VideoCritic: Diagnosing and Localizing Reasoning Errors in Video-Language Models

Chenwei Xu[1*] Jianshu Zhang[1] Shang Wu[1] Lie Lu[2]
Pranav Maneriker[2] Fan Du[2] Manling Li[1] Han Liu[1]
[1]Northwestern University [2]Dolby Laboratories

Abstract

Vision-language models (VLMs) have made remarkable progress in video reasoning tasks. However, they still frequently produce inaccurate reasoning chains, such as hallucinating nonexistent objects, misreading perceptual details, or confusing spatial and temporal relations. To address these challenges, we introduce VIDEOCRITIC-BENCH*, a benchmark that targets fine-grained reasoning errors in video-language understanding, and* VIDEOCRITIC-3B*, a 3B-parameter critic model that detects and categorizes reasoning errors.* VIDEOCRITIC-BENCH *contains two complementary splits: (1) Synthetic, constructed by injecting controlled reasoning errors into ground-truth chains; and (2) Realistic, a human-verified collection of authentic reasoning errors mined from both small and large VLMs. Together, these splits support systematic training and realistic evaluation of video reasoning robustness. We further develop* VIDEOCRITIC-3B*, a lightweight critic model for structured detection of common reasoning failures. Across experiments, it achieves strong performance on hallucination and perceptual errors. We also characterize the remaining challenges on logical and spatio-temporal grounding via stage-wise ablations.*

1. Introduction

Recent progress in Vision–Language Models (VLMs) has significantly advanced their capabilities in video understanding and multimodal reasoning. Modern VLMs can integrate information across frames, perform temporal grounding, and generate multi-step explanations for complex video queries. However, despite these advances, systematic reasoning failures remain common. Empirical studies show that VLMs frequently hallucinate entities or events not present in the video, misrecognize perceptual details such as object counts or attributes, and incorrectly resolve spatial or temporal relations between actions and actors [9, 12, 15]. These issues indicate not only perception mistakes but also deeper logical inconsistencies within the generated reasoning chains, for example, contradictions between earlier and later steps, causal inferences unsupported by visual evidence, or misaligned event ordering. Such failures are especially problematic in video reasoning because errors propagate across multiple steps: a single hallucinated detail can mislead subsequent inference, and misgrounded references can distort the entire reasoning trajectory. Consequently, even when final answers appear plausible, the underlying reasoning often remains unfaithful to the video content. These challenges fundamentally limit the trustworthiness, safety, and interpretability of current VLM-based video systems, highlighting the need for fine-grained diagnostic tools and principled mechanisms to detect, categorize, and ultimately mitigate reasoning errors in multimodal models.

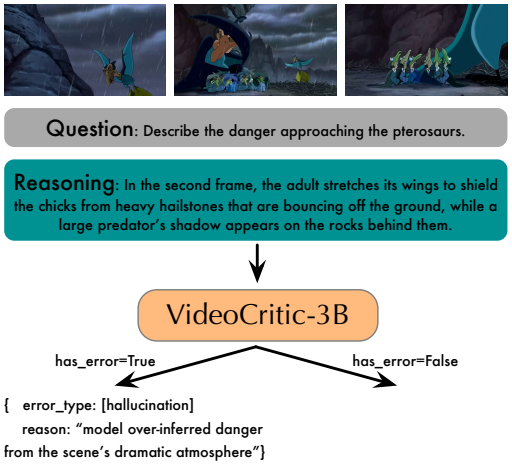

Figure 1. **VIDEOCRITIC-3B.** VIDEOCRITIC-3B takes the video frames, question, and reasoning chain as input. It first checks whether the chain contains any errors; if so, it outputs the detected error types along with a brief justification.

To address these issues, recent work has focused on improving VLM robustness and reasoning quality. Reinforcement fine-tuning has been used to enhance spatio-temporal perception in conversational video models [13], yielding

*Corresponding author. cxu@u.northwestern.edu

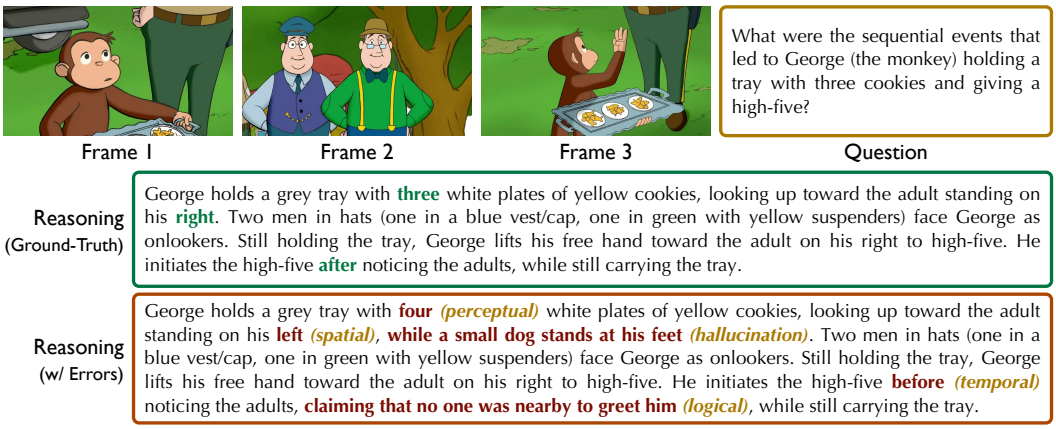

Figure 2. **Illustration of the Five Reasoning Error Types**. We visualize the five types of reasoning errors observed in video-language understanding. The top (green) box presents the correct, temporally coherent reasoning chain, while the bottom (orange) box injects one instance of every error type: perceptual, spatial, hallucination, temporal, and logical errors. Perceptual errors distort visual counts or attributes; spatial errors mislocate referenced entities; hallucination introduces unsupported content; temporal errors disrupt event order; and logical errors contradict previously stated or visually evident facts.

improvements in temporal grounding and object tracking. Other efforts, such as Fact-R1 [33], advance factual consistency and explainability through multimodal reasoning for verifiable video misinformation detection, while Ego-R1 [24] extends coherent reasoning to ultra-long egocentric videos via a Chain-of-Tools process guided by reinforcement learning. In parallel, Video-RTS [28] improves efficiency through data-efficient reinforcement learning and adaptive test-time scaling, achieving strong performance under significantly reduced supervision. Despite these advances, existing approaches remain highly specialized, each targeting a narrow aspect of the broader reasoning problem. As a result, reasoning failures—from factual hallucinations to temporal misalignment—continue to appear even in state-of-the-art systems. This motivates the need for a unified framework capable of systematically diagnosing fine-grained reasoning errors across diverse VLMs.

Evaluation benchmarks are essential for revealing these weaknesses. Conventional video-QA benchmarks [8, 16, 26, 29] largely score only final answers, obscuring whether models rely on genuine spatio-temporal reasoning or shortcut correlations. Recent benchmarks move toward interpretable, diagnostic assessment: probing temporal grounding [15], spatio-temporal reasoning [6], chain-of-thought reasoning [20], and expert-level comprehension [37], while others quantify hallucination in video LMMs [12]. Together, these works underscore that assessing how models reason and categorizing reasoning errors is as important as answer accuracy for robust video reasoning model development.

In this work, we present VIDEOCRITIC, a framework for systematically diagnosing and categorizing reasoning errors in video–language models (VLMs). Central to our approach is VIDEOCRITIC-BENCH, a benchmark with complementary synthetic and realistic splits, providing fine-grained annotations under a unified five-category taxonomy. Complementing the benchmark, we develop VIDEOCRITIC-3B, a compact critic that takes video frames, a question, and a candidate reasoning chain as input, and returns a structured JSON verdict indicating whether the chain contains errors, which error types are present, and a brief justification for analysis and auditing.

Another key component of our work is leveraging critic models to improve factual accuracy and reasoning robustness in VLMs. Recent studies show that incorporating a learned "critic" can catch mistakes that the primary model overlooks. For example, adding a Critic-V [32] module can enhance the reliability and accuracy of a vision-language model's answers. Inspired by this, we develop a dedicated critique model, VIDEOCRITIC-3B, to automatically analyze and validate the reasoning steps of a base VLM, as in Figure 1. In essence, the critic serves as an automated auditor that flags hallucinated details, contradictions, and misgrounded references in a given reasoning trace, producing structured diagnostics (has_error, error types, and a brief note) for fine-grained analysis. The benchmark and critic model are co-designed: the taxonomy directly shapes the critic's multi-head outputs, enabling consistent training, calibrated classification, and chain-level interpretability.

