Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

# TimeJudge: empowering video-LLMs as zero-shot judges for temporal consistency in video captions*#

Yangliu HU†1, Zikai SONG†‡1, Junqing YU1, Yiping Phoebe CHEN2, Wei YANG1

[1]*Huazhong University of Science and Technology, Wuhan 430074, China*

[2]*La Trobe University, Melbourne 3086, Australia*

†E-mail: huyangliu@hust.edu.cn; skyesong@hust.edu.cn

Received June 14, 2025; Revision accepted Oct. 20, 2025; Crosschecked Nov. 13, 2025; Published online Nov. 26, 2025

**Abstract:** Video large language models (video-LLMs) have demonstrated impressive capabilities in multimodal understanding, but their potential as zero-shot evaluators for temporal consistency in video captions remains underexplored. Existing methods notably underperform in detecting critical temporal errors, such as missing, hallucinated, or misordered actions. To address this gap, we introduce two key contributions. (1) TimeJudge: a novel zero-shot framework that recasts temporal error detection as answering calibrated binary question pairs. It incorporates modality-sensitive confidence calibration and uses consistency-weighted voting for robust prediction aggregation. (2) TEDBench: a rigorously constructed benchmark featuring videos across four distinct complexity levels, specifically designed with fine-grained temporal error annotations to evaluate video-LLM performance on this task. Through a comprehensive evaluation of multiple state-of-the-art video-LLMs on TEDBench, we demonstrate that TimeJudge consistently yields substantial gains in terms of recall and F1-score without requiring any task-specific fine-tuning. Our approach provides a generalizable, scalable, and training-free solution for enhancing the temporal error detection capabilities of video-LLMs.

**Key words:** Video large language model (Video-LLM); Multimodal large language model (MLLM); MLLM-as-a-Judge; Video caption; Benchmark

https://doi.org/10.1631/FITEE.2500412 **CLC number:** TP391

## 1 Introduction

Video large language models (video-LLMs) are rapidly transitioning from research prototypes to real-world products, making rigorous and scalable evaluation indispensable. Although human assessment remains the gold standard, it is slow, costly, and subjective, prompting a shift toward automated, model-based protocols. Building on the success of the "LLM-as-a-Judge" paradigm for text, the emerging "Video-LLM-as-a-Judge" framework proposes to apply powerful video-LLMs to score, rank, and filter the video captions generated by other video language models (VLMs) (Zheng et al., 2023; Liu and Zhang, 2025). Besides offering low-cost, high-throughput evaluation, this approach accelerates the evolution of video-LLMs from dialogue systems into general-purpose multimodal agents with broad applications in evaluation, alignment, retrieval, and reasoning (Bai YS et al., 2023; Lee et al., 2023; Li RS et al., 2023; Liang et al., 2023; Liao et al., 2024; Xu et al., 2024).

However, when applied to video-understanding benchmarks, existing video-LLM judges prove

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 62272184 and 62402189), the China Postdoctoral Science Foundation (Nos. 2024M751012, 2025T180429, and GZC20230894), and the Postdoctor Project of Hubei Province (No. 2024HBBHCXB014)

# Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2500412) contains supplementary materials, which are available to authorized users

ORCID: Zikai SONG, https://orcid.org/0009-0006-6651-2027

unreliable. As illustrated in Fig. 1, directly prompting them to verify the fidelity of machine-generated captions often yields incorrect verdicts, particularly regarding temporal coherence, missing events, hallucinated actions, or misordered sequences. We identify three underlying causes: (1) an over-reliance on static spatial cues (e.g., objects and scenes) at the expense of temporal perspicacity; (2) limited capacity for multi-hop, high-level reasoning in the temporal domain; (3) pronounced biases toward textual priors or visually dominant patterns rather than grounding decisions on video evidence. These shortcomings, rooted in both training data and model architecture, erode the reliability of video-LLM-based evaluation for temporally sensitive tasks such as event localization and action detection.

To mitigate these limitations, we observe that reducing cognitive load and explicitly weighting modality evidence make video-LLMs markedly more attuned to temporal inconsistencies. Building on this insight, we introduce TimeJudge, a zero-shot protocol that boosts a video-LLM's ability to spot temporal errors in captions. TimeJudge decomposes the original, open-ended verification task into a sequence of lightweight binary queries, nudging the model to inspect fine-grained temporal relations while dynamically calibrating its confidence according to the relative contributions of visual and textual cues. By lowering the reasoning burden and steering attention toward the relevant spatiotemporal evidence, TimeJudge delivers substantial gains in temporal consistency assessment without any additional training or task-specific fine-tuning, as illustrated in Fig. 2.

To rigorously benchmark temporal error detection and validate our method, we introduce the temporal error detection benchmark (TEDBench). TEDBench comprises 381 videos paired with 1524 captions containing controlled temporal errors, plus 3048 bidirectional question–answer (QA) pairs that probe fine-grained temporal reasoning. Videos are sampled from diverse public corpora something-something V2 (Goyal et al., 2017), Moments in Time (Monfort et al., 2020), and Charades (Sigurdsson et al., 2016). Captions are generated or perturbed by GPT-4o through rule-based transformations and then manually verified. Experiments with

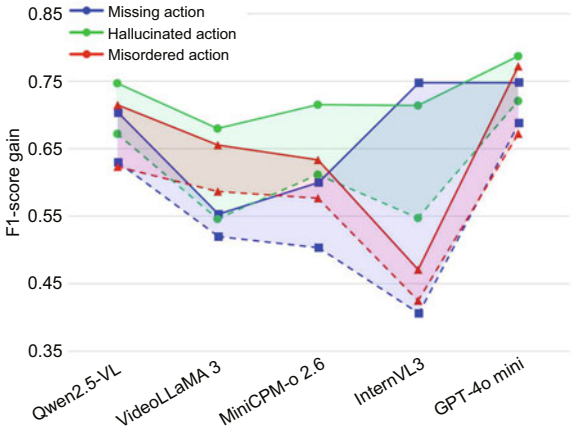


**Fig. 2 Performance with and without TimeJudge. Five video-LLMs show F1-score gains with TimeJudge (solid) over the baseline (dashed) on missing (blue), hallucinated (green), and misordered (red) errors in TEDBench. References to color refer to the online version of this figure**

