \{0, 1\}$, whether the model's Yes/No judgment matches y_i; $t_i \in \{0, 1\}$, whether the reported discrepancy corresponds to the correct type c_i (mistake samples only); and $e_i \in \{0, 1\}$, whether the model's observation of the video content is semantically equivalent to T_i (mistake samples only). Cooking-domain synonyms and acceptable generalizations (*e.g.*, "lettuce" for "red leaf lettuce") are accepted when judging e_i. The reliability of this evaluation was validated against human annotations, achieving over 94% agreement for each criterion; details are provided in the supplementary material.

Based on these three judgments, we define the following metrics. **MPC F1** treats mistake samples ($y_i = 1$) as the positive class:

$$\text{MPC F1} = \frac{2\,\text{TP}}{2\,\text{TP} + \text{FP} + \text{FN}}, \tag{1}$$
$$\text{TP} = \sum_{i \in \mathcal{M}} r_i, \quad \text{FP} = \sum_{i \notin \mathcal{M}} (1 - r_i), \quad \text{FN} = \sum_{i \in \mathcal{M}} (1 - r_i),$$

where $\mathcal{M} = \{i \mid y_i = 1\}$. Since the dataset is imbalanced (989 mistake vs. 300 correct samples), we also report **MPC Macro F1**, the average of per-class F1-scores. Because type classification is meaningful only when the presence check is correct, **Type Accuracy** is defined as the product $r_i \cdot t_i$:

$$\text{Type Acc} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} r_i \cdot t_i. \tag{2}$$

Correction Accuracy further requires an accurate observation, yielding $r_i \cdot t_i \cdot e_i$:

$$\text{Corr Acc} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} r_i \cdot t_i \cdot e_i. \tag{3}$$

Since the goal of mistake explanation is to correctly describe the video content, this metric most directly reflects the capability. **Full Pass Acc.** extends evaluation to the entire dataset, requiring all three criteria for mistake samples and only r_i for correct samples:

$$\text{Full Pass Acc.} = \frac{1}{N} \sum_{i=1}^{N} p_i, \quad p_i = \begin{cases} r_i & \text{if } y_i = 0 \\ r_i \cdot t_i \cdot e_i & \text{if } y_i = 1 \end{cases} \tag{4}$$

5 Experiments

We evaluate 13 open-source MLLMs capable of processing video input and organize them into two groups. Group A consists of five general-purpose MLLM families—Qwen2.5-VL [3], Qwen3-VL [2], InternVL3 [43], InternVL3.5 [33], and Tarsier [32]—each providing a 7–8 B and a 30 B-class variant to enable controlled comparison across model sizes. Including both InternVL3 and InternVL3.5 further allows comparison across model versions within the same architecture. Group B adds three video-specialized models—ST-LLM [19], LLaVA-Video-7B [39], and VTimeLLM [13]—which employ temporal modeling components, allowing an examination of whether architectures designed for temporal reasoning confer advantages in mistake explanation. We additionally examine the role of temporal ordering via a frame-shuffling ablation across all models (see the supplementary material). We deploy all these models locally and exclude proprietary MLLMs since the COM Kitchens license prohibits transmitting videos to third parties.

All models are evaluated zero-shot on the full 1,289-sample test set. Several studies have investigated the relationship between the number of input frames and MLLM performance on video understanding benchmarks [17, 31, 38]. These works observe that model accuracy tends to saturate at relatively low frame counts, typically around 8–16 frames, even for videos under a minute. Although such findings are task-dependent and do not directly generalize to our setting, 16 frames lies within the commonly reported saturation range and is widely adopted as a standard default; we therefore use 16 uniformly sampled frames across all models to ensure a fair and consistent comparison. We use greedy decoding with a temperature of 0.1 across all models to ensure reproducibility. For the LLM-as-judge evaluation described in Sec. 4.2, we use GPT-5-mini [29]. All inference is conducted on a single NVIDIA V100 or A100 GPU.

5.1 Main Results

Table 2 summarizes the results. MPC F1 and MPC Macro F1 measure detection ability, while Type Accuracy and Corr. Accuracy assess explanation quality on mistake samples, with Corr. Accuracy additionally requiring an accurate observation on top of correct type classification. Even the best-performing model, InternVL3.5-38B, achieves an MPC F1 of 87.6% but only a Corr. Acc. of 50.2%, revealing a substantial gap between detection and explanation.