To summarize, we define our contributions as follows: (i) We introduce VIDEOCRITIC-BENCH, a first evaluation benchmark with complementary synthetic and realistic splits, providing annotated test cases that cover diverse reasoning errors; (ii) We define a clear five-category taxonomy of reasoning errors for video reasoning tasks; (iii) We propose VIDEOCRITIC-3B, a lightweight critic model that detects and categorizes errors in a primary model's reasoning trace; and (iv) Through comprehensive experiments, we show that our benchmark and critic model uncover hidden failure patterns in state-of-the-art VLMs, highlighting strategies to improve robustness and transparency of video reasoning.

2. VIDEOCRITIC-BENCH

We introduce VIDEOCRITIC-BENCH, a benchmark for diagnosing video reasoning errors, and VIDEOCRITIC-3B, a compact critic model for detecting such errors. Together, they form an end-to-end suite for diagnosing and evaluating video reasoning errors. VIDEOCRITIC-BENCH formalizes five error categories: logical, hallucination, perceptual, temporal grounding, and spatial grounding. These categories define the evaluation taxonomy and scoring axes. The benchmark comprises a synthetic split and a realistic split. The synthetic split comprises reasoning chains with errors injected by rule-based and LLM-assisted procedures. The realistic split comprises reasoning chains generated by real VLMs, including both smaller and larger models. Using the VIDEOCRITIC-BENCH training set aligned with this taxonomy, we train VIDEOCRITIC-3B to identify errors in VLM reasoning chains.

2.1. VIDEOCRITIC-BENCH

VIDEOCRITIC-BENCH includes two parts: synthetic and realistic. We introduce the error injection in Figure 3. For the synthetic part, we inject errors into reasoning chains using two mechanisms: rule-based and LLM-assisted. We manually instantiate the five error categories in our taxonomy. For the realistic part, we extract reasoning chains from both smaller (3–4B) and larger (72B & proprietary) models. Because we cannot control these models' reasoning to conform to our taxonomy, we instead detect and annotate occurrences of the five error categories in their outputs. To prevent any form of train–test leakage, all synthetic chains are generated exclusively from VideoEspresso [10] ground-truth reasoning, and no VLM-generated chains from the Realistic split are used during training.

Reasoning Error Taxonomy. We introduce a five-category taxonomy, as shown in Figure 2, that captures the major errors of video language model reasoning and disentangles ungrounded generation from misperception and misgrounding. Logical errors denote invalid inference steps (e.g., contradictions, unsupported causal links, arithmetic/magnitude mistakes) that are inconsistent with the video evidence or with earlier steps in the chain of thought. Hallucination refers to content asserted without support in the video (including extrinsic additions or claims that contradict what is shown), independent of whether other steps are logically sound. Perceptual errors arise from misrecognition at the visual level, incorrect objects, actions, attributes, or counts, such that the reasoning may be coherent, but is grounded in a faulty read of the frames. Temporal grounding errors indicate misalignment in time, such as incorrect timestamps, order, or duration (e.g., attributing an event to an earlier segment or reversing the before/after order). Spatial grounding errors indicate mislocalization in space, such as incorrect regions/objects, left–right confusion, or failure to point to the spatial evidence that supports an answer. We treat these categories as complementary: hallucination captures the absence of evidence; perceptual captures incorrect evidence; and spatial/temporal grounding captures the misplacement of otherwise relevant evidence, while logical errors assess the validity of the reasoning over whatever evidence. Introducing five types of error instead of only hallucination avoids the dominant video-specific failures: misgrounded temporal or spatial references and valid-sounding but logically flawed chains, which can occur even when all entities are visible. Using five categories (perceptual, spatial, temporal, hallucination, logical) directly mirrors the perceptual grounding and reasoning pipeline and reduces confounds that plagued prior single-type benchmarks.

Rule-based Error Injection. We design a fully rule-based pipeline to generate controlled reasoning errors aligned with our five-category taxonomy. Given a clean VideoEspresso-style QA record, the system parses the evidence and captions to extract candidate objects, actions, and temporal evidence. It then samples error types from a fixed distribution and instantiates them through deterministic templates: misrecognitions for perceptual, spatial or temporal swaps for grounding, unsupported claims for hallucination, and invalid inferences for logical errors. Each injected sentence is concise, grammatically valid, and automatically verified to match its intended category. The validated statements are appended to the original evidence, forming reasoning chains with precisely localized, labeled errors. This rule-based process enables scalable, reproducible generation of fine-grained synthetic data for evaluation and model training. In the current benchmark release, the rule-based subset in the test split contains no temporal or spatial grounding cases (Table 1), and spatio-temporal evaluation is therefore dominated by the human-verified realistic split.

LLM-Assisted Error Injection. We augment the rule-based injector with a controlled LLM path to synthesize more natural logical and hallucination statements. The LLM reuses the same sampled plan and context, keeping the error distribution and sampling unchanged. It realizes only the surface sentence; perceptual and grounding errors remain template-driven. Each LLM sentence is constrained by a typed prompt and then validated against our taxonomy and lexical rules. Invalid outputs fall back to deterministic templates, preserving completeness and reproducibility. All mutations are appended to the evidence and tagged with their generation source, isolating the benefits of LLM phrasing without sacrificing control.

Realistic Reasoning Errors. For each question–answer (QA) instance, we generate multiple candidate reasoning chains from small VLMs and, in parallel, from a high-capacity VLM; all candidates are processed in a unified scoring–judging pipeline. For a chain c and cue e_i , let $E(c) = \{e_i\}$ denote the set of detected cues obtained from

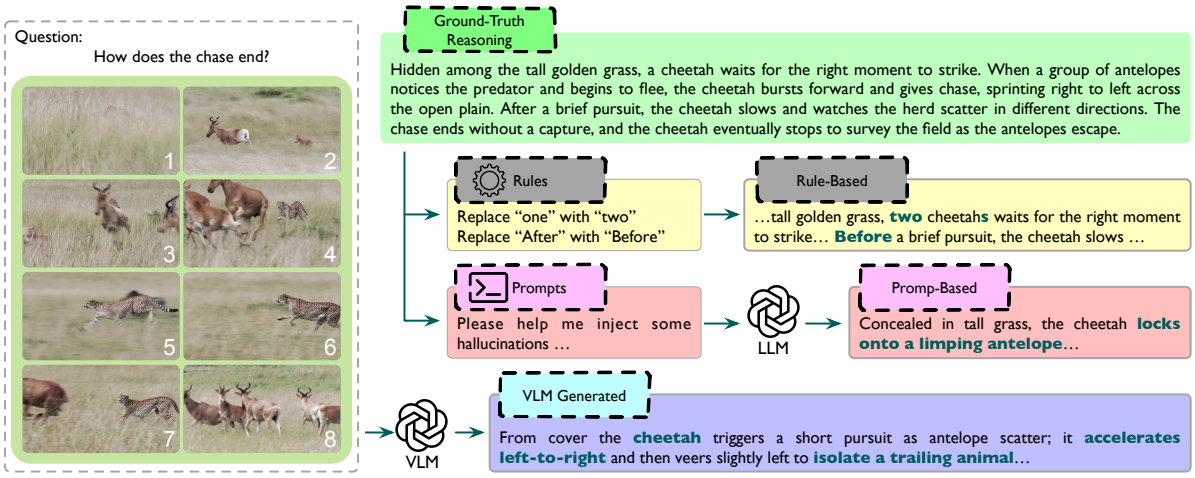

Figure 3. **Error-injection Methods**. Given a video clip and a question (left), we start from a ground-truth chain (top). We then create synthetic corrupted chains via two mechanisms: (a) rule-based injection, which applies atomic string edits that instantiate our taxonomy with minimal changes (e.g., swapping 'one'→'two' for perceptual miscount, or 'after'→'before' for temporal inversion); and (b) LLM-assisted injection, which asks an external LLM to append short naturalistic corruptions under typed constraints. We use rule-based and LLM-assisted injections to construct the Synthetic split for training and testing. In contrast, the Realistic split is mined from raw VLM-generated chains and human-verified, and is used only for evaluation to avoid train–test leakage.