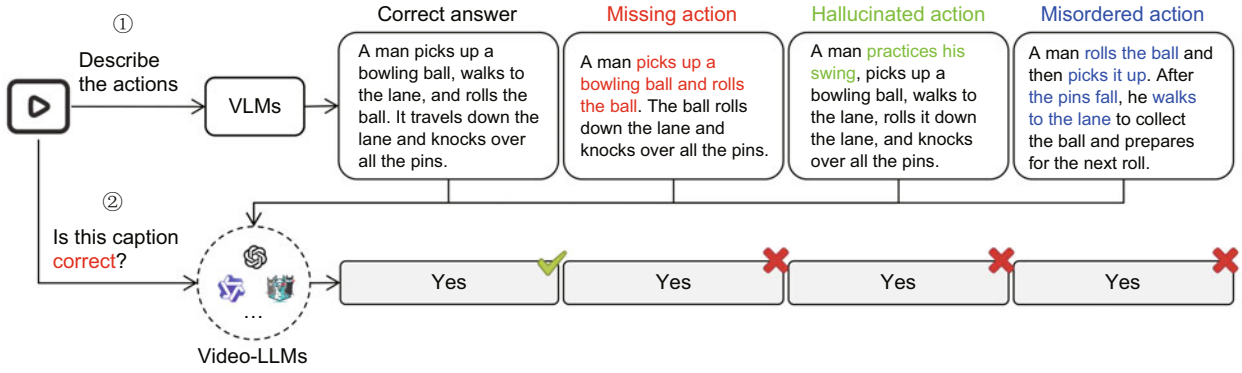


**Fig. 1 Limitations of the current video-LLMs in temporal error detection. Given a video and a VLM-generated caption, even advanced video-LLMs such as GPT-4o often fail to detect temporal errors (such as captions containing missing, hallucinated, or misordered actions) and incorrectly mark them as correct. This reflects their reliance on textual fluency or familiar visual cues rather than true temporal alignment with the video**

several recent video-LLMs show that in their default configurations, these models frequently miss temporal inconsistencies. After applying TimeJudge, however, they achieve substantially higher accuracy on TEDBench, underscoring the effectiveness and generality of our approach.

## 2 Related works

### 2.1 MLLM-as-a-Judge paradigm

With the growing comprehension and generation abilities of LLMs, automatic evaluation has become increasingly feasible. To align model judgments with human standards, researchers establish the data foundation by integrating existing or new resources, gathering high-quality human judgments (Wang BS et al., 2023; Deshpande et al., 2024; Vu et al., 2024), or generating synthetic data with LLMs to alleviate annotation efforts (Wang BJ et al., 2024; Wu et al., 2024). Building on this, methods such as supervised fine-tuning (Li JL et al., 2023; Wang YD et al., 2023; Xie et al., 2024), directed optimization (Rafailov et al., 2023; Park et al., 2024), and meta-rewarding (Wang TL et al., 2024; Wu et al., 2024) further improve LLM evaluation capabilities. Liu and Zhang (2025) presented the first systematic study on the "Video-LLM as a Judge" paradigm, revealing the unreliability of existing models and introducing agent–debate for stricter evaluation, albeit requiring reference answers. In contrast, TimeJudge targets temporal error detection by refining the evaluation pipeline and guiding model attention, without relying on reference answers or model fine-tuning.

### 2.2 MLLM-as-a-Judge benchmarking

Evaluating LLMs-as-a-Judge typically focuses on specific dimensions, such as the overall performance (Zheng et al., 2023; Wang YC et al., 2024), bias detection (Park et al., 2024; Shi et al., 2024), reasoning (Tan et al., 2024; Ye et al., 2024), and multilingual capabilities (Son et al., 2024). Table 1 summarizes recent multimodal judge benchmarks, most targeting image-based tasks. Chen et al. (2024) proposed MLLM-as-a-Judge, revealing that multimodal large language models (MLLMs) excel at pairwise judgment but struggle with scoring and ranking consistency. VL-RewardBench (Li L et al., 2024) covers multimodal queries, hallucination detection, and complex reasoning, highlighting deficiencies in basic visual perception. Wang ZT et al. (2025) introduced Objective Safety Bench, using diffusion models to generate rule-violating images for safety evaluation. Multimodal RewardBench (Yasunaga et al., 2025) challenges reward models with expert-annotated data in six domains, exposing reasoning and safety limitations in state-of-the-art video-LLMs. Pu et al. (2025) proposed comprehensive cross-modal benchmarks including video-to-text evaluation, exposing gaps between model and human judgments. In contrast, TEDBench is the first benchmark specifically focused on temporal error detection in video-LLMs, assessing models' ability to judge temporal alignment between video and caption and revealing weaknesses in handling temporal inconsistencies.

## 3 TimeJudge framework

In this section, we propose TimeJudge, a zero-shot framework to verify caption fidelity to video content and enhance video-LLMs' ability to detect temporal errors. After defining the task (Section 3.1), we introduce TimeJudge, which simplifies the judgment process through problem decomposition (QD) (Section 3.2), calibrates confidence based on modality contribution (Section 3.3), improves reliability with opposite question pairs, and aggregates judgments via weighted voting (Section 3.4).