Group A models detect mistakes with reasonable accuracy, with MPC F1 ranging from 71.4 to 87.6%, and their MPC Macro F1 remains within about 10 points of MPC F1, indicating that detection is not heavily biased toward either class. However, Corr. Acc. drops to 24–50%, indicating that models often fail to ground their explanations in the actual video content even when they correctly detect a mistake. This gap suggests that the bottleneck lies not in detecting inconsistencies but in accurately perceiving and describing fine-grained visual details.

Group B models appear to detect mistakes but without genuine understanding. ST-LLM achieves an MPC F1 of 80.3%, comparable to Group A, yet its MPC

Table 2: Main results on Cook-MRE (%). Each evaluation metric is defined in Sec. 4.2, and the hint condition in Sec. 5.4. **Bold**: best; underline: second best.

		Recognition F1		Macro F1		Type Acc.	Correction Acc.		Full Pass	
Model	Size	Default	Hint	Default	Hint	Default	Default	Hint	Default	Hint
InternVL3.5	38B	**87.6**	**87.3**	**76.3**	**76.7**	**77.9**	**50.2**	**57.4**	55.5	**62.1**
	8B	80.4	76.8	68.9	64.0	65.0	32.9	40.8	43.4	47.8
InternVL3	38B	85.9	86.4	75.6	75.2	75.6	49.7	56.5	**57.0**	61.0
	8B	71.4	73.3	62.1	63.3	53.3	34.1	37.2	46.2	47.9
Qwen2.5-VL	32B	77.8	74.0	65.4	62.6	62.6	31.6	32.5	41.3	42.7
	7B	72.5	71.6	62.5	57.7	52.9	28.9	31.3	41.3	38.7
Qwen3-VL	32B	76.7	73.7	64.4	61.3	60.1	30.5	33.1	40.5	42.0
	8B	80.0	73.9	64.6	58.5	64.3	23.7	30.3	32.0	36.8
Tarsier	34B	75.6	81.0	63.6	53.7	55.9	29.1	40.2	39.6	36.3
	7B	82.3	82.7	48.0	42.1	63.8	26.1	20.1	22.3	15.7
ST-LLM	7B	80.3	86.8	51.7	43.4	46.3	7.8	2.6	10.7	2.0
LLaVA-Video	7B	26.0	25.6	31.1	30.2	9.8	2.4	3.4	21.3	21.2
VTimeLLM	7B	22.4	58.8	30.2	36.6	8.7	1.9	7.8	22.7	10.6

Macro F1 is only 51.7%—a gap of nearly 30 points revealing a strong bias toward predicting mistakes regardless of actual content. Similarly, Tarsier-7B shows an MPC F1 of 82.3% but a Macro F1 of only 48.0%. Their Corr. Acc. further confirms this: ST-LLM scores 7.8%, indicating that the model flags mistakes but cannot describe what actually happens in the video. LLaVA-Video-7B and VTimeLLM produce MPC F1 below 26.0%, suggesting that these models struggle to make reliable judgments in the first place.

5.2 Scaling Effect by Type and Difficulty

Because Cook-MRE provides type and difficulty labels, we can examine where scaling effects are most pronounced rather than observing only aggregate trends. We focus on the three Group A families, each offering a large and small variant, and compare their Full Pass Accuracy difference on verb and ingredient problems—the two element types that have both easy and hard difficulty levels (Fig. 4).

For verb easy problems, most families show positive scaling deltas, indicating that increased model size helps when visual distinctions are relatively clear. However, this benefit diminishes on hard problems: hard deltas remain near zero for Qwen2.5-VL and Qwen3-VL, and for ingredient the same families even show slight negative deltas on hard problems.