lightweight validators (covering the five taxonomy types in VIDEOCRITIC-BENCH, as well as auxiliary signals such as answer-mismatch and self-inconsistency). Each has a type $\tau(e_i)$, a calibrated confidence $\mathrm{conf}(e_i) \in [0,1]$, and a nonnegative class weight $w_{\tau(e_i)}$. To capture inter-generator agreement, we define the consensus-adjusted confidence

$$\widetilde{\mathrm{conf}}(e_i) = \mathrm{conf}(e_i) + \beta\, \mathbf{1}_{\mathrm{gen}},$$

where $\mathbf{1}_{\mathrm{gen}} = 1$ if the same cue is detected by another generator and 0 otherwise, and $\beta \geq 0$ is a consensus boost. The chain's suspicion score is

$$s(c) = \sum_{e_i \in E(c)} w_{\tau(e_i)}\, \widetilde{\mathrm{conf}}(e_i).$$

Let $T(c)$ be the normalized token set of c; we deduplicate by discarding chains c' for which the Jaccard similarity

$$J(c,c') = \frac{|T(c) \cap T(c')|}{|T(c) \cup T(c')|}$$

exceeds a similarity threshold. A chain is shortlisted if it attains a high $s(c)$, contains a critical cue $e^* \in \mathcal{C}$ with high $\widetilde{\mathrm{conf}}(e^*)$, or exhibits sufficient multi-type evidence (measured by a monotone function $g(E(c))$); we then retain only a small top subset per QA. A judge ensemble $\{v_j\}_{j \in \mathcal{J}}$, with votes $v_j \in \{\text{yes}, \text{no}\}$, determines whether a chain is erroneous and assigns a category $\ell(c) \in \mathcal{L}$, where $\mathcal{L} = \{\text{logical}, \text{hallucination}, \text{perceptual}, \text{temporal}, \text{spatial}\}$; acceptance is by judge consensus, e.g.,

$$\mathbf{1}\Big[\sum_{j \in \mathcal{J}} \mathbf{1}(v_j = \text{yes}) \geq \kappa\Big] = 1,$$

with κ a consensus threshold. Accepted items receive structured critiques and undergo brief human verification of `has_error`, tags, and yielding a judge- and human-confirmed Realistic split for training and evaluation.

3. VIDEOCRITIC-3B

We employ a three-phase training framework for joint error detection and explanation: supervised fine-tuning (SFT) followed by two targeted DPO passes. Given K frames V, a question q, and a candidate reasoning chain c (with answer), VIDEOCRITIC-3B predicts a structured verdict $\{$`has_error`, `types` $\subseteq \{$logical,hallucination,perceptual,temporal,spatial$\}$, `note`$\}$, where `note` provides a concise rationale for the detected errors. The model is trained in three phases: supervised fine-tuning (SFT) for grounding and calibration, followed by two Direct Preference Optimization (DPO) passes to align generation with verifiable human-defined preferences. Generic DPO tends to favor high-frequency error patterns, especially hallucination and perceptual errors, because they dominate the preference data. This can inadvertently suppress logical error detection. We apply two targeted DPO passes to sharpen type selectivity while maintaining sensitivity to less frequent categories such as logical errors. VIDEOCRITIC-3B uses a frozen vision tower to encode input frames, a lightweight projector to map visual features into the language space, and a 3B decoder-only language model equipped with five type-specific heads for multi-label detection; we derive the overall `has_error` score from these heads to ensure consistency between `has_error` and the predicted type set. (An auxiliary coarse head is used only during SFT as an additional supervision signal and is not used for inference-time scoring.)

Stage A: Supervised Fine-Tuning (SFT). Each training instance consists of interleaved inputs $[\texttt{IMG}_1, \ldots, \texttt{IMG}_K, q, c]$,

Table 1. **Breakdown of data provenance, error coverage, and aggregate statistics for training and testing splits.** We build this data based on the reasoning chain from Han et al. [10]. Training data combine rule-based synthesis and VLM-injected reasoning traces with balanced error coverage. Testing data reflect real-world VLM behavior, where hallucination and logical errors are most common. However, we also retain a small coverage of perceptual, temporal, and spatial grounding errors. Columns report item counts, data share, average error mentions per item, and per-type prevalence (%). Ment./Item denotes the average number of labeled error mentions (sentential spans) per reasoning chain.

Split	Source	Items	Share (%)	Ment./Item	Logical	Hallucination	Perceptual	Temporal	Spatial
Train	Rule-based	5,000	39.9	2.01	35.1	35.0	34.9	35.8	34.4
	Qwen2.5-VL-72B [3] (VLM-injected)	3,040	24.3	3.86	77.2	82.0	56.9	73.5	68.8
	Moonshot-v1 [1] (VLM-injected)	1,479	11.8	3.02	55.9	65.2	49.8	24.5	14.9
	Clean prompts [10] (No error)	3,000	24.0	0.00	0.0	0.0	0.0	0.0	0.0
	Overall (Train)	12,519	100.0	2.10	39.4	41.6	33.6	35.0	32.2
Test	Rule-based	5,000	30.5	0.80	15.3	31.3	30.7	0.0	0.0
	Moonshot-v1 (Realistic)	2,340	14.3	0.30	12.1	2.2	10.4	2.1	3.6
	Gemini-2.5-Flash [7] (Realistic)	2,288	14.0	0.22	12.5	4.6	0.4	1.9	2.6
	Qwen2.5-VL-72B (Realistic)	1,530	9.3	0.09	8.9	0.4	0.0	0.0	0.1
	Qwen2.5-VL-3B (Realistic)	860	5.2	0.24	7.6	6.7	0.0	0.0	0.6
	LLaVA-1.5-4B [14] (Realistic)	1,329	8.1	1.08	20.6	61.2	0.0	0.1	0.1
	InternVL3.5-4B [5] (Realistic)	1,111	6.8	0.88	6.6	56.3	0.0	0.1	0.1
	Qwen3-VL-4B [30] (Realistic)	552	3.4	3.29	19.2	100.0	0.0	0.0	0.0
	Moonshot-v1 (VLM-injected)	1,382	8.4	1.80	44.4	50.9	38.4	18.4	8.4
	Overall (Test)	16,392	100.0	0.75	15.9	27.4	14.2	2.1	1.7

where q is the question and c is the candidate reasoning chain. The critic is optimized with two complementary objectives: (i) sequence generation of the structured JSON verdict, and (ii) multi-label detection of error presence. Let y_e denote the binary label for overall error (`has_error`) and y_t the indicator for each error type $t \in \mathcal{T}$. The model predicts corresponding logits z_e and z_t through a coarse classifier and five type-specific heads. The supervised fine-tuning loss is: $\mathcal{L}_{\text{SFT}} = \lambda_{\text{text}} \, \text{CE}(\hat{y}_{1:T}, y^{\star}_{1:T}) + \lambda_e \, \text{BCE}(\sigma(z_e), y_e) + \sum_{t \in \mathcal{T}} \lambda_t \, \text{BCE}(\sigma(z_t), y_t)$, where $\hat{y}_{1:T}$ is the generated JSON token sequence and $y^{\star}_{1:T}$ is the ground truth. To avoid overfitting and preserve visual features, the vision tower remains frozen; we train only the projector, language model, and detection heads. Rare error types are up-weighted via mild class reweighting and sampled with a balanced sampler. After SFT, each type head is calibrated with temperature scaling on a held-out validation split, and thresholds $\{\tau_t\}_{t \in T}$ are selected to maximize dev-set F1. At inference, we define the overall error prediction as the logical OR of per-type predictions: $\texttt{has\_error} = \mathbb{1}[\max_{t \in T} p_t \geq \tau_t]$.

Stage B: Direct Preference Optimization. SFT yields grammatical but sometimes over- or under-inclusive critiques. We therefore apply DPO on pairs (y^+, y^-) of candidate JSONs for the same (V, q, c), where a deterministic verifier selects y^+ if it (i) matches the gold `has_error`, (ii) maximizes Jaccard overlap with the gold type set, and (iii) satisfies format/length and "no-hedging" rules. We optimize $\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,y^+,y^-)} \log \sigma\Big(\beta\big[\log \pi_\theta(y^+|x) - \log \pi_\theta(y^-|x) - \log \pi_{\text{ref}}(y^+|x) + \log \pi_{\text{ref}}(y^-|x)\big]\Big)$, using the SFT checkpoint as π_{ref} and keeping the vision tower frozen. A light agreement penalty encourages the generated `types` to match the detector mask, improving textual faithfulness without degrading calibrated probabilities.