**Table 1 Comparison of our TEDBench with existing multimodal judge benchmarks**

| Benchmark | Modality | Scale ($\times 10^3$) | Annotation | Task |
|---|---|---|---|---|
| MLLM-as-a-Judge | Image | 15.45 | Human | Human-model agreement |
| VL-RewardBench | Image | 1.54 | Human, LLMs | General QA, reasoning, hallucination |
| Objective Safety Bench | Image | 1.4 | Diffusion models | Image safety |
| Multimodal RewardBench | Image | 5.21 | Human, LLMs | General correctness, knowledge, etc. |
| JudgeAnything | Image, video, etc. | 9 | Human, LLMs | Understanding, generation |
| TEDBench (Ours) | Video | 3.05 | Human, LLMs | Temporal error detection |

### 3.1 Problem formulation

Given a video $v$, a caption $c$, and a question $q$, our goal is to perform structured judgment in two parts: (1) determine whether the caption $c$ satisfies the temporal condition implied by $q$; (2) identify which specific temporal constraints are violated, if any. Formally, we define the evaluation function as follows:

$$A(v, c, q) \rightarrow (s, R), \tag{1}$$

where $s$ indicates the correct label (either "correct" or "incorrect"), and $R$ denotes the specific temporal aspects violated by the inspected caption.

### 3.2 Question decomposition (QD)

Directly judging caption correctness is challenging for the current video-LLMs due to the complexity of temporal understanding. To reduce ambiguity and cognitive load, we decompose the task into a series of binary questions, each targeting a specific aspect of consistency. Each question $Q$ is paired with its inverse $Q'$ to enable cross-validation, reducing yes/no bias and enhancing judgment stability. As shown in Fig. 3, all questions are generated via GPT-4o using structured prompts to ensure clarity and controlled complexity, and guiding video-LLMs to focus on temporal cues and subtle inconsistencies in plausible captions. We then query the video-LLMs with these question pairs and aggregate their confidence scores to make the final decision, considering the consistency between answers within each pair. Detailed questions are provided in the supplementary materials.

### 3.3 Modality-sensitive calibration (MSC)

To mitigate language bias and visual neglect in video-LLMs, we propose a confidence calibration method based on modality contributions. As shown in Fig. 3, given a video–caption pair with the predicted probability $P \in [0, 1]$, we measure each modality's contribution by masking it and calculating the change in confidence:

$$\Delta c = P - P_{\text{video-only}}, \quad \Delta v = P - P_{\text{caption-only}}. \tag{2}$$

The combined contribution is computed as follows:

$$\Delta = \alpha \Delta v + (1 - \alpha)\Delta c, \tag{3}$$

where $\alpha \in [0, 1]$ balances the visual and textual importance. The calibrated confidence $P_{\text{cal}}$ is then

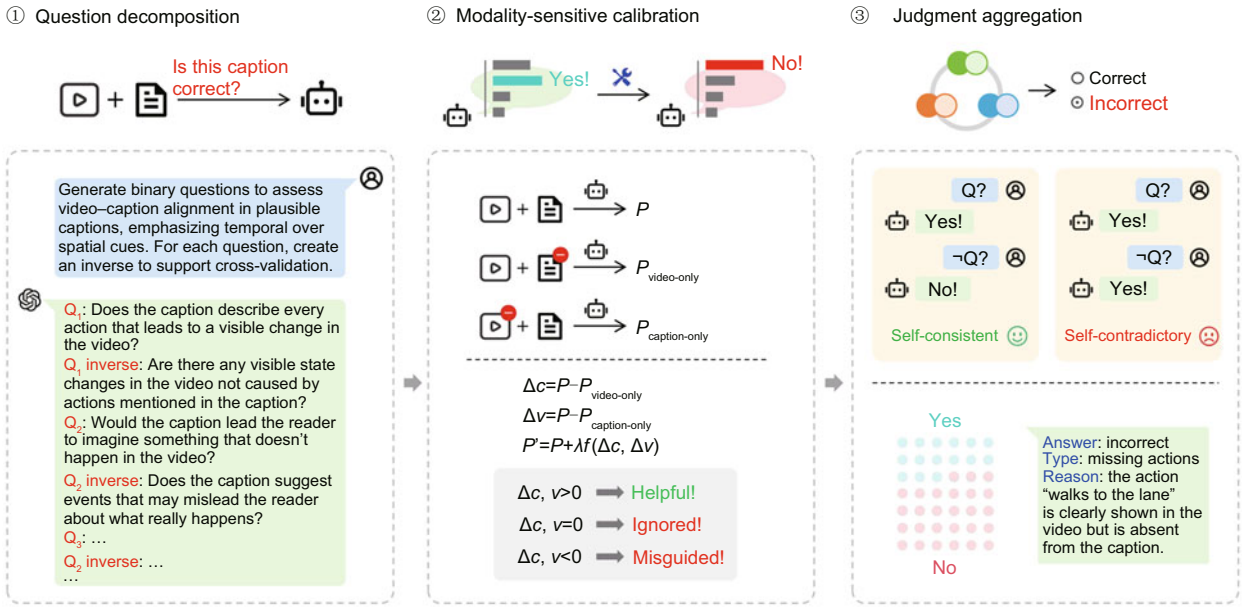


**Fig. 3 An overview of the TimeJudge. TimeJudge first decomposes the overall judgment into temporally focused binary questions using GPT-4o, pairing each question with its inverse for self-consistency checks. Then, it evaluates modality contributions by comparing model predictions under full, video-only, and caption-only inputs, calibrating confidence based on the differences $\Delta c$ and $\Delta v$. Finally, logical consistency across question pairs is enforced, and answers are aggregated via weighted voting to produce the final decision**

obtained by adding an offset in logit space:

$$P_{\text{cal}} = T^{-1}(T(P) + \lambda f(\Delta)), \tag{4}$$

where $T(P) = \ln\left(\frac{P}{1-P}\right)$ is the logit function, $T^{-1}$ is its inverse (i.e., sigmoid), $f$ = arctanh maps $\Delta$ to a logit space, and $\lambda$ indicates the calibration strength. This calibration method favors predictions that leverage both modalities positively, encouraging balanced multimodal reasoning.

