Confidence-Aware Tool Orchestration for Robust Video Understanding

Yangfan He[1 2] **Yujin Choi**[1 3] **Jaehong Yoon**[1*]

[1]NTU Singapore [2]University of Minnesota, Twin Cities [3]UNIST

Project Page: https://rova-v2.github.io/

Abstract

Video reasoning language models implicitly assume that every input frame is equally reliable. This leads to what we term the *Blind Trust Problem*: under realistic perturbations such as motion blur, glare, or occlusion, frontier video reasoning models can suffer 15-30%p accuracy drops on real-world embodied benchmarks, while remaining unaware that their visual evidence has been degraded. To address this challenge, we propose Robust-TO, an agentic video understanding framework that explicitly integrates per-frame trustworthiness into every stage of reasoning. Robust-TO organizes heterogeneous visual perception tools under a unified evidence interface. Each tool receives a sub-query derived from the original question and a set of trustworthy frames selected by the reliability-relevance score. It returns evidence in a shared format: a concrete prediction (e.g., a bounding box, motion trajectory, recognized text, or action label), temporal grounding, and a calibrated reliability score. During reasoning, these calibrated scores guide evidence weighting in a three-tier synthesis process (high/medium/low) and define a confidence-cost GRPO reward that jointly optimizes correctness, evidence reliability, and efficiency. On two video reasoning benchmarks spanning eight tasks, Robust-TO achieves 56.4% average accuracy on clean inputs, surpassing the strongest open-source baseline by 10.6%p and outperforming Gemini-2.5-Pro (46.2%). Under five realistic corruption types, Robust-TO maintains 54.3% average accuracy, 5.8%p above the strongest open-source baseline, while exhibiting the smallest clean-to-corrupted accuracy drop among all compared methods.

1 Introduction

When asked which car ran the red light in a video with a smudged windshield and motion blur, a careful observer does not respond immediately. They first decide which moments are worth trusting, lean on the clearest evidence, and re-examine ambiguous segments only if they remain uncertain. The decision proceeds in stages: first, a quick scan identifies interpretable segments; next, a focused examination inspects relevant details (e.g., faces, license plates, traffic-light state); and only if these are insufficient, the model deliberately revisits challenging frames with a clearer sense of what to seek. Contemporary video large language models (Video-LLMs) [10, 23, 3, 14] largely omit these processes. They typically rely on uniform frame sampling [3, 10, 23], encode the sampled frames through a vision backbone, and produce answers without explicitly evaluating whether the underlying visual evidence is sufficiently reliable to justify the prediction. This design collapses three fundamentally distinct capabilities into a one-step feed-forward prediction.

*Corresponding Author

Preprint.

We formalize this implicit assumption as the *Blind Trust Problem*: every frame is treated as equally informative, every perception output as equally reliable, and the model's confidence in its answer is decoupled from the visual conditions that produced it. Its cost is well documented and silent: recent benchmarks [11, 28, 1, 9] show frontier video reasoning models losing 15-30%p on UrbanVideo under common corruptions, while their self-reported confidence remains largely unchanged. Zhang et al. [29] highlight that scaling parameters and data alone do not necessarily lead to robustness. In safety-critical settings such as forensic video analysis, surveillance review, or post-hoc autonomous driving analysis, this silent failure is precisely the mode that must be eliminated.

To address this problem, we introduce Robust-TO, an agentic video understanding framework that performs robust reasoning on real-world videos through adaptive visual tool use guided by frame-wise reliability estimates. The pipeline proceeds in three stages (see Fig. 2): **(1) Frame Selection via Quality Profiling**: we design a parameter-free `assess_quality` tool (see Tab. 9) that characterizes each frame's degradation in terms of blur, brightness deviation, and occlusion, producing a disturbance profile. This profile captures both the *dominant* corruption type and its *severity*. Frames are then jointly scored by reliability and query relevance, filtering out corrupted yet query-relevant distractors, and retaining the top-K trustworthy frames for downstream perception. **(2) Confidence-Guided Tool Routing**: To obtain fine-grained perceptual evidence that addresses each distinct aspect of the query, we decompose the input query into atomic sub-queries and route each sub-query to the perception tool best suited to the dominant corruption observed in the selected frames. Every tool call returns a `(result, confidence)` pair, where confidence is computed as the product of the tool's intrinsic certainty and the estimated reliability of its input frames, thereby down-weighting tool outputs derived from degraded inputs during reasoning. **(3) Video Reasoning with Reliability-Aware Evidence**: `(result, confidence)` pairs collected across all sub-query tool calls are grouped into three reliability tiers (high/medium/low) to synthesize the final answer; high-tier evidence drives the conclusion, medium-tier one is retained only if consistent, and low-tier one is considered only when no stronger evidence is available, with residual uncertainty explicitly reported in the final answer.