4. Experiments

We conduct comprehensive experiments to evaluate VIDEOCRITIC across three dimensions: dataset characteristics, chain-aware QA robustness, and reasoning error detection. Our studies span both synthetic and realistic settings, comparing VIDEOCRITIC-3B with strong VLM baselines under multiple evidence conditions. We further analyze per-type detection performance, calibration behavior, and the effect of each training stage, including supervised fine-tuning and preference optimization.

4.1. VIDEOCRITIC-BENCH

Dataset Statistics. Building on the reasoning-chain annotations of Han et al. [10], VIDEOCRITIC-BENCH contains a synthetic split (rule-based and LLM-assisted error injections) and a realistic split mined from diverse VLM outputs. All items are labeled under five error types, following the taxonomy in Section 2.1. Table 1 reports item counts, per-type prevalence, and the average number of labeled error mentions per chain (Ment./Item). The training split is constructed to be near-balanced across types, while the test split reflects naturally occurring VLM failures with strong skew toward hallucination and logical errors and very sparse

Table 2. **Chain-aware QA accuracy on VIDEOCRITIC-BENCH.** We report answer accuracy (%) for five VLMs under five evidence conditions: Raw: no reasoning chain; Gold: ground-truth chain; Rule-based: synthetic chains with rule-injected errors; LLM-injected: synthetic chains with LLM-injected errors; Realistic: chains mined from real VLM outputs. Each entry is the accuracy (in %) achieved by the corresponding model.

Model	Raw	Gold	Rule-based	LLM-injected	Realistic
Qwen2.5-VL [3]	50.07	75.62	64.88	64.82	40.36
Gemini-2.5-Flash [7]	64.33	80.20	64.41	67.79	53.89
Intern3.5VL [5]	64.40	83.93	63.95	63.52	58.74
LLaVA-OV-1.5 [2]	53.83	81.17	42.00	42.33	35.04
Gemma3–12B [23]	43.83	72.67	69.32	68.51	63.13

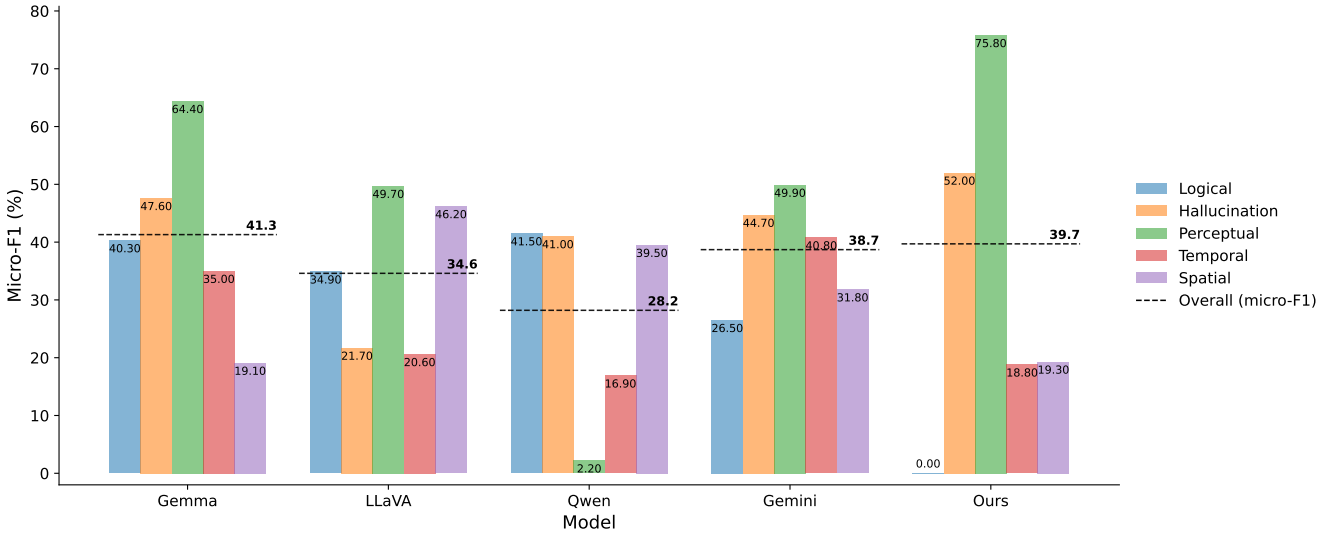

Figure 4. **Model Performance Across Error Types.** We report the Micro-F1 score of each model's performance in detecting different error types. Overall is micro-F1 aggregated across types. Gemma3–12B[1] [23], LLaVA–OneVision–1.5–8B[2] [2], Qwen2.5–VL–7B[3] [3], Gemini2.5–Flash[4] [7]. Ours' corresponds to VIDEOCRITIC-3B after the first DPO pass.

temporal/spatial cases. The realistic split aggregates real reasoning chains produced by Gemini-2.5-Flash [7], LLaVA-1.5-4B [11, 14], InternVL3.5-4B [5], Qwen variants [3], and others. This composition provides a balanced training set and a "stress-test" evaluation set that mirrors real VLM failure modes. This sparsity is partly due to the rule-based test subset having zero temporal/spatial prevalence (Table 1).

Chain-aware QA. As in Table 2, we assess how the presence and quality of reasoning chains affect VLM answer accuracy by evaluating each model under five evidence conditions: no chain, the gold chain which is ground truth reasoning, synthetic rule-based error injected chains, LLM-based error injected chains, and realistic reasoning error chains. Gold chains consistently improve accuracy, indicating that VLMs can benefit from correct external reasoning. In contrast, both synthetic and realistic erroneous chains reduce accuracy to varying extents, revealing that existing VLMs are susceptible to flawed reasoning and often propagate errors when the provided chain is incorrect.

Error-type Detection. We further compare the micro–F1 performance of four VLMs on each error category to assess their aggregate detection capability. For evaluation, we treat each reasoning chain as a multi-label instance by taking the union of all annotated sentence-level error mentions as its gold type set, and report multi-label micro-F1 over the resulting type sets. As shown in Figure 4, Gemma3–12B [23] achieves the strongest overall micro–F1 among the baselines, driven by a high sensitivity to perceptual and hallucination errors. Gemini2.5–Flash [7] provides competitive overall performance and remains the most reliable model for temporal grounding. LLaVA–OneVision–1.5–8B [2] performs best on spatial misgrounding, but lags on hallucination detection, while Qwen2.5–VL–7B [3] shows solid logical detection with weaker performance on perceptual cues. Overall, the micro–F1 results reinforce that current VLMs excel in different subsets of reasoning failures, but none offer uniformly strong error detection across all types.

4.2. VIDEOCRITIC-3B

VIDEOCRITIC-3B Training. VIDEOCRITIC-3B is initialized from a Qwen2.5-VL-3B backbone with the vision tower kept frozen and a trainable multimodal projector & language decoder. We first perform supervised fine-tuning (SFT) on the VIDEOCRITIC-BENCH training split. SFT is optimized with AdamW under a cosine schedule (LM learning rate 2×10^{-5}; projector and head learning rates 10^{-4}; 5% warmup), cutoff length 4096, bf16 precision, and an effective batch size of 32. To improve type selectivity, we apply two targeted DPO [21] passes: a first pass that improves selectivity on prevalent error patterns, followed by a second pass that restores sensitivity to logical errors that can be suppressed by preference-data skew. Both stages use the SFT model as the reference policy, keep the vision tower frozen, and train only the decoder and projector for one epoch with AdamW (LM learning rate 5×10^{-6}; projector learning rate 10^{-5}; weight decay 0.01; 5% warmup; β_{dpo}=0.1). All evaluations decode a single JSON line under a JSON-only constraint using nucleus sampling (temperature = 0.2, top-p = 0.9, max_new_tokens = 256). Per–head temperature scaling and threshold tuning are performed once on a held–out development subset, and the overall decision follows the OR–gating rule described in Section 3. Unless otherwise noted, VIDEOCRITIC-3B refers to the checkpoint after the first (generic) DPO pass, which yields the highest overall micro-F1 in Table 4. We denote the second logical-focused pass as DPO+, and report it as a trade-off variant that restores logical detection while reducing overall micro-F1.