### 3.4 Judgment aggregation (JA)

After obtaining modality-calibrated confidence scores for each question pair, we proceed to aggregate them into a final binary decision. For a question pair $(Q, Q')$ with calibrated probabilities $(P_{\text{yes}}, P_{\text{no}})$ and $(P'_{\text{yes}}, P'_{\text{no}})$, we define the "yes" support score and self-consistency weight as follows:

$$s^{\text{yes}} = \frac{1}{2}(P_{\text{yes}} + P'_{\text{no}}), \quad w^{\text{yes}} = 1 - |P_{\text{yes}} - P'_{\text{no}}|\,. \tag{5}$$

As shown in Fig. 3, $w^{\text{yes}}$ reflects the model's internal consistency, approaching 1 when the predictions for "yes" and its inverse "no" align. Similarly, we define the "no" support score and consistency weight as follows:

$$s^{\text{no}} = \frac{1}{2}(P_{\text{no}} + P'_{\text{yes}}), \quad w^{\text{no}} = 1 - |P_{\text{no}} - P'_{\text{yes}}|\,. \tag{6}$$

We then aggregate support over all $N$ question pairs $(Q_i, Q'_i)$ via weighted voting:

$$S_{\text{yes}} = \frac{\sum_{i=1}^{N} w_i^{\text{yes}} s_i^{\text{yes}}}{\sum_{i=1}^{N} w_i^{\text{yes}}}, \quad S_{\text{no}} = \frac{\sum_{i=1}^{N} w_i^{\text{no}} s_i^{\text{no}}}{\sum_{i=1}^{N} w_i^{\text{no}}}\,. \tag{7}$$

The final decision is "Yes" if $S_{\text{yes}} > S_{\text{no}}$, and "No" otherwise. This strategy encourages semantically consistent predictions while mitigating phrasing-induced variability, leading to more reliable decisions across diverse temporal reasoning scenarios.

## 4 TEDBench

It is broadly recognized that video-LLMs struggle with temporal understanding, yet no comprehensive benchmarks exist to thoroughly investigate this issue. To address this gap, we introduce TEDBench, a multidimensional benchmark specifically designed to evaluate and improve video-LLMs' ability to detect temporal errors.

### 4.1 Construction

#### 4.1.1 Data collection

We selected videos from various public datasets, including something-something V2, Moment in Time, and Charades, focusing on temporally sensitive content to test the model's understanding of temporal sequences rather than just static frames. The selected videos cover various scenes, actions, scene transitions, and time spans, offering the model significant diversity and challenges. To avoid information leakage, subtitles are removed to ensure that analysis relies solely on visual content.

#### 4.1.2 Action categorization

As shown in Fig. 4, we compiled 75 actions and categorized them into four levels based on their temporal and semantic complexity. Atomic actions (31%) are fine-grained movements such as bending or twisting, serving as basic units for complex actions and requiring temporal modeling. Compositional actions (25%) are short functional activities built from atomic motions, such as walking or lifting, often with clear goals. Interactive actions (24%) involve human–object interactions such as opening a drawer or pouring water, often requiring multimodal understanding. Complex behaviors (20%) involve multistep actions with context and intent, such as cooking or cleaning. Most videos in our dataset play for <6 s. These short clips focus on a few actions, aiding evaluation of the model's fine-grained temporal understanding.

#### 4.1.3 Data augmentation

As shown in Fig. 4, each original video is augmented with multiple captions: one correct caption and several captions containing temporal errors, including missing, hallucinated, and misordered actions. GPT-4o generates these captions as follows: missing and hallucinated errors are generated by modifying correct captions, and misordered errors are generated by shuffling video frames for more natural results. To ensure challenge and realism, temporal errors are designed to be reasonably consistent with the video content, since completely illogical errors are easily detected. Cross-verification ensured quality by having reviewers assess the accuracy and contextual relevance of each video–caption pair and

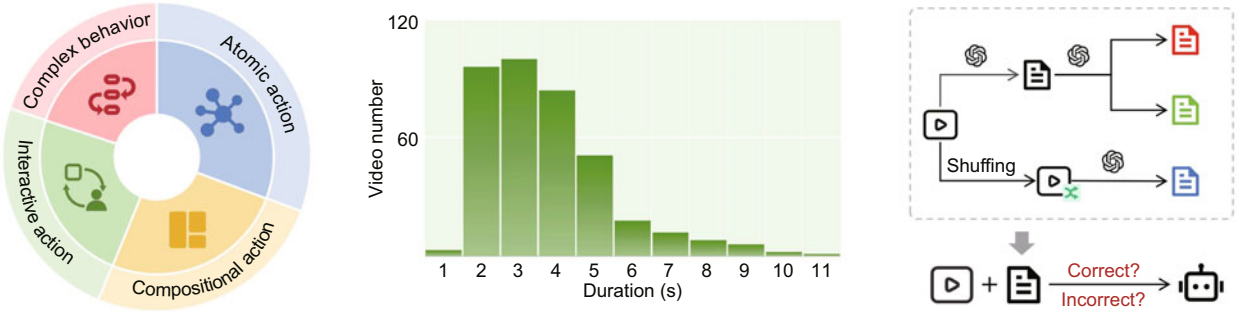


**Fig. 4 An overview of the TEDBench dataset. Left panel: videos are categorized into four levels: atomic action, compositional action, interactive action, and complex behavior, capturing a wide spectrum of temporal reasoning challenges. Middle panel: the video duration distribution shows that most clips play for <6 s, offering concise yet detail-rich content. Right panel: our pipeline generates both correct and diverse erroneous captions to create video–caption pairs for evaluating the judgment capabilities of video-LLMs**

make necessary corrections.