The host VLM is trained end-to-end with Group Relative Policy Optimization (GRPO) [22]. The training reward combines four signals: (i) *correctness reward* that scores answer accuracy; (ii) *confidence-cost reward* that encourages high-confidence outputs while penalizing expensive tool calls on degraded frames; (iii) *sub-query efficiency reward* that penalizes both over- and under-decomposition of the query, anchored to a VLM-estimated target count; and (iv) *format reward*.

We demonstrate the effectiveness of the proposed Robust-TO on UrbanVideo-Bench [30] and VSI-Bench [26], spanning eight tasks under both clean and corrupted conditions generated via RoVA [9]. On clean benchmarks, Robust-TO with Qwen3-VL-7B achieves 56.4% average accuracy, surpassing Gemini-2.5-Pro (46.2%) and the supervised fine-tuned Qwen2.5-VL-7B (45.8%). On corrupted UrbanVideo-Bench, Robust-TO achieves an average accuracy of 54.3%, outperforming the strongest open-source baseline, Video-R1 [6], by 5.8%p and the best proprietary model, Gemini-2.5-Pro, by 16.2%p. Moreover, the adaptive key-frame selector reduces the average number of processed frames by 35% (from 32 to 20.7) and cuts per-sample inference time by over 35%, while simultaneously improving accuracy by 1.6%p (Tab. 4). These results highlight that the adaptive visual tool use in Robust-TO, guided by per-frame reliability estimates, benefits both corrupted and clean videos.

2 Related Work

Video Large Language Models. Recent Video-LLMs extend strong image-language pretraining to the video domain by explicitly modeling temporal information. LLaVA-OneVision [10] demonstrates that a single architecture can transfer image instruction tuning to video; InternVL [4], Qwen2/2.5-VL [23, 3] introduce dynamic FPS sampling, window attention, and absolute time embeddings to support hour-long inputs; earlier work such as Video-LLaVA [14] unifies image and video representations using a shared projector. These models achieve top results on Video-MME [7], MVBench [12], EgoSchema [16], and NExT-QA [25]. They all assume the visual signal is clean: frames are uniformly or densely sampled, and the pipeline never questions whether a given frame deserves attention. Robust-TO inherits this design as a limiting cas when frames are clean, it works exactly the same; when they are not, noise at the frame level alters the entire reasoning path.

Agentic and Tool-Augmented Video Reasoning. Researchers have also explored video understanding as iterative information acquisition. ReAct [27] and Toolformer [20] establish the paradigm

of interleaving reasoning traces with tool calls; ToolLLM [18] scales this idea to thousands of APIs. In the video domain, VideoAgent [24] uses an LLM controller with a CLIP retriever [19] and a captioner to iteratively select frames. A memory-augmented variant [5] adds temporal and object-level memory, while Graph-VideoAgent [15] maintains an explicit entity-relation graph. These systems show that selectively gathering evidence outperforms dense encoding, but their tool interfaces only report what was found, not how reliably. Concretely, when an LLM controller calls a CLIP retriever, it cannot tell whether a high-similarity frame is informative or whether both the query and the frame are overwhelmed by noise. Robust-TO closes this gap by providing explicit (output, confidence) pairs, treating input quality as a direct reasoning signal for the LLM rather than burying it inside retriever scores.