Table 3. **Per-type Calibration and Threshold Tuning.** We report calibration metrics (Brier, ECE_{15}) and thresholded decision metrics (Precision/Recall/F1) for the five error-type heads on a 400-sample Realistic dev subset. Per-head temperature scaling improves probability calibration (lower Brier/ECE), and dev-set threshold tuning yields more stable decisions for low-frequency types (e.g., temporal/spatial recover non-zero F1).

Head	Brier ↓	ECE_{15} ↓	Prec. ↑	Rec. ↑	F1 ↑
Halluc. (before)	0.239	0.266	**0.317**	0.511	**0.391**
Halluc. (after)	**0.230**	**0.241**	0.286	**0.600**	0.387
Logical (before)	0.336	0.445	0.164	0.985	0.282
Logical (after)	**0.269**	**0.363**	**0.166**	**1.000**	**0.285**
Percept. (before)	0.442	0.565	0.144	**1.000**	0.252
Percept. (after)	**0.292**	**0.412**	**0.158**	0.814	**0.265**
Temporal (before)	0.316	0.539	0.028	**0.909**	0.054
Temporal (after)	**0.265**	**0.490**	**0.143**	0.182	**0.160**
Spatial (before)	0.108	0.268	0.000	0.000	0.000
Spatial (after)	**0.038**	**0.031**	**0.143**	**0.133**	**0.138**

Calibration and Threshold Stability. We evaluate per-head temperature scaling and dev-set threshold tuning on a 400-sample subset of the Realistic split. Table 3 reports calibration metrics (Brier, ECE_{15}) and thresholded decision metrics (precision/recall/F1) for the five error-type heads before and after calibration. After temperature scaling, Brier and ECE decrease for every head, indicating improved probability calibration. With dev-set threshold tuning, low-frequency heads become more stable and recover non-zero F1 (e.g., temporal and spatial). Across the five types, macro-/micro-F1 improve from 0.196/0.232 (uncalibrated, threshold 0.5) to 0.247/0.297 after temperature scaling and threshold sweeping, and we use this shared calibration protocol for all subsequent evaluations.

Table 4. **Ablation on training stages for VIDEOCRITIC-3B.** We report per-type F1 (%) for each error category. **Overall** denotes **micro-F1** aggregated across all five categories (i.e., all type labels pooled). Raw denotes the frozen base 3B model; SFT adds supervised fine-tuning; DPO applies generic preference optimization; DPO+ further applies a logical-focused DPO pass.

Stage	Logical	Halluc.	Percept.	Temporal	Spatial	Overall (micro-F1)
Raw	11.1	0.0	7.3	0.0	0.0	3.9
SFT	25.3	34.9	23.3	20.3	8.7	29.7
DPO	0.0	52.0	75.8	18.8	19.3	39.7
DPO^{+}	27.9	41.9	23.4	5.0	0.0	23.3

Comparison to Prior VLMs. Figure 4 shows that our model exhibits distinct strengths and weaknesses across the five error categories. It achieves the highest scores on hallucination and perceptual errors: two of the most prevalent failure modes in VIDEOCRITIC-BENCH, indicating strong sensitivity to unsupported content and visual misrecognitions. Performance on temporal and spatial grounding is comparable to mid-tier baselines, though still below the best-performing models in these categories. The primary limitation lies in logical error detection, where existing VLMs such as Gemma3–12B [23] and Qwen2.5–VL–7B [3] remain substantially stronger. This is likely because logical inconsistencies depend less on visual cues and more on robust textual reasoning, an ability that larger language backbones capture more effectively. However, with targeted fine-tuning, we substantially improve the performance of logical error detection, as shown in Table 4. Taken together, the results show that while our model excels on high-frequency visual and hallucination-related errors, additional refinement is required to match state-of-the-art performance on logic-intensive reasoning failures.

Ablation Studies. Table 4 quantifies the effect of each training stage on per-type error detection. The raw 3B backbone has almost no diagnostic ability with micro-F1 near 3.9, with near-zero performance on hallucination, temporal, and spatial errors. Supervised fine-tuning (SFT) substantially improves all categories, raising the overall F1 score. Adding generic DPO further boosts hallucination and perceptual detection (52.0 and 75.8 F1 score, respectively) and yields the highest micro-F1 score, but collapses the logical head with score, reflecting a strong bias toward the prevalent hallucination/perceptual patterns. The additional logical-focused DPO pass (DPO^{+}) recovers logical F1 (27.9) and improves

hallucination precision, but at the cost of temporal and spatial detection and reduces the Overall score to 23.3. These trends highlight a trade-off: generic DPO is highly effective for frequent error types, but can destabilize rarer categories, motivating more balanced preference design in future work.

5. Related Works

Video Reasoning. The capabilities of Vision-Language Models (VLMs) are rapidly evolving from static image perception [4, 19, 22] to dynamic video reasoning [6, 18, 20, 31, 34, 37]. Given multiple sampled frames from a video, VLMs are required not only to accurately perceive each individual frame, but also to understand the underlying spatiotemporal relationships among frames and organize them into a unified long-term memory for consistent video comprehension. To unlock stronger video reasoning capabilities, recent works such as LLaVA-Vid [35], VideoChat-R1 [13], Fact-R1 [33], Ego-R1 [24], and Video-RTS [28] adopt data-centric strategies that synthesize large-scale video QA datasets, combined with training paradigms including supervised fine-tuning and reinforcement learning. Although these approaches have led to notable progress, a substantial gap still remains between current VLMs and human-level video reasoning. For instance, existing models often struggle with temporal understanding [15], fail to maintain consistent identity tracking across frames [34], and are prone to hallucinations during the reasoning process [9, 12].

Critic Models. As tasks grow increasingly complex, critic models play a crucial role in enhancing system capabilities by providing structured feedback, such as assigning scores, performing pairwise comparisons, and delivering detailed natural-language critiques. Prior work has demonstrated that incorporating critic mechanisms into VLMs can substantially improve factual accuracy, logical consistency, and overall robustness in reasoning. For instance, Critic-V [32] adopts an actor–critic framework in which a reasoner generates rationales while a critic evaluates them in real time, thereby improving accuracy through DPO [21]. CAViAR [17] and ReAgent-V [38] introduce critic modules that identify and select the most reliable final answer from multiple candidates. More recent works, such as R1-Reward [36] and LLaVA-Critic-R1 [27], focus on constructing verifiable critic datasets and training generative critic models via reinforcement learning, enabling models to self-improve through iterative feedback. Within the domain of video reasoning, the need for such critic models is even more pronounced, as video understanding involves richer factual content and introduces more aspects that must be carefully verified.

Video Reasoning Benchmarks. Prior datasets address slices of video–language reasoning failures, whereas VIDEOCRITIC-BENCH provides the first unified, fine-grained benchmark explicitly designed for error diagnosis. VidHalluc [12] focuses on temporal hallucinations and offers answer-level labels without reasoning chains; TempCompass [15] evaluates temporal understanding but does not annotate reasoning errors or provide chain-level supervision. VCR-Bench [20] includes human-written rationales but aims to assess correct multi-step reasoning rather than identifying failure modes. MMVU [37] provides expert-level reasoning on domain-specific videos, but lacks an explicit error taxonomy and does not supervise the detection of incorrect reasoning. In contrast, VIDEOCRITIC-BENCH introduces a five-type taxonomy (Figure 2), offers both synthetic and realistic errors, and includes chain-level annotations with explicit error mentions. Moreover, it is the only benchmark to pair controlled error injection with a human-verified stress-test split of authentic VLM failures, enabling rigorous and comprehensive assessment of robust video reasoning. We compare VIDEOCRITIC-BENCH with prior video understanding error datasets in Table 5.