#### 4.1.4 Statistics

Using our data augmentation strategy, we created a temporal error detection dataset with 381 videos, each having multiple captions, including 381 correct and 1143 erroneous ones, along with error type annotations for detailed model evaluation. Based on this, we generated 3048 QA pairs to assess the model's ability to detect temporal errors in the captions. For each video–caption pair, we adopt a bidirectional consistency check, whereby a prediction is considered correct only if the model answers both the forward and the reverse questions correctly, enabling a stricter evaluation of its logical consistency and real-world reliability.

### 4.2 Evaluation protocol

In this evaluation, for each video–caption pair, the model answers two semantically equivalent but oppositely phrased binary questions, such as the following: "Does the caption accurately reflect the content of the video?" and "Is there any inconsistency between the caption and the video content?" The judgment is considered correct only if both answers are correct, reducing bias from one-sided responses. Models must respond strictly with "Yes" or "No," without ambiguous answers such as "Uncertain," and provide brief explanations to facilitate comparison with ground truth. To ensure logical consistency, GPT-4o jointly evaluates the question, model answer, and explanation, accepting only self-consistent responses. Performance is measured by accuracy, recall, and F1-score. We prioritize recall, as missing errors are more harmful downstream, while moderate false-positive results can be efficiently reviewed. Precision can be improved via post-processing, e.g., high-confidence filtering, cross-question consistency, or a conservative "uncertain/human-review" mode, but may reduce recall and complicate the pipeline.

## 5 Experiments

In this section, we present experiments to evaluate the effectiveness of TimeJudge and analyze the impact of each component and key hyperparameter and thereafter discuss its limitations.

### 5.1 Implementation details

We evaluated four of the most advanced open-source video-LLMs on TEDBench for temporal error detection: Qwen2.5-VL 7B (Bai S et al., 2025), VideoLLaMA 3 7B (Zhang et al., 2025), InternVL3 8B (Zhu et al., 2025), and MiniCPM-o 2.6 8B (Yu et al., 2025), each using different architectures and training strategies. GPT-4o mini (Hurst et al., 2024) was also included as a high-performance reference. Each model is tested under the default (base) and enhanced (TimeJudge, ours) settings. We report accuracy, recall, and F1-score, prioritizing high recall to avoid missed detections.

To demonstrate the robustness of TimeJudge, we adopt a unified configuration ($\alpha = 0.5$, $\lambda = 8.0$, $N = 10$) across all models, ensuring consistency in evaluation. However, these parameters remain adjustable to achieve each model's best performance.

We conducted experiments under each model's default settings to ensure fairness. All models process eight frames per video using temporal segment network (TSN) sampling (Wang LM et al., 2016), which divides a video into eight temporal segments and samples one frame from each segment to cover the entire duration. Greedy decoding was used as the default strategy across all open-source models. Experiments were run on identical hardware (4× NVIDIA A100 40 GB GPUs).

### 5.2 Comparisons

Table 2 summarizes the results across the three tasks. The base rows reveal that all models perform suboptimally, especially in terms of recall, indicating that when captions appear highly plausible (grammatically correct and contextually typical), models tend to accept them as correct. Models perform better on hallucinated actions, probably due to targeted optimization for such errors. After applying TimeJudge, all models consistently improve, demonstrating its effectiveness in enhancing temporal reasoning.

#### 5.2.1 Missing action

Most models exhibit weak baseline performance (recall <60%), reflecting insensitivity to omitted actions in captions. Applying TimeJudge boosted recall by about 30 percentage points (PPs) for VideoLLaMA 3 and 25 PPs for InternVL3, although F1-score gains were limited due to overdetection. In contrast, Qwen2.5-VL and GPT-4o mini showed steady improvements across all metrics, especially in terms of F1-score, indicating a more balanced judgment. MiniCPM-o 2.6 showed modest recall gains but notable F1-score improvement, indicating better precision. Detecting missing actions requires fine-grained video–text alignment, which our MSC method effectively strengthens.

#### 5.2.2 Hallucinated action

Baseline results are relatively strong but still leave room for improvement. MiniCPM-o 2.6, VideoLLaMA 3, and InternVL3 improved the F1-score by $>$ 10%, showing stronger resistance to hallucinated captions. GPT-4o mini achieved the highest F1-score, suggesting a more balanced decision boundary. Detecting hallucinations is challenging because it relies on "negative evidence," recognizing that something did not occur in the video. Our method enhances access to such evidence by structuring judgment through QD and inverse verification.

**Table 2 Performance on the TEDBench**

| Video-LLMs | $N_{\text{par}}$ | Method | Missing action | | | Hallucinated action | | | Misordered action | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Accuracy (%) | Recall (%) | F1 | Accuracy (%) | Recall (%) | F1 | Accuracy (%) | Recall (%) | F1 |
| Random | | | 25.00 | 25.00 | 0.2500 | 25.00 | 25.00 | 0.2500 | 25.00 | 25.00 | 0.2500 |
| Qwen2.5-VL | 7B | Base | 50.26 | 50.16 | 0.6295 | 54.59 | 52.59 | 0.6723 | 50.13 | 50.08 | 0.6282 |
| | | Ours | **61.29** | **57.03** | **0.7029** | **68.77** | **62.84** | **0.7463** | **72.05** | **72.34** | **0.7186** |
| VideoLLaMA 3 | 7B | Base | 53.41 | 53.63 | 0.5196 | 59.38 | 61.76 | 0.5478 | 58.40 | 58.08 | 0.5920 |
| | | Ours | **66.67** | **83.96** | **0.5528** | **66.67** | **65.38** | **0.6801** | **69.16** | **73.55** | **0.6599** |
| InternVL3 | 8B | Base | 47.11 | 47.44 | 0.5031 | 55.51 | 54.25 | 0.6126 | 56.25 | 55.71 | 0.5821 |
| | | Ours | **66.01** | **72.93** | **0.5997** | **75.20** | **84.04** | **0.7149** | **71.13** | **85.46** | **0.6382** |
| MiniCPM-o 2.6 | 8B | Base | 54.16 | 62.00 | 0.4059 | 62.73 | 69.48 | 0.5492 | 56.56 | 62.38 | 0.4322 |
| | | Ours | **68.50** | **62.39** | **0.7474** | **72.83** | **75.44** | **0.7137** | **61.30** | **75.57** | **0.4776** |
| GPT-4o mini | | Base | 61.29 | 57.62 | 0.6878 | 65.88 | 61.02 | 0.7204 | 63.12 | 60.25 | 0.6766 |
| | | Ours | **68.50** | **62.35** | **0.7479** | **74.67** | **67.81** | **0.7859** | **75.72** | **72.17** | **0.7752** |