Reinforcement Learning for Multimodal Reasoning. Group Relative Policy Optimization (GRPO) was introduced in DeepSeekMath [22] as a memory-efficient alternative to PPO [21]. It eliminates the value network by estimating advantages from group-normalized rewards. DeepSeek-R1 [8] shows that rule-based GRPO can elicit reflection and verification without any SFT cold-start. In the video domain, Video-R1 [6] adds a temporal-order auxiliary reward; DeepVideo-R1 [17] reformulates GRPO as advantage regression with difficulty-aware augmentation; and VideoChat-R1 [13] applies GRPO to spatio-temporal grounding. These rewards, however, say nothing about how the answer was reached. Robust-TO augments the correctness signal with a confidence-cost term and a question-adaptive sub-query term. Both terms produce useful gradients even when the correctness signal is ambiguous, and both directly tie back to visual quality through the unified confidence.

3 Robust-TO: Robust Video Understanding with Tool Orchestration

3.1 Blind Trust Problem

Real-world videos are rarely pristine, as motion blur, glare, low-light noise, and occlusions frequently degrade visual quality. However, modern Video-LLMs typically process frames under the implicit assumption that they are equally reliable. This blind trust is harmful because degraded frames enter the reasoning process as evidence, and their corrupted visual signals can distort the final prediction.

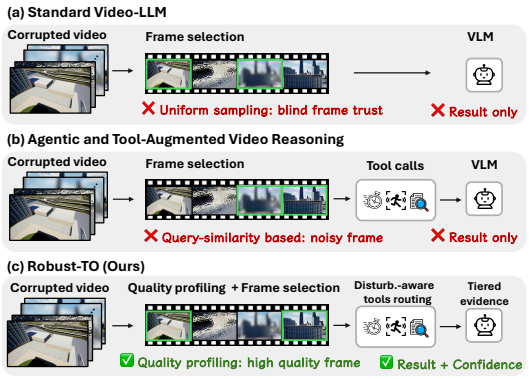


Figure 1: Comparison of video reasoning pipelines under corrupted video.

As shown in Fig. 1, existing methods either select frames uniformly or by query similarity alone, both of which can admit corrupted frames and lead to incorrect answers. Robust-TO addresses this by selecting clean frames based on both quality profiling and query similarity, even under corrupted video conditions. Moreover, it provides a confidence score alongside each answer, quantifying how much the supporting evidence can be trusted, which is particularly valuable when the input video is corrupted.

3.2 Problem Setting

Let $\mathcal{V} = \{f_1, \ldots, f_N\}$ denote a clean video of N frames. Motivated by real-world perturbations, we assume that the learner observes video streams $\tilde{\mathcal{V}} = \{\tilde{f}_1, \ldots, \tilde{f}_N\}$, where each frame may be corrupted by unknown degradations with some probability:

$$\tilde{f}_i = \mathcal{D}_i(f_i, \delta_i), \qquad \mathcal{D}_i : \mathcal{X} \times [0,1] \to \mathcal{X}, \qquad i = 1, \ldots, N, \tag{1}$$

with per-frame severity $\delta_i \in [0,1]$ (δ_i=0: no degradation; δ_i=1: complete information loss). Neither the corruption family $\mathcal{D}$ nor the severity schedule δ is known to the system in advance, and different frames may be affected by distinct disturbance types (e.g., motion blur, glare, or partial occlusion).

Given a text query q over $\tilde{\mathcal{V}}$, the goal is to produce a correct answer a while satisfying two requirements: *(i)* infer the trustworthiness of each frame from the observed pixels alone, without supervision;