The InternVL family is a notable exception. InternVL3 achieves the largest delta on verb easy (+35.5) and maintains substantial gains on verb hard (+12.4), ingredient easy (+14.0), and ingredient hard (+17.4). InternVL3.5 follows a similar pattern with positive deltas across all four conditions. This is the only

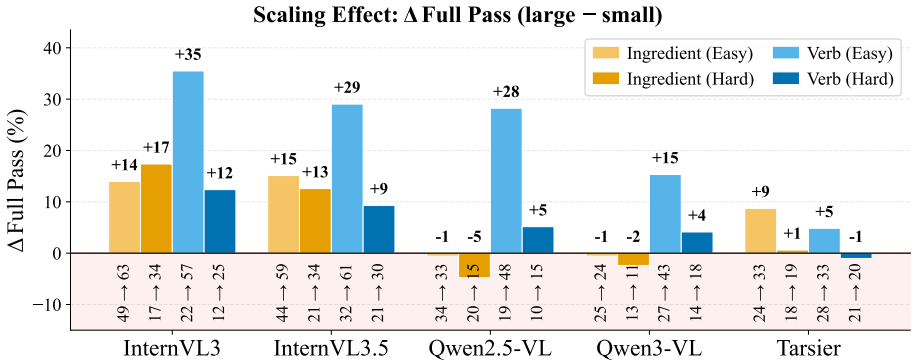


Fig. 4: Scaling effect on Full Pass Accuracy (%) for verb and ingredient, measured as the difference between the large and small variant within each Group A family.

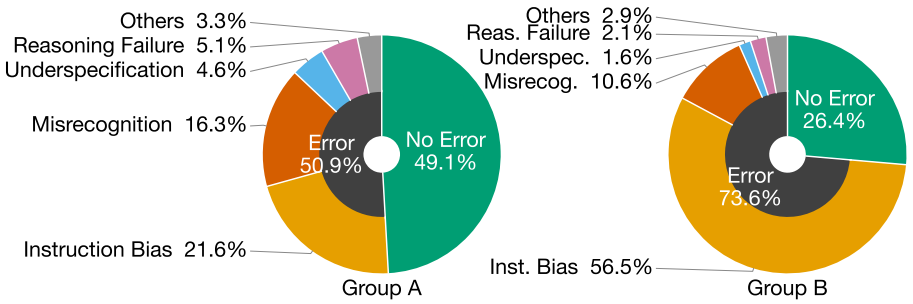


Fig. 5: Error type distribution averaged across models in Group A and Group B. See the supplementary materials for per-model distribution.

family for which scaling yields uniformly positive effects regardless of problem type or difficulty.

One architectural difference that may explain this pattern is vision encoder capacity. Qwen2.5-VL [3], Qwen3-VL, and Tarsier all use a fixed-size vision encoder across model sizes, scaling only the LLM decoders. In contrast, InternVL3 [43] and InternVL3.5 increase the vision encoder size alongside the LLM when moving to the larger variant. For verb problems, this is consistent with the finding of ProMQA [11], which reported that LLM scaling alone yields limited improvement on technical error understanding. These observations suggest a family-level correlation between vision encoder capacity and scaling gain, though other architectural differences may also contribute.

5.3 Error Analysis

Given the importance of scaling the vision encoder observed in Sec. 5.2, we further analyze error types to understand why models fall short of Full Pass Accuracy.

For this purpose, we classify incorrect explanations using GPT-5-mini [29] to identify structural causes for the limited improvement with increased model size.

We define the following four error categories as general failure patterns of MLLMs. **Instruction Bias**: the model does not refer to the video content and instead generates an explanation driven by the textual instruction alone. **Misrecognition**: the model attends to the video but incorrectly identifies the target object or action. **Reasoning Failure**: visual recognition is accurate, yet the model draws an incorrect conclusion in its explanation. **Underspecification**: recognition is broadly correct but the explanation lacks the specificity required for a complete answer. These four categories are derived from two independent axes—visual recognition quality and explanation specificity. The remaining cases are categorized as **Others**. We confirmed that categorization by the LLM and by humans matched with 89.2% accuracy, and the four categories cover over 95% of all observed failures. See the supplementary material for further details of categorization reliability.

Instruction Bias is the most common error category in both groups (Fig. 5), accounting for 21.6% of errors in Group A and 56.5% in Group B. This indicates that generating explanations driven by textual instructions rather than properly analyzing the video content is a widespread failure mode among current MLLMs.

In Group A, beyond Instruction Bias, Misrecognition (16.3%) and Reasoning Failure (5.1%) together account for 21.4% of errors. Since both categories by definition involve attending to the video, a substantial portion of Group A errors arise not from ignoring the video but from incorrect recognition or reasoning about its content. This suggests that strengthening visual perception and reasoning capabilities is key to further improving Group A models.