6. Conclusion

We introduced VIDEOCRITIC, a framework for diagnosing and categorizing reasoning errors in video language models. Our contributions are threefold. (1) We released VIDEOCRITIC-BENCH, a benchmark with a synthetic split built via rule-based and LLM-assisted error injections and a realistic, human-verified split mined from diverse VLMs. Each item is annotated under a five-type taxonomy, enabling analysis beyond answer accuracy. The dataset provides 12.5K training and 16.4K testing items; training is near-balanced across types, while testing reflects real-world failures dominated by hallucination and logical errors. (2) We proposed VIDEOCRITIC-3B, a 3B critic model that predicts error existence and per-type presence through constrained JSON outputs. The model is trained with supervised fine-tuning and Direct Preference Optimization (DPO), followed by per-head calibration and OR-gated decisions. (3) We designed an evaluation suite covering chain-aware QA robustness, error-type detection, and ablations on training stages. Baselines show complementary strengths, yet none is uniformly reliable. VIDEOCRITIC-3B achieves its strongest gains on hallucination and perceptual errors, two high-prevalence categories in the realistic split.

References

[1] Moonshot AI. Moonshot-v1, 2024. Large language model with up to 128K context window. 5

[2] Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Chunsheng Wu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025. 6, 1

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 5, 6, 7

[4] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021. 8

[5] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 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*, pages 24185–24198, 2024. 5, 6, 1

[6] Zixu Cheng, Jian Hu, Ziquan Liu, Chenyang Si, Wei Li, and Shaogang Gong. V-star: Benchmarking video-llms on video spatio-temporal reasoning. *arXiv preprint arXiv:2503.11495*, 2025. 2, 8

[7] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 5, 6

[8] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 2

[9] Hongcheng Gao, Jiashu Qu, Jingyi Tang, Baolong Bi, Yue Liu, Hongyu Chen, Li Liang, Li Su, and Qingming Huang. Exploring hallucination of large multimodal models in video understanding: Benchmark, analysis and mitigation. *arXiv preprint arXiv:2503.19622*, 2025. 1, 8

[10] Songhao Han, Wei Huang, Hairong Shi, Le Zhuo, Xiu Su, Shifeng Zhang, Xu Zhou, Xiaojuan Qi, Yue Liao, and Si Liu. Videoespresso: A large-scale chain-of-thought dataset for fine-grained video reasoning via core frame selection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26181–26191, 2025. 3, 5

[11] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer, 2024. 6

[12] Chaoyu Li, Eun Woo Im, and Pooyan Fazli. Vidhalluc: Evaluating temporal hallucinations in multimodal large language models for video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13723–13733, 2025. 1, 2, 8

[13] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025. 1, 8

[14] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 5, 6

[15] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? *arXiv preprint arXiv:2403.00476*, 2024. 1, 2, 8

[16] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023. 2

[17] Sachit Menon, Ahmet Iscen, Arsha Nagrani, Tobias Weyand, Carl Vondrick, and Cordelia Schmid. Caviar: Critic-augmented video agentic reasoning. *arXiv preprint arXiv:2509.07680*, 2025. 8

[18] Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv preprint arXiv:2504.01805*, 2025. 8

[19] Renjie Pi, Jianshu Zhang, Jipeng Zhang, Rui Pan, Zhekai Chen, and Tong Zhang. Image textualization: An automatic framework for creating accurate and detailed image descriptions. *Advances in Neural Information Processing Systems*, 37, 2024. 8

[20] Yukun Qi, Yiming Zhao, Yu Zeng, Xikun Bao, Wenxuan Huang, Lin Chen, Zehui Chen, Jie Zhao, Zhongang Qi, and Feng Zhao. Vcr-bench: A comprehensive evaluation framework for video chain-of-thought reasoning. *arXiv preprint arXiv:2504.07956*, 2025. 2, 8

[21] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023. 7, 8

[22] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 8

[23] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. 6, 7

[24] Shulin Tian, Ruiqi Wang, Hongming Guo, Penghao Wu, Yuhao Dong, Xiuying Wang, Jingkang Yang, Hao Zhang, Hongyuan Zhu, and Ziwei Liu. Ego-r1: Chain-of-tool-thought for ultra-long egocentric video reasoning. *arXiv preprint arXiv:2506.13654*, 2025. 2, 8

[25] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 1

[26] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, et al. Lvbench: An extreme long video understanding benchmark. *arXiv preprint arXiv:2406.08035*, 2024. 2

[27] Xiyao Wang, Chunyuan Li, Jianwei Yang, Kai Zhang, Bo Liu, Tianyi Xiong, and Furong Huang. Llava-critic-r1: Your critic model is secretly a strong policy model. *arXiv preprint arXiv:2509.00676*, 2025. 8

[28] Ziyang Wang, Jaehong Yoon, Shoubin Yu, Md Mohaiminul Islam, Gedas Bertasius, and Mohit Bansal. Video-rts: Rethinking reinforcement learning and test-time scaling for efficient and enhanced video reasoning. *arXiv preprint arXiv:2507.06485*, 2025. 2, 8

[29] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786, 2021. 2

[30] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 5, 1

[31] Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, et al. Spatial mental modeling from limited views. In *Structural Priors for Vision Workshop at ICCV'25*, 2025. 8

[32] Di Zhang, Jingdi Lei, Junxian Li, Xunzhi Wang, Yujie Liu, Zonglin Yang, Jiatong Li, Weida Wang, Suorong Yang, Jianbo Wu, et al. Critic-v: Vlm critics help catch vlm errors in multimodal reasoning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9050–9061, 2025. 2, 8

[33] Fanrui Zhang, Dian Li, Qiang Zhang, Junxiong Lin, Jiahong Yan, Jiawei Liu, Zheng-Jun Zha, et al. Fact-r1: Towards explainable video misinformation detection with deep reasoning. *arXiv preprint arXiv:2505.16836*, 2025. 2, 8

[34] Jianshu Zhang, Dongyu Yao, Renjie Pi, Paul Pu Liang, and Yi R Fung. Vlm2-bench: A closer look at how well vlms implicitly link explicit matching visual cues. *arXiv preprint arXiv:2502.12084*, 2025. 8

[35] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 8

[36] Yi-Fan Zhang, Xingyu Lu, Xiao Hu, Chaoyou Fu, Bin Wen, Tianke Zhang, Changyi Liu, Kaiyu Jiang, Kaibing Chen, Kaiyu Tang, et al. R1-reward: Training multimodal reward model through stable reinforcement learning. *arXiv preprint arXiv:2505.02835*, 2025. 8

[37] Yilun Zhao, Haowei Zhang, Lujing Xie, Tongyan Hu, Guo Gan, Yitao Long, Zhiyuan Hu, Weiyuan Chen, Chuhan Li, Zhijian Xu, et al. Mmvu: Measuring expert-level multi-discipline video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8475–8489, 2025. 2, 8

[38] Yiyang Zhou, Yangfan He, Yaofeng Su, Siwei Han, Joel Jang, Gedas Bertasius, Mohit Bansal, and Huaxiu Yao. Reagent-v: A reward-driven multi-agent framework for video understanding. *arXiv preprint arXiv:2506.01300*, 2025. 8

7. Data & Code

We include the test set in the supplementary material. The training set and code will be released after the paper is accepted and published.

8. Realistic Split Annotation Pipeline

The Realistic split collects authentic reasoning errors made by a diverse set of video–language models (VLMs). Unlike the Synthetic split, where we inject controlled corruptions into ground-truth chains, the Realistic split starts from raw VLM-generated reasoning and filters it through a multi-stage pipeline combining heuristic validators, LLM critics, and human verification.

8.1. VLM Candidate Generation

We begin from the 1 382 multiple-choice questions in the `bench_hard` subset of VideoEspresso. For each (v, q, a) item we sample five reasoning chains from several small open-source VLMs (all $\leq$4B parameters), using fixed decoding hyperparameters (temperature, top-p, and maximum length) for reproducibility. The small models include Qwen2.5-VL-3B-Instruct [25], Qwen3-VL-4B-Instruct [30], InternVL3.5-4B-Instruct [5], and LLaVA-OneVision-1.5-4B-Instruct [2]. In parallel, we sample additional chains from larger local VLMs (Qwen3-VL-32B, Qwen2.5-VL-72B) and from API models such as Gemini 2.5 Flash and Moonshot, again with five generations per question. This yields a mixed pool of candidate chains per QA, spanning both weaker and stronger generators and covering a wide variety of realistic error patterns.

8.2. Automatic Cue Detection and Scoring

Each candidate chain c is passed through a bank of lightweight validators that operate over the text, the video-derived object/box lexicons, and the answer options. Validators fire *cues* $e_i \in E(c)$ corresponding to our five taxonomy types—logical, hallucination, perceptual, temporal grounding, and spatial grounding—as well as auxiliary signals such as answer mismatch, self-inconsistency, or explanation errors.