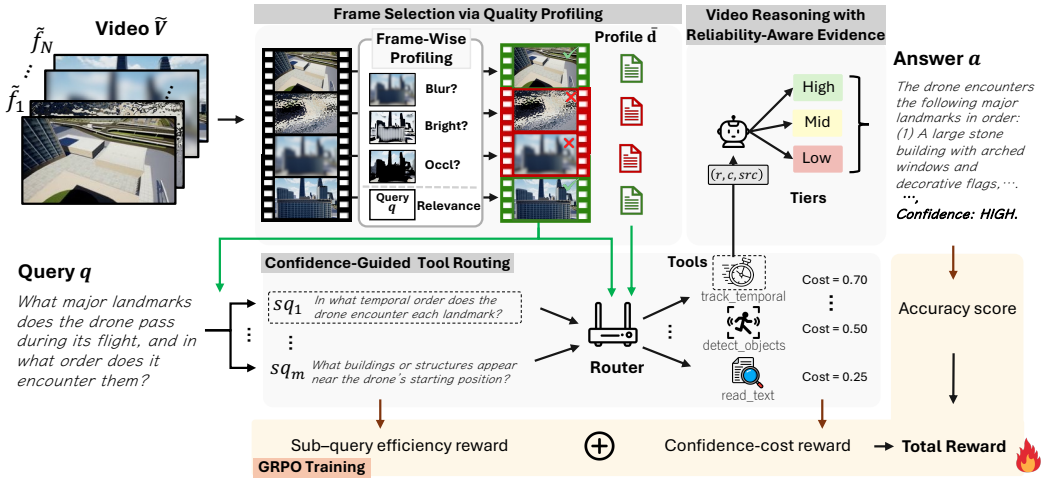


Figure 2: Overview of Robust-TO. Given a real-world video $\tilde{\mathcal{V}}$ and a query q, the host VLM first profiles each frame's quality, then selects the most reliable frames. It decomposes q into atomic sub-queries and routes each to the perception tool best matched to the dominant corruption. Finally, it predicts the answer by grouping evidence into reliability tiers. The host VLM is trained end-to-end with GRPO using a reward that combines accuracy, efficiency, and confidence-cost trade-off.

and *(ii)* condition evidence acquisition (i.e., the selection of input frames and the dispatch of perception tools that gather evidence from them) and reasoning on these inferred reliability signals, so that conclusions are never grounded in frames whose visual information has been severely degraded by corruption. When $\delta_i = 0$ for all i, the video is clean, and the pipeline becomes identical to standard video reasoning with no additional overhead. To achieve *adaptive* robustness under heterogeneous and unknown corruption, we propose Robust-TO, a novel framework that identifies and estimates frame-level degradations at inference time while minimizing their disruptive effects on downstream reasoning, illustrated in Fig. 2.

3.3 Frame Selection via Quality Profiling and Adaptive Tool Use for Video Reasoning

Frame Selection via Quality Profiling. Since both the corruption type and severity are unknown a priori (Eq. (1)) and can vary across frames, a vision language model (VLM) cannot reliably select trustworthy frames or appropriate tools without first assessing frame degradation. We therefore estimate the quality of each frame upfront, converting latent corruption into an explicit per-frame signal for downstream frame selection and tool routing. We define quality profiling as the process of estimating a per-frame disturbance score $d(f_i)$ (for simplicity, we use f_i to denote the observed frame $\tilde{f}_i$) alongside its component-wise degradation indicators to characterize the type and severity of corruption in each frame. We focus on blur, brightness deviation, and occlusion because they cover the dominant sources of visual unreliability in real-world video capture, including motion or defocus blur, glare, and under- or overexposure, and physical occlusion. Moreover, each can be estimated efficiently from signal-level statistics without training data or reference frames. Given the host VLM as a controller, we first invoke `assess_quality` (the parameter-free image-quality-assessment tool in our tool set; see Tab. 9) on each frame to compute

$$d(f_i) \;=\; \text{mean}\left(d_{\text{blur}}(f_i), d_{\text{bright}}(f_i),\, d_{\text{occl}}(f_i)\right), \tag{2}$$

where d_{blur} quantifies spatial sharpness via the inverse Laplacian variance (blurrier frames score higher), d_{bright} measures illumination distortion, assigning higher scores to frames that are either too dark or too bright, and d_{occl} captures the fraction of the frame lacking informative edge structure, estimated from Sobel-magnitude statistics. Please see detailed formulations in Sec. C. We min-max normalize each component across $\tilde{\mathcal{V}}$ so that blur, brightness, and occlusion contribute on a comparable scale. Beyond the aggregate disturbance score $d(f_i)$, the component-wise values $(d_{\text{blur}}, d_{\text{bright}}, d_{\text{occl}})$ indicate which degradation is most prominent in each frame. The controller can then use this disturbance profile to prefer frames and tools that are more reliable under the corresponding degradation.