In Group B, 56.5% of errors are Instruction Bias, with Misrecognition at 10.6% and Reasoning Failure and Underspecification each below 2.2%. The overwhelming concentration in Instruction Bias indicates that, when these models fail, they do not meaningfully refer to the video and largely remain at the text-processing stage—a qualitatively different failure mode from Group A.

5.4 Effect of Hint

As a task ablation, we simplify the MRE task by providing the mistake types, allowing models to bypass the MRE Type Classification subtask. In the default condition used so far, the model must consider all four possible mistake types simultaneously. We use type-specific hint prompts, in which the mistake category is explicitly specified so that the model focuses on one element at a time (see the supplementary material for the exact prompt). Given the mistake type, Type Accuracy is redundant; thus we compare the default and hint conditions based on MPC F1, Corr. Accuracy, and Full Pass Accuracy. For correct samples, the model must answer "no mistake" to all applicable type-specific prompts, structurally increasing false-positive opportunities by up to four times. Results under both conditions are reported in Tab. 2.

For Group A, the InternVL family shows the largest improvement under hint prompts, with Corr. Accuracy increasing by up to +8.0 points and Full Pass

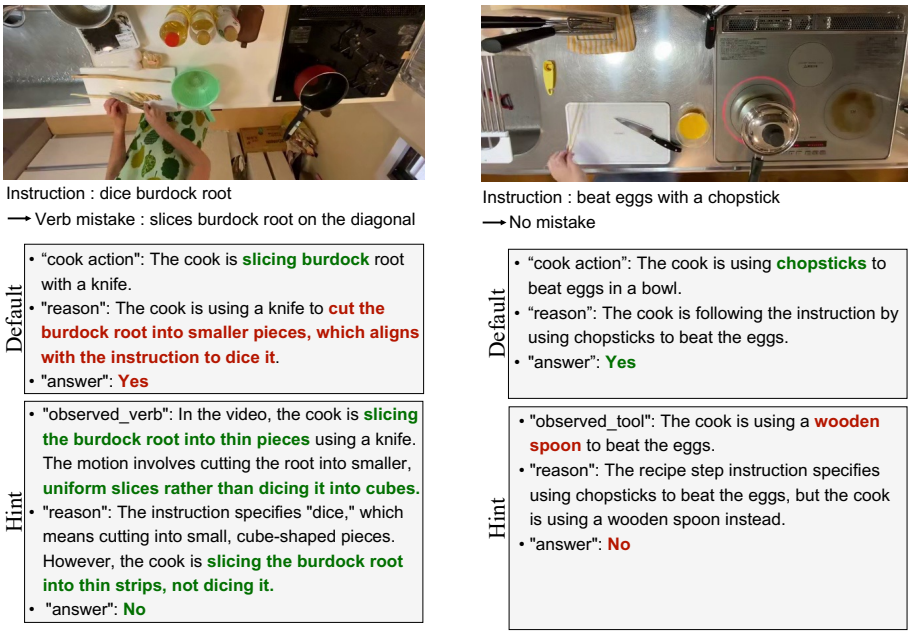


Fig. 6: Qualitative comparison of default and hint conditions on InternVL3-38B. (a) Improvement: the default response overlooks the difference between "dice" and "slice," while the hint prompt leads to a correct and detailed explanation. (b) Regression: the default response correctly identifies no mistake, but the hint prompt causes the model to hallucinate a tool that does not appear in the video.

Accuracy by up to +6.6 points. Other families show smaller or mixed effects. For example, models like Qwen2.5-VL-32B and Qwen3-VL-32B exhibit a decrease in MPC F1, indicating that narrowing the focus can also increase false positives.

For Group B, hint prompts increase MPC F1 for some models—VTimeLLM rises from 22.4% to 58.8%, and ST-LLM from 80.3% to 86.8% —yet MPC Macro F1 does not improve accordingly, suggesting that the apparent gain reflects increased bias toward predicting mistakes rather than genuine improvement. However, Full Pass Accuracy consistently deteriorates across all three models.