$N_{\text{par}}$: number of parameters. Better results are in bold. Base: results with the original model predictions. Ours: results after applying TimeJudge. All models consistently improve across three temporal error types after applying TimeJudge, demonstrating the latter's general effectiveness. Notably, Qwen2.5-VL and GPT-4o mini achieved comprehensive gains across all metrics, while other models showed significant gains in terms of recall

#### 5.2.3 Misordered action

Models struggle with this task due to limited temporal modeling in the current video-LLMs. Our method significantly improved the performance, with the F1-scores of Qwen2.5-VL and GPT-4o mini increasing by approximately 10%, reflecting enhanced sensitivity to action sequences. VideoLLaMA 3 and InternVL3 showed significant recall gains but still tend toward overdetection. MiniCPM-o 2.6's low F1-score despite high recall reflects a high false-positive rate. This task demands capturing detailed action order and rhythm; our method strengthens temporal judgment via rigorous reasoning and consistency constraints.

### 5.3 Effectiveness of different components

We conducted ablation studies to evaluate the contribution of each component in our proposed framework. Starting from the base video-LLMs, we progressively incorporated QD, MSC, and JA. Taking Qwen2.5-VL as an example, Table 3 summarizes the results on TEDBench. Adding QD improves the accuracy and recall by breaking complex judgments into simpler binary questions, with larger gains on hallucinated and misordered actions. MSC further boosts the performance across all three tasks by calibrating confidence, reducing modality bias, and promoting balanced multimodal reasoning. Building on QD and MSC, JA provides additional gains by integrating results from all question pairs through self-consistency checks and weighted voting. These components complement each other, and our full framework consistently surpasses the baseline and partial variants, confirming each module's essential role in enhancing video-LLMs' temporal reasoning and judgment.

### 5.4 Hyperparameter ablation studies

We conducted ablation studies on three key hyperparameters: modality weight $\alpha$, calibration strength $\lambda$, and the number of QA pairs $N$. Using the F1-score of Qwen2.5-VL as an example, we varied one parameter at a time while keeping others fixed to isolate their effects. As shown in Fig. 5 (left panel), performance peaks at a balanced $\alpha$ (around 0.5), highlighting the importance of jointly leveraging both video and caption modalities. In contrast, extreme values ($\alpha = 0$ or $\alpha = 1$) lead to significant performance degradation, indicating that relying solely on either modality is suboptimal. Fig. 5 (middle panel) demonstrates the effectiveness of calibration: introducing a nonzero $\lambda$ consistently improves the performance compared to the experiment without calibration ($\lambda = 0$). The best results are achieved at $\lambda = 8$, while further increases cause the gains to plateau or slightly decline, probably due to the overconfidence effects. Fig. 5 (right panel) shows that the performance improves with more QA pairs $N$, plateauing near $N = 10$. However, the missing action task benefits beyond this point, reflecting its complexity and the need for more questions to cover diverse scenarios. However, a larger $N$ also increases the computational cost, as TimeJudge requires $3 \times 2N$ model calls ($N$ QA pairs, three passes for calibration), highlighting a trade-off between performance and efficiency. These ablation results validate our design and confirm the method's robustness under reasonable parameter changes. We also provide additional baseline comparisons in the supplementary materials.

### 5.5 Challenges and limitations

While TimeJudge generally performs robustly in detecting temporal errors, certain extreme

**Table 3 Impact of each component of the TimeJudge**

| Method | Missing action | | | Hallucinated action | | | Misordered action | | |
|---|---|---|---|---|---|---|---|---|---|
| | Accuracy (%) | Recall (%) | F1 | Accuracy (%) | Recall (%) | F1 | Accuracy (%) | Recall (%) | F1 |
| Random | 25.00 | 25.00 | 0.2500 | 25.00 | 25.00 | 0.2500 | 25.00 | 25.00 | 0.2500 |
| Base | 50.26 | 50.16 | 0.6295 | 54.59 | 52.59 | 0.6723 | 50.13 | 50.08 | 0.6282 |
| QD | 51.32 | 50.70 | 0.6599 | 62.34 | 57.56 | 0.7139 | 66.14 | 64.82 | 0.6759 |
| QD+MSC | 55.25 | 52.84 | 0.6857 | 67.06 | 61.32 | 0.7372 | 71.26 | 71.89 | 0.7084 |
| QD+MSC+JA (TimeJudge) | 61.29 | 57.03 | 0.7029 | 68.77 | 62.84 | 0.7463 | 72.05 | 72.34 | 0.7186 |

Progressive addition of QD, MSC, and JA steadily improves all metrics across temporal error types, with the full framework achieving the best overall performance