Given the disturbance scores, the controller calls the `select_frames` tool, which ranks candidate frames using a reliability-relevance score. The tool assigns a score $s(f_i)$ to each candidate frame f_i:

$$s(f_i) = (1 - d(f_i)) \cdot \mathrm{sim}(\phi(f_i), \psi(q)), \quad f_i \in \mathcal{F}. \tag{3}$$

Here, $1 - d(f_i)$ estimates frame reliability, while $\mathrm{sim}(\phi(f_i), \psi(q))$ measures relevance to the query via cosine similarity. $\mathcal{F}$ denotes the set of valid frames whose reliability and query relevance both exceed their respective thresholds. Next, it returns the top-K remaining frames for downstream perception, where $K \in [4, 12]$ is adaptively chosen by the host VLM according to query complexity (See sensitivity analysis in Sec. 4.3). By prioritizing frames that are both visually trustworthy and query-relevant, this selection strategy provides a reliable foundation for robust downstream reasoning.

Confidence-Guided Tool Routing. After high-reliability frames are selected, the host VLM determines which perception tools should process them. Complex queries often require multiple perceptual skills. For example, answering whether an object moves after appearing in a scene requires both spatial localization and motion analysis. Since different perception tools specialize in different skills and exhibit different robustness to blur, illumination degradation, and occlusion, the controller first decomposes the query q into atomic sub-queries $\{sq_1, \ldots, sq_m\}$ conditioned on the original question and the selected frames. Each sub-query targets a single perceptual primitive, such as object localization, motion analysis, OCR, or attribute recognition, and is generated through in-context instructions without a dedicated parser.

Routing then follows a two-stage *plug-and-play* protocol that is agnostic to the specific tool library. First, the semantic type of each sub-query determines the candidate tools. For example, spatial queries are matched to `detect_objects`, while temporal queries are matched to `track_temporal` or `recognize_action` (see Tab. 9). Second, the averaged disturbance profile $\bar{\mathbf{d}} = (\bar{d}_{\mathrm{blur}}, \bar{d}_{\mathrm{bright}}, \bar{d}_{\mathrm{occl}})$ selects the most reliable candidate by identifying the dominant corruption type: when blur is predominant, the controller favors `caption_frame`, which is more tolerant to spatial degradation, over `detect_objects`, which depends on sharp boundaries; when brightness distortion dominates, it prioritizes tools that are more robust to extreme illumination.

Let $\mathbf{F}_j \subseteq \mathcal{F}$ denote the set of selected high-reliability frames provided to the j-th tool call for sub-query sq, with j indexing tool calls globally. Each call returns a result-confidence pair (r_j, c_j) in a shared output format, where r_j is the tool's perception output (e.g., a bounding box, recognized text, action label, or caption) and c_j is the confidence score associated with that output.

$$(r_j, c_j) = T_j(\mathbf{F}_j, sq), \qquad c_j \;=\; \underbrace{c_j^{\mathrm{intrinsic}}}_{\text{tool self-assessment}} \;\times\; \underbrace{\rho(\mathbf{F}_j)}_{\text{input reliability}}, \tag{4}$$

where $c_j^{\mathrm{intrinsic}} \in [0, 1]$ is a self-assessment returned by the tool itself (e.g., the mean detection score or token log-probability; see Tab. 9), and $\rho(\mathbf{F}_j)$ is the input reliability, defined as a conservative *mean* of the per-frame reliabilities $1 - d(f)$:

$$\rho(\mathbf{F}_j) \;=\; \frac{1}{\lceil n/3 \rceil} \sum_{f \in \mathbf{F}_{j,\mathrm{lowest}}} \big(1 - d(f)\big), \qquad n = |\mathbf{F}_j|, \tag{5}$$