Figure 6 illustrates how hint prompts can both improve and degrade the response of InternVL3-38B on individual samples. In the improvement case (left), without hints, InternVL3-38B overlooks the distinction between "dice" and "slice" and judges the action as consistent. With a hint, the model correctly identifies that the cook is slicing into thin pieces rather than dicing into cubes. In the regression case (right), the same model correctly answers "yes" without hints for a sample with no mistake. However, with a hint, it reports a "wooden spoon" that does not appear in the video and incorrectly flags a tool mistake.

These quantitative and qualitative results suggest that type-specific hint prompts direct the model's attention toward fine-grained details that are otherwise overlooked, which is consistent with the prevalence of Instruction Bias observed

in Sec. 5.3. However, this narrowed focus also introduces a bias toward reporting mistakes, leading to hallucinated observations as illustrated in Fig. 6 (b). This effect is particularly pronounced for Group B models, whose lower baseline accuracy makes them more susceptible to errors as the number of judgments increases. The effect of hints is thus a trade-off between improved attention to relevant details and increased risk of false positives.

6 Conclusion

This study proposed Cook-MRE, a benchmark for evaluating MLLMs' ability to recognize and explain cooking mistakes in single-step videos. By reusing the existing cooking video dataset and synthesizing mistake samples through text instruction modification, Cook-MRE provides 1,289 evaluation samples containing 989 mistakes across four operation elements and two difficulty levels. An LLM-based evaluation along three criteria—mistake presence, type, and correction—enables large-scale in-depth assessment of free-form explanation.

Evaluation of 13 MLLMs revealed that performance degrades substantially from recognition to explanation, with even the best model achieving only approximately 57% Full Pass Accuracy (Sec. 5.1). Scaling effects were largely limited to the InternVL family, while other families showed inconsistent or negligible gains (Sec. 5.2). Error analysis showed that Group A models primarily suffer from misrecognition and reasoning failures, while Group B models are dominated by Instruction Bias (Sec. 5.3). The hint condition improved Correction Accuracy for a subset of Group A models but failed to benefit Group B, suggesting that explaining mistakes fundamentally requires improved fine-grained visual recognition rather than prompt engineering (Sec. 5.4).

Currently, Cook-MRE is limited to the cooking domain, single-step videos, and mistakes involving four operation elements. Mistakes related to measurement, timing, or temperature, as well as those requiring multi-step context, are outside the current scope. Furthermore, because mistake samples are synthesized by modifying text instructions, some samples may describe situations that are unlikely to occur in actual cooking practice. Investigating the extent to which these factors affect evaluation outcomes and extending the framework to other procedural domains remain directions for future work.

In addition, the current evaluation covers 13 open-source MLLMs. Proprietary models, which may exhibit different performance characteristics, were not included. Expanding the evaluation to encompass a broader range of models would provide a more comprehensive picture of the current state of mistake explanation capability.

Acknowledgements

This work was partly supported by JST ASPIRE Program Grant Number JPMJAP2303 and JST K Program Grant Number JPMJKP25V1, Japan.