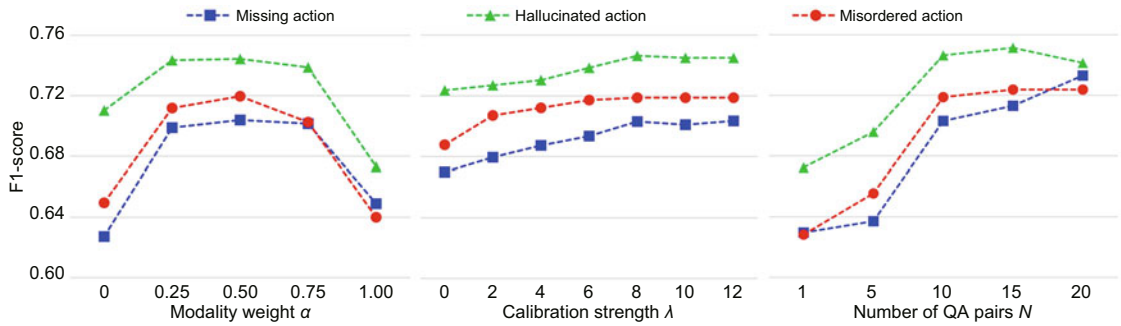


**Fig. 5 Ablation on key hyperparameters using F1-score on the TEDBench. Left panel: modality weight $\alpha$ peaks near 0.5, highlighting the need to combine video and caption. Middle panel: calibration strength $\lambda$ performs best at 8. Right panel: F1-score improves with the increasing number of QA pairs $N$ but plateaus after 10, except for the missing action task, which is more complex and requires more questions to cover diverse scenarios**

scenarios reveal its limitations. This section summarizes key factors affecting its performance.

#### 5.5.1 Incomplete QD

Even with state-of-the-art video-LLMs such as GPT-4o, subtle temporal errors may be missed. Future work could use finer-grained subquestions or per-video–caption-pair customization, albeit at substantially increased computational cost.

#### 5.5.2 Long videos and fine-grained temporal dynamics

Evaluating long videos is computationally expensive. In videos with high temporal resolution or subtle actions, TimeJudge may miss fine-grained temporal dynamics, leading to undetected errors. Multiscale temporal encoding can help reduce cost while preserving key temporal information.

#### 5.5.3 External factors

Factors such as video-LLM hallucinations, modality biases, rare or complex actions, and adversarial video–caption pairs can negatively affect TimeJudge's performance, emphasizing the need for improved model architectures and more comprehensive video-LLMs.

## 6 Conclusions

Using video-LLMs to replace human evaluators is becoming mainstream, but these models still struggle with detecting temporal inconsistencies in captions. In this work, we present TimeJudge, a novel zero-shot framework for detecting temporal errors in video captions. By decomposing judgments into simpler binary questions, calibrating confidence based on modality contributions, and aggregating results via consistency-weighted voting, TimeJudge significantly improves temporal error detection across multiple video-LLMs without requiring fine-tuning. Additionally, we introduce TEDBench, the first benchmark specifically designed for this task, featuring diverse video types and various caption errors, laying a foundation for more robust and reliable evaluation of vision-language models. In the future, we plan to expand our dataset with larger videos and more tasks to increase its impact.

## Acknowledgments

The computations in this work were supported by the High-Performance Computing (HPC) Platform of Huazhong University of Science and Technology.

## Contributors

Yangliu HU conducted the research, performed the experiments, analyzed the data, and drafted the paper. Zikai SONG and Wei YANG supervised the work and revised the paper. All the authors read and approved the final version of the paper.

## Conflict of interest

All the authors declare that they have no conflict of interest.

## Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## References


Bai S, Chen K, Liu X, et al., 2025. Qwen2.5-VL technical report. https://doi.org/10.48550/arXiv.2502.13923

Bai YS, Ying JH, Cao YX, et al., 2023. Benchmarking foundation models with Language-Model-as-an-Examiner. https://doi.org/10.48550/arXiv.2306.04181

Chen DP, Chen RX, Zhang SL, et al., 2024. MLLM-as-a-Judge: assessing multimodal LLM-as-a-Judge with vision-language benchmark. https://doi.org/10.48550/arXiv.2402.04788

Deshpande D, Ravi SS, CH-Wang S, et al., 2024. GLIDER: grading LLM interactions and decisions using explainable ranking. https://doi.org/10.48550/arXiv.2412.14140

Goyal R, Kahou SE, Michalski V, et al., 2017. The "something something" video database for learning and evaluating visual common sense. Proc IEEE Int Conf on Computer Vision, p.5842-5850. https://doi.org/10.1109/ICCV.2017.622

Hurst A, Lerer A, Goucher AP, et al., 2024. GPT-4o system card. https://doi.org/10.48550/arXiv.2410.21276

Lee H, Phatale S, Mansoor H, et al., 2023. RLAIF: scaling reinforcement learning from human feedback with AI feedback. Proc 41st Int Conf on Machine Learning.

Li JL, Sun SC, Yuan WZ, et al., 2023. Generative judge for evaluating alignment. https://doi.org/10.48550/arXiv.2310.05470

Li L, Wei YC, Xie ZH, et al., 2024. VL-RewardBench: a challenging benchmark for vision-language generative reward models. https://doi.org/10.48550/arXiv.2411.17451

Li RS, Patel T, Du XY, 2023. PRD: peer rank and discussion improve large language model based evaluations. https://doi.org/10.48550/arXiv.2307.02762

Liang T, He ZW, Jiao WX, et al., 2024. Encouraging divergent thinking in large language models through multi-agent debate. Proc Conf on Empirical Methods in Natural Language Processing, p.17889-17904. https://doi.org/10.18653/v1/2024.emnlp-main.992

Liao RT, Erler M, Wang HY, et al., 2024. VideoINSTA: zero-shot long video understanding via informative spatial-temporal reasoning with LLMs. Proc Findings of the Association for Computational Linguistics, p.6577-6602. https://doi.org/10.18653/v1/2024.findings-emnlp.384