where $\mathbf{F}_{j,\mathrm{lowest}}$ is the subset of $\lceil n/3 \rceil$ frames in $\mathbf{F}_j$ with the lowest reliability. This penalizes tool inputs that contain severely corrupted frames and prevents a small number of clean frames from masking unreliable visual evidence. Multiplying the tool confidence c_j by the frame reliability $\rho(\mathbf{F}_j)$ allows the model to avoid overconfident reasoning based on unreliable visual evidence. We define each evidence item as a tuple $(r_j, c_j, \mathbf{F}_j, \{d(f)\}_{f \in \mathbf{F}_j})$, where r_j is the perception-tool output, c_j is its confidence score defined in Eq. (4), $\mathbf{F}_j$ is the set of source frames used by the j-th tool call, and $\{d(f)\}_{f \in \mathbf{F}_j}$ are the corresponding disturbance scores. Thus, each evidence item jointly records the tool output, its calibrated confidence, the source frames used to produce it, and the corresponding disturbance scores. Thus, each evidence item fully tracks the tool output, its confidence, the identity of its source frames, and their disturbance scores. The complete evidence set accumulated over all sub-query tool calls is then given by $\mathcal{I} = \{(r_j, c_j, \mathbf{F}_j, \{d(f)\}_{f \in \mathbf{F}_j})\}_j$, which serves as the basis for the reliability-aware video reasoning stage described later.

Video Reasoning with Reliability-Aware Evidence. After all sub-queries are processed, the host VLM produces the final answer a by integrating $\mathcal{I}$, the complete evidence set gathered across all

sub-query tool calls. Whereas each tool call yields only a local result r_j, the answer a is produced only at this stage, after reliability-aware integration over the full evidence set. Here, we note that duplicate tool calls may occur across sub-queries. We disambiguate them with a unique global index j for each tool call, while each evidence item records its source sub-query sq_k and invoked tool T_j.

Based on these signals, the host VLM groups evidence into three reliability tiers: *high*, *medium*, and *low*. It first infers a preliminary conclusion from high-reliability evidence (*high*), then evaluates each medium-reliability item (*medium*) against this conclusion through in-context reasoning. Medium-reliability evidence is retained only when it agrees with the conclusion, while evidence that points to a different answer or reports an inconsistent attribute is discarded as unreliable. Low-reliability evidence *(low)* is used only as a fallback when no high-reliability evidence is available, and the answer explicitly marks the remaining uncertainty. Thus, reliable evidence determines the prediction, and uncertain evidence can support but cannot change the conclusion. On clean videos, the reliability criterion naturally admits nearly all evidence, so the method reasons over the full evidence set.

3.4 Confidence-Cost Trade-off Reward for GRPO Training

We train the host VLM with GRPO to perform confidence-guided routing and evidence integration, using a reward that balances confidence and efficiency. The core intuition is that reliable evidence (high confidence) is valuable, but obtaining it may require expensive tools (high cost). The reward encourages the model to seek high-confidence outputs while penalizing unnecessary tool expenditures, thus naturally balancing the confidence-cost trade-off. We compute all reward components on a shared single-rollout trajectory τ, defined over the full episode from frame selection, sub-query decomposition, and tool routing to final evidence integration.

Confidence-Cost Reward. For each tool T_j called on sub-query sq with output (r_j, c_j), we define:

$$R_{\text{cc}}(c_j, T_j) \;=\; c_j \;-\; \lambda \cdot \text{cost}(T_j), \tag{6}$$

where $\text{cost}(T_j)$ denotes the tool cost, defined as the wall-clock runtime normalized by the runtime of `caption_frame` (see Tab. 9). For a trajectory τ with N_{call} tool calls, the total reward averages the per-call rewards:

$$R_{\text{cc}}^{\text{total}}(\tau) \;=\; \frac{1}{N_{\text{call}}} \sum_{k=1}^{N_{\text{call}}} R_{\text{cc}}\big(c_{j_k},\, T_{j_k}\big). \tag{7}$$

Here, j_k denotes the index of k-th tool call, and c_{j_k} is the confidence returned by tool T_{j_k}.

Sub-Query Efficiency Reward. Let m^* be the question-dependent optimal number of sub-queries. We want the VLM to