References

1. Bacharidis, K., Argyros, A.A.: Vision-based mistake analysis in procedural activities: A review of advances and challenges. arXiv preprint arXiv:2510.19292 (2025)
2. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-VL technical report. arXiv preprint arXiv:2511.21631 (2025)
3. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2.5-VL technical report. arXiv preprint arXiv:2502.13923 (2025)
4. Cong, L., Zhang, Z., Wang, X., Di, Y., Jin, R., Gerasimiuk, M., Wang, Y., Dinesh, R.K., Smerkous, D., Smerkous, A., et al.: LabOS: The AI-XR co-scientist that sees and works with humans. arXiv preprint arXiv:2510.14861 (2025)
5. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: The EPIC-Kitchens dataset: Collection, challenges and baselines. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(11), 4125–4141 (2020)
6. Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., et al.: Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 24108–24118 (2025)
7. Ghoddoosian, R., Dwivedi, I., Agarwal, N., Dariush, B.: Weakly-supervised action segmentation and unseen error detection in anomalous instructional videos. In: Proc. of the International Conference on Computer Vision (ICCV). pp. 10128–10138 (2023)
8. Grauman, K., Westbury, A., Torresani, L., Kitani, K., Malik, J., Afouras, T., Ashutosh, K., Baiyya, V., Bansal, S., Boote, B., et al.: Ego-Exo4D: Understanding skilled human activity from first-and third-person perspectives. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19383–19400 (2024)
9. Guo, W., Pu, Y., Kong, Y.: Procedural mistake detection via action effect modeling. arXiv preprint arXiv:2512.03474 (2025)
10. Haneji, Y., Nishimura, T., Kameko, H., Shirai, K., Yoshida, T., Kajimura, K., Yamamoto, K., Cui, T., Nishimoto, T., Mori, S.: EgoOops: A dataset for mistake action detection from egocentric videos referring to procedural texts. In: Proc. of the International Conference on Computer Vision Workshop (ICCVW). pp. 2690–2700 (October 2025)
11. Hasegawa, K., Imrattanatrai, W., Cheng, Z.Q., Asada, M., Holm, S., Wang, Y., Fukuda, K., Mitamura, T.: ProMQA: Question answering dataset for multimodal procedural activity understanding. In: Proc. of the Conference on Human Language Technologies (NAACL-HLT). pp. 11598–11617 (April 2025)
12. Hu, K., Wu, P., Pu, F., Xiao, W., Zhang, Y., Yue, X., Li, B., Liu, Z.: Video-MMMU: Evaluating knowledge acquisition from multi-discipline professional videos. arXiv preprint arXiv:2501.13826 (2025)
13. Huang, B., Wang, X., Chen, H., Song, Z., Zhu, W.: VTimeLLM: Empower LLM to grasp video moments. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14271–14280 (2024)
14. Huang, W.J., Li, Y.M., Xia, Z.W., Tang, Y.M., Lin, K.Y., Hu, J.F., Zheng, W.S.: Modeling multiple normal action representations for error detection in procedural tasks. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27794–27804 (2025)

15. Lee, S.P., Lu, Z., Zhang, Z., Hoai, M., Elhamifar, E.: Error detection in egocentric procedural task videos. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18655–18666 (June 2024)
16. Li, C., Im, E.W., Fazli, P.: VidHalluc: Evaluating temporal hallucinations in multimodal large language models for video understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13723–13733 (2025)
17. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Luo, P., et al.: MVBench: A comprehensive multi-modal video understanding benchmark. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22195–22206 (2024)
18. Li, Y., Jain, A., Bellos, F., Corso, J.J.: Mistake attribution: Fine-grained mistake understanding in egocentric videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23966–23976 (2026)
19. Liu, R., Li, C., Tang, H., Ge, Y., Shan, Y., Li, G.: ST-LLM: Large language models are effective temporal learners. In: Proc. of the European Conference on Computer Vision (ECCV). p. 1–18 (2024)
20. Liu, Y., Li, S., Liu, Y., Wang, Y., Ren, S., Li, L., Chen, S., Sun, X., Hou, L.: TempCompass: Do video LLMs really understand videos? In: Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL). pp. 8731–8772 (2024)
21. Maeda, K., Hirasawa, T., Hashimoto, A., Harashima, J., Rybicki, L., Fukasawa, Y., Ushiku, Y.: COM Kitchens: An unedited overhead-view video dataset as a vision-language benchmark. In: Proc. of the European Conference on Computer Vision (ECCV) (2024)
22. Nagarajan, T., Torresani, L.: Step differences in instructional video. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18740–18750 (2024)
23. Nishimoto, T., Nishimura, T., Yamamoto, K., Shirai, K., Kameko, H., Haneji, Y., Yoshida, T., Kajimura, K., Cui, T., Nishiwaki, C., et al.: BioVL-QR: Egocentric biochemical vision-and-language dataset using micro QR codes. In: Proc. of the IEEE International Conference on Image Processing (ICIP). pp. 695–700. IEEE (2025)
24. Patsch, C., Wu, Y., Zakour, M., Salihu, D., Steinbach, E.: MistSense: Versatile online detection of procedural and execution mistakes. In: Proc. of the International Conference on Computer Vision (ICCV). pp. 14528–14537 (2025)
25. Peddi, R., Arya, S., Challa, B., Pallapothula, L., Vyas, A., Gouripeddi, B., Zhang, Q., Wang, J., Komaragiri, V., Ragan, E., Ruozzi, N., Xiang, Y., Gogate, V.: CaptainCook4D: A dataset for understanding errors in procedural activities. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Proc. of the Advances in Neural Information Processing Systems (NeurIPS). vol. 37, pp. 135626–135679 (2024)
26. Perrett, T., Darkhalil, A., Sinha, S., Emara, O., Pollard, S., Parida, K.K., Liu, K., Gatti, P., Bansal, S., Flanagan, K., et al.: HD-EPIC: A highly-detailed egocentric video dataset. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 23901–23913 (2025)
27. Schoonbeek, T.J., Houben, T., Onvlee, H., Van der Sommen, F., et al.: Industreal: A dataset for procedure step recognition handling execution errors in egocentric videos in an industrial-like setting. In: Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 4365–4374 (2024)