Liu M, Zhang WS, 2025. Is your video language model a reliable judge? https://doi.org/10.48550/arXiv.2503.05977

Monfort M, Andonian A, Zhou BL, et al., 2020. Moments in Time dataset: one million videos for event understanding. *IEEE Trans Pattern Anal Mach Intell*, 42(2):502-508. https://doi.org/10.1109/TPAMI.2019.2901464

Park J, Jwa S, Ren MY, et al., 2024. OffsetBias: leveraging debiased data for tuning evaluators. Proc Findings of the Association for Computational Linguistics, p.1043-1067. https://doi.org/10.18653/v1/2024.findings-emnlp.57

Pu S, Wang YC, Chen DP, et al., 2025. Judge anything: MLLM as a judge across any modality. https://doi.org/10.48550/arXiv.2503.17489

Rafailov R, Sharma A, Mitchell E, et al., 2023. Direct preference optimization: your language model is secretly a reward model. https://doi.org/10.48550/arXiv.2305.18290

Shi JW, Yuan ZH, Liu YN, et al., 2024. Optimization-based prompt injection attack to LLM-as-a-Judge. Proc ACM SIGSAC Conf on Computer and Communications Security, p.660-674. https://doi.org/10.1145/3658644.3690291

Sigurdsson GA, Varol G, Wang XL, et al., 2016. Hollywood in Homes: crowdsourcing data collection for activity understanding. Proc 14th European Conf on Computer Vision, p.510-526. https://doi.org/10.1007/978-3-319-46448-0_31

Son G, Yoon D, Suk J, et al., 2024. MM-Eval: a multilingual meta-evaluation benchmark for LLM-as-a-Judge and reward models. https://doi.org/10.48550/arXiv.2410.17578

Tan SJ, Zhuang SY, Montgomery K, et al., 2024. JudgeBench: a benchmark for evaluating LLM-based judges. https://doi.org/10.48550/arXiv.2410.12784

Vu T, Krishna K, Alzubi S, et al., 2024. Foundational autoraters: taming large language models for better automatic evaluation. Proc Conf on Empirical Methods in Natural Language Processing, p.17086-17105. https://doi.org/10.18653/v1/2024.emnlp-main.949

Wang BJ, Chern S, Chern E, et al., 2024. Halu-J: critique-based hallucination judge. https://doi.org/10.48550/arXiv.2407.12943

Wang BS, Yue X, Sun H, 2023. Can ChatGPT defend its belief in truth? Evaluating LLM reasoning via debate. Proc Findings of the Association for Computational Linguistics, p.11865-11881. https://doi.org/10.18653/v1/2023.findings-emnlp.795

Wang LM, Xiong YJ, Wang Z, et al., 2016. Temporal segment networks: towards good practices for deep action recognition. Proc 14th European Conf on Computer Vision, p.20-36. https://doi.org/10.1007/978-3-319-46484-8_2

Wang TL, Kulikov I, Golovneva O, et al., 2024. Self-taught evaluators. https://doi.org/10.48550/arXiv.2408.02666

Wang YC, Yuan JY, Chuang YN, et al., 2024. DHP benchmark: are LLMs good NLG evaluators? Proc Findings of the Association for Computational Linguistics, p.8079-8094. https://doi.org/10.18653/v1/2025.findings-naacl.451

Wang YD, Yu ZH, Zeng ZR, et al., 2023. PandaLM: an automatic evaluation benchmark for LLM instruction tuning optimization. https://doi.org/10.48550/arXiv.2306.05087

Wang ZT, Hu SM, Zhao SY, et al., 2025. MLLM-as-a-Judge for image safety without human labeling. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.14657-14666. https://doi.org/10.1109/CVPR52734.2025.01366

Wu TH, Yuan WZ, Golovneva O, et al., 2024. Meta-rewarding language models: self-improving alignment with LLM-as-a-Meta-Judge. https://doi.org/10.48550/arXiv.2407.19594

Xie TH, Qi XY, Zeng Y, et al., 2024. SORRY-bench: systematically evaluating large language model safety refusal behaviors. https://doi.org/10.48550/arXiv.2406.14598

Xu YF, Sun YZ, Xie ZE, et al., 2024. VTG-GPT: tuning-free zero-shot video temporal grounding with GPT. *Appl Sci*, 14(5):1894. https://doi.org/10.3390/app14051894

Yasunaga M, Zettlemoyer L, Ghazvininejad M, 2025. Multimodal RewardBench: holistic evaluation of reward models for vision language models. https://doi.org/10.48550/arXiv.2502.14191

Ye JY, Wang YB, Huang Y, et al., 2024. Justice or prejudice? Quantifying biases in LLM-as-a-Judge. https://doi.org/10.48550/arXiv.2410.02736

Yu TY, Zhang HY, Li QM, et al., 2025. RLAIF-V: open-source AI feedback leads to super GPT-4V trustworthiness. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.19985-19995. https://doi.org/10.1109/CVPR52734.2025.01861

Zhang BQ, Li KH, Cheng ZS, et al., 2025. VideoLLaMA 3: frontier multimodal foundation models for image and video understanding. https://doi.org/10.48550/arXiv.2501.13106

Zheng LM, Chiang WL, Sheng Y, et al., 2023. Judging LLM-as-a-Judge with MT-bench and Chatbot Arena. Proc 37$^{th}$ Int Conf on Neural Information Processing Systems, Article 2020.

Zhu JG, Wang WY, Chen Z, et al., 2025. InternVL3: exploring advanced training and test-time recipes for open-source multimodal models. https://doi.org/10.48550/arXiv.2504.10479

## List of supplementary materials

1 Question decomposition

2 Baseline comparisons

Table S1 Comparison of baseline methods on Qwen2.5-VL