Every cue is associated with a type $\tau(e_i)$, a base confidence $\mathrm{conf}(e_i) \in [0, 1]$ (chosen from a small set of discrete buckets), and a non-negative class weight $w_{\tau(e_i)}$ that reflects the importance of that error type. To capture agreement across generators, we apply a consensus boost:

$$\widetilde{\mathrm{conf}}(e_i) = \mathrm{conf}(e_i) + \beta\, \mathbf{1}_{\text{gen}},$$

where $\mathbf{1}_{\text{gen}} = 1$ if an equivalent cue is detected in a chain from another generator and 0 otherwise, and $\beta \geq 0$ is a fixed boost. The overall *suspicion score* of chain c is then

$$s(c) = \sum_{e_i \in E(c)} w_{\tau(e_i)}\, \widetilde{\mathrm{conf}}(e_i).$$

Soft cues from option alignment (e.g., high Jaccard overlap between the chain and a wrong option) are injected into the same pool and treated as additional hallucination/logical signals. At this stage there is no learned calibration; all confidences are deterministic functions of validator hits and consensus.

8.3. Deduplication and Shortlisting

For each QA, we next deduplicate and shortlist chains based on both textual similarity and score. Let $T(c)$ be the normalized token set of chain c. We discard c' whenever the token Jaccard similarity

$$J(c, c') = \frac{|T(c) \cap T(c')|}{|T(c) \cup T(c')|}$$

exceeds a fixed threshold (0.85 in our implementation), keeping only one representative among near-duplicates.

The remaining chains are sorted by $s(c)$ in descending order. A chain is eligible for the shortlist if it (i) has $s(c)$ above a score threshold, (ii) contains at least one *critical* cue $e^* \in \mathcal{C}$ (e.g., strong answer mismatch) with high $\widetilde{\mathrm{conf}}(e^*)$, or (iii) exhibits sufficient multi-type evidence according to a monotone function $g(E(c))$ that increases with the number and diversity of error types. We greedily build a shortlist of at most $K_{\max} = 4$ chains per QA, prioritizing consensus-marked chains and promoting diversity over error types.

8.4. LLM Judging and Human Verification

Each shortlisted chain is further judged by an ensemble of high-capacity LLM critics (Qwen2.5-VL-72B, Gemini 2.5 Flash, and Moonshot). Given (v, q, a, c), each judge returns a binary decision $v_j \in \{\text{yes}, \text{no}\}$ indicating whether the chain contains a reasoning error, together with a subset of error labels $\ell_j(c) \subseteq \mathcal{L}$, where $\mathcal{L} = \{\text{logical}, \text{hallucination}, \text{perceptual}, \text{temporal}, \text{spatial}\}$. A chain is accepted as *candidate realistic error* if at least κ judges vote `yes`; we then assign it the merged label set $\ell(c) = \bigcup_{j:v_j=\text{yes}} \ell_j(c)$ and store the judges' rationales.

All shortlisted chains are finally labeled by two independent human annotators with access to the video, question, answer, and the candidate chain. Annotators assign `has_error` and the full type set following Appendix C. Judge outputs are used only for shortlisting and for post-hoc quality checks; annotators do not see judge rationales during labeling. We discard any chain with disagreement on `has_error` or on any type, yielding the released Realistic split."

9. Error Taxonomy & Annotation Guidelines

This section specifies the labeling rules used in VIDEOCRITIC-BENCH for the five error types (logical, hallucination, perceptual, temporal grounding, and

Table 5. **Comparison of VIDEOCRITIC-BENCH with prior Video Understanding Error Datasets.** VIDEOCRITIC-BENCH provides a five-type error taxonomy, chain-level supervision, both synthetic and realistic VLM-generated errors, and a human-verified stress-test split. Prior benchmarks typically focus on a single error type, lack chain-level reasoning, or omit realistic VLM failures.

Dataset	Size	Error Types	Chain Supervision	Error Source	Evaluation Modes	Realistic Errors
VidHalluc	9.3K	Hallucination only	No	Real videos	Binary QA (Y/N)	No
TempCompass	7.5K	Temporal only	No	Real videos	MC, YN, captioning	No
VCR-Bench	1.0K	None (implicit)	Yes (gold chains)	Human curated	QA + CoT scoring	No
MMVU	3.0K	Domain reasoning	Yes (expert notes)	Expert videos	MC-QA	No
VIDEOCRITIC-BENCH	**12.5K / 16.4K**	**5 types (L/H/P/T/S)**	**Yes**	Synthetic + Realistic	Multiple Choice QA	**Yes (human-verified)**

spatial grounding), together with global instructions and quality checks for human annotators. The goal is to keep labels consistent across the synthetic and realistic splits and to disambiguate closely related failure modes.:contentReference[oaicite:0]index=0

9.1. Logical Errors

Definition. A sentence is tagged as *logical* when it contains an invalid inference that is not directly attributable to misperception or misgrounding. Typical cases include contradictions with earlier steps, unsupported causal links, and arithmetic or magnitude errors that are inconsistent with the video evidence or the previously stated facts.

Positive cues. Common surface markers are causal connectives such as "because", "since", "therefore", "thus", "hence" followed by a conclusion that does not follow from the frames or the ground-truth evidence graph (e.g., "Therefore the statue must be moving toward the camera", "Since the chef burned the dish earlier, the crowd jeers now").

Borderline cases and priority. Statements that simply paraphrase an answer option without adding causal structure are not tagged as logical. When a sentence both misorders events or misplaces objects *and* draws an invalid conclusion, temporal or spatial grounding takes precedence; the logical tag is reserved for errors without a clear temporal/spatial anchor.

9.2. Hallucination

Definition. *Hallucination* covers any newly introduced entity or event that is not supported by the video: a noun phrase (typically $\leq$2 content tokens) that does not appear in captions, key items, or the evidence lexicon, or a claim that directly contradicts what is shown.

Positive cues. Examples include inventing extra objects ("an extra referee walks in"), new actors ("a neon drone hovers beside the scene"), or unseen events ("the hero triggers an explosion off-screen").

Hallucination vs. perceptual. If the referenced object exists but its attributes or count are wrong (e.g., incorrect colour or number), the error is labeled *perceptual* instead of hallucination. Hallucination is reserved for entities or events that are entirely absent from the ground-truth evidence.

9.3. Perceptual Errors

Definition. *Perceptual* errors arise from misreading visual attributes of an actually present object or action: wrong colour, texture, count, or fine-grained appearance. The reasoning may be logically coherent, but it is grounded in an incorrect description of the frames.

Positive cues. Typical examples include statements such as "Image 2 shows a green banner", when the banner is red; "Image 4 reports three helmets" when only one is visible; or "The cook wears a blue apron" when the apron is white.

Borderline cases. Introducing a wholly new object is hallucination. Misordering events is temporal. If a sentence mentions a wrong attribute without tying it to a specific frame, annotators only apply the perceptual tag when the mismatch is clear from the evidence lexicon; otherwise it is left untagged.

9.4. Temporal Grounding

Definition. *Temporal grounding* errors describe incorrect relations in time: wrong before/after order, misassigned timestamps, or incorrect duration of events.

Positive cues. These errors typically involve explicit temporal markers or frame indices, such as "before/after/earlier/later" or "image i vs. image j", where the stated order contradicts the actual frame order (e.g., "Image 3 is said to occur after image 4" when the reverse is true).

Borderline cases and priority. Vague temporal language ("eventually", "soon") without a clear comparison of two events is not tagged as temporal. Whenever a sentence contains an explicit before/after or frame-index comparison, the temporal tag is applied first; any accompanying unsupported

conclusion is not additionally labeled as logical unless there is an independent logical error.

9.5. Spatial Grounding

Definition. *Spatial grounding* errors concern incorrect localization in the image plane: misplacing objects with respect to left/right, up/down, foreground/ background, or specific regions indicated by bounding boxes.

Positive cues. Examples include claims such as "the statue sits on the right wall" when it is clearly on the left, "the diver is above the shark" when they are reversed, or "the truck is on the north side of the road" when it is on the south.

Borderline cases and priority. Generic location phrases ("nearby", "over there") without directional keywords are normally not tagged as spatial. If a sentence both hallucinates a new object and specifies its position, the hallucinated entity is labeled as hallucination; spatial is reserved for misplacing objects that do exist in the evidence. As with temporal errors, explicit directional anchors ("left", "upper", coordinates) give priority to the spatial tag over logical.

9.6. Global Instructions and Tricky Cases

Multi-type spans. A single sentence may legitimately carry multiple tags. The automatic pipeline accumulates all detected cues per sentence, and annotators are encouraged to keep multi-type labels (e.g., hallucination+spatial) when multiple failure modes are clearly expressed.

Missing vs. extra events. Extra invented events are labeled as hallucination. When an event is omitted but the remaining events are misordered or mislocalized, annotators use temporal or spatial tags; there is no separate "missing" label, and omissions can be described in the free-form note field.

Local vs. global contradictions. Guidelines focus on local, sentence-level inconsistencies. Annotators label each explicit mismatch separately; there is no additional global error type beyond tagging every offending claim and describing any broader contradiction in the note.

Short chains. Very short chains (even a single sentence) are still eligible for tagging. Heuristic detectors and annotators apply the same taxonomy; brevity alone is not a reason to skip labeling.

9.7. Quality Checks and Human Verification

All LLM-generated chains in the Realistic split are first filtered and typed by the automatic pipeline (heuristic cues and LLM detectors) and are then reviewed by two independent human verifiers. Annotators see the video frames, question, answer, candidate chain, and the candidate chain only; proposed types are not shown during labeling. Their task is to confirm or reject the automatic judgments, not to invent new entities or rewrite the story: notes and tags must only reference objects and events supported by the evidence graph.

A chain is included in the released Realistic split only if *both* verifiers agree that it contains an error and agree on the full set of error types. Chains with disagreement on `has_error` or on any type are discarded. Final manifests record type frequencies per shard to allow audit of drift over time.

10. Prompts and Evaluation Details

This section documents the prompts and inference settings used for (1) generating reasoning chains for VIDEOCRITICBENCH, (2) running critic models and detection baselines, and (3) computing QA accuracy. We summarize the common prompt templates and then detail detector hyperparameters, frame usage, parsing policies, and evaluation metrics.

Reasoning generation prompts. All VLMs used for chain generation (Qwen2.5-VL-3B, Qwen3-VL-4B, InternVL3.5-4B, LLaVA-OneVision-1.5-4B, and Gemma3) share a canonical VQA+reasoning prompt. The system message instructs the model to act as "a meticulous multimodal assistant for video question answering" and to always conclude with a line of the form `Final: (LETTER)`. The user prompt concatenates image tags (`Image-k: <image>`), the multiple-choice question and options, optional frame captions, and a final instruction: "Think aloud in 3–6 short steps under 'Reasoning:'. Then output a single line `Final: (LETTER)` choosing one option." Decoding settings differ slightly across models but always sample five chains per question (`num_return_sequences=5`). For example, Qwen2.5-VL-3B uses temperature 0.2, top-$p = 0.9$, and `max_new_tokens=256`; Qwen3-VL-4B uses temperature 0.3; InternVL3.5-4B uses 0.4; LLaVA-OneVision-1.5-4B uses 0.3, all with the same top-p and length cap. Gemma3 baselines reuse this template with only the underlying HF model identifier changed.

API-based generators used to mine the Realistic split (Gemini 2.5 Flash and Kimi/Moonshot) employ the same text prompts. Gemini uses the canonical system prompt and user structure, plus a short reminder template enforcing exact letter output; we set temperature 0.6, top-$p = 0.9$, and `max_output_tokens=512`, throttled to approximately three requests per second. Kimi uses the same textual prompt via Moonshot's API with temperature 0.7, top-$p = 0.9$, and `max_tokens=512`. In both cases, we request five chains

User Prompt

Image is the real image:
[IMG_1] <image>
…
Question:
{question}

Provided reasoning chain:
{chain}

Instructions:
- Diagnose whether the reasoning contains errors and list TYPES among logical/hallucination/perceptual/temporal/spatial.
- Output ONE JSON line: {"has_error": bool, "types": [...], "note": "..."} (≤30 words, no hedging).

Figure 5. **Error-Detection Prompt**.

User Prompt

TEXT:
{INPUT Reasoning Chain}

There are {FRAME_COUNT} images referred to as "image 1", "image 2", ...

Inject EXACTLY {K} {ERROR_TYPE} error(s) by appending {K} short sentence(s) (≤16 words each).
Keep them plausible and concise. No bullet lists, no numbering, no newlines.

Definitions (one-liners):
- hallucination: unsupported detail.
- perceptual: wrong color/count of an existing object and image.
- grounding_spatial: misplace an existing object within an image.
- grounding_temporal: misorder events between images.
- logical: add a faulty causal or deductive claim.

Rules:
- For perceptual/grounding_spatial/grounding_temporal: explicitly mention the image number(s).
- For hallucination/logical: do NOT mention any image number or digits.
- Append-only: do not change the original wording.
- Use everyday nouns; avoid fantasy objects.

Return JSON only in this form:
{
"mutated": "...",
"num_errors": {K},
"errors": [
{"type": "{ERROR_TYPE}", "claim": "the exact sentence you appended"}
]
}
# Note: (The "mix" variant replaces the third paragraph with "Inject EXACTLY {K} errors... Use these error types in order: {ERROR_SEQUENCE}" and adds "Do not combine multiple error types in one sentence." in the rules list.)

Figure 6. **Error-Injection Prompt**.

per question and stream the resulting reasoning into the cue-detection and cross-checking pipeline.

Critic and detector prompts. VIDEOCRITIC-3B is prompted as a JSON-only auditor. Its system message describes it as "a rigorous vision–language auditor" and explicitly forbids natural-language commentary outside JSON. The user prompt provides frame sentinels, captions, key items, the question, and the candidate reasoning, and requests a JSON object summarizing errors (e.g., `"has_error": bool, "types": [...], "note": "..."`), as in Figure 5. During our evaluations, we decode with temperature 0.2, top-$p = 0.9$, and `max_new_tokens=256`.

Baseline critic runs with InternVL3.5-8B follow a similar schema. The system prompt instructs the model to inspect reasoning chains over videos, and the user prompt supplies `<image>` tokens, the question, and chain, plus strict instructions to emit a JSON object with keys `has_error`, `types`, and a short `note` ($\leq$ 30 words). The default decoding configuration is greedy (temperature 0.0, `do_sample=False`), with top-$p = 0.9$ and `max_new_tokens=256`. Other large detectors used only for mining and cross-checking (Qwen2.5-VL-72B, Gemini 2.5 Flash, and Kimi) share a lighter three-line detection prompt: a "decision" line (`yes/no`), an "errors" line listing types from {logical, hallucination, perceptual, grounding.spatial, grounding.temporal}, and a brief "reason" line.

LLM-based error injection (synthetic split). For the Synthetic split, we use specialized prompts to inject controlled errors into clean VideoEspresso-style chains. Single-type

templates ask a model to append exactly K short sentences that realize a specified error type, accompanied by textual definitions (e.g., hallucination vs. perceptual vs. grounding vs. logical) and constraints such as "grounding errors must cite an image index" or "hallucination/logical errors must not mention frame numbers." Mixed-type templates extend this by enforcing a specific sequence of types and explicitly forbidding mixing multiple types within a single sentence. Hyperparameters bound the number and length of injected errors (default 1–10 sentences, $\leq$16 words each) and require JSON-only output so that injected spans can be stored in structured fields, as in Figure 6.

Frame usage and formatting. All detectors cap the number of frames via a `-max-images` flag (default 6). For local models (InternVL, Qwen), frames are loaded from disk, padded or truncated to this limit, and passed as true image tensors; each frame is associated with an implicit index (Image 1, . . . , Image K) that the prompts and validators reference. Gemini and Kimi receive the same ordered list via their multimodal chat APIs (images encoded as base64 or API-native image payloads).

Retry and parsing policies. For InternVL/VideoCritic-3B, responses are parsed by searching for the first JSON block; if parsing fails, the sample is flagged `invalid_response` and no retry is attempted. In contrast, the Gemini and Kimi detection scripts enforce the three-line schema described above, apply a tolerant parser that normalizes case and whitespace (e.g., accepts "decision:" or "Decision:"), and retry failed API calls up to three times with model-specific backoff. Records that never yield a valid decision/errors block are marked as format failures and excluded from downstream metrics.