28. Sener, F., Chatterjee, D., Shelepov, D., He, K., Singhania, D., Wang, R., Yao, A.: Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 21096–21106 (June 2022)
29. Singh, A., Fry, A., Perelman, A., Tart, A., Ganesh, A., El-Kishky, A., McLaughlin, A., Low, A., Ostrow, A., Ananthram, A., et al.: Openai gpt-5 system card. arXiv preprint arXiv:2601.03267 (2025)
30. Storks, S., Bar-Yossef, I., Li, Y., Zhang, Z., Corso, J.J., Chai, J.: Transparent and coherent procedural mistake detection. In: Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 13979–14013 (2025)
31. Tateno, M., Kato, G., Hara, K., Kataoka, H., Sato, Y., Yagi, T.: HanDyVQA: A video qa benchmark for fine-grained hand-object interaction dynamics. arXiv preprint arXiv:2512.00885 (2025), (Accepted to CVPR 2026)
32. Wang, J., Yuan, L., Zhang, Y., Sun, H.: Tarsier: Recipes for training and evaluating large video description models. arXiv preprint arXiv:2407.00634 (2024)
33. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025)
34. Wang, X., Kwon, T., Rad, M., Pan, B., Chakraborty, I., Andrist, S., Bohus, D., Feniello, A., Tekin, B., Frujeri, F.V., Joshi, N., Pollefeys, M.: HoloAssist: an egocentric human interaction dataset for interactive ai assistants in the real world. In: Proc. of the International Conference on Computer Vision (ICCV). pp. 20270–20281 (October 2023)
35. Xu, Y., Wu, Y., Yu, J., Yan, Z., Jiang, T., He, Y., Zhao, Q., Chen, K., Qiao, Y., Wang, L., et al.: ExpVid: A benchmark for experiment video understanding & reasoning. arXiv preprint arXiv:2510.11606 (2025)
36. Yagi, T., Ohashi, M., Huang, Y., Furuta, R., Adachi, S., Mitsuyama, T., Sato, Y.: Finebio: A fine-grained video dataset of biological experiments with hierarchical annotation. International Journal of Computer Vision **133**(10), 7352–7367 (2025)
37. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Dong, G., et al.: Qwen2.5 technical report. arXiv preprint arXiv:2412.15115 (2024)
38. Zhang, Y., Chew, Y., Dong, Y., Leo, A., Hu, B., Liu, Z.: Towards video thinking test: A holistic benchmark for advanced video reasoning and understanding. In: Proc. of the International Conference on Computer Vision (ICCV). pp. 20626–20636 (2025)
39. Zhang, Y., Wu, J., Li, W., Li, B., Ma, Z., Liu, Z., Li, C.: LLaVA-Video: Video instruction tuning with synthetic data. Transactions on Machine Learning Research (2025), https://openreview.net/forum?id=EElFGvt39K
40. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al.: Judging LLM-as-a-judge with MT-Bench and chatbot arena. Proc. of the Advances in Neural Information Processing Systems (NeurIPS) **36**, 46595–46623 (2023)
41. Zhou, J., Shu, Y., Zhao, B., Wu, B., Liang, Z., Xiao, S., Qin, M., Yang, X., Xiong, Y., Zhang, B., et al.: MLVU: Benchmarking multi-task long video understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13691–13701 (2025)
42. Zhou, L., Xu, C., Corso, J.: Towards automatic learning of procedures from web instructional videos. In: Proc. of the AAAI Conference on Artificial Intelligence (AAAI). vol. 32 (2018)

43. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., et al.: InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479 (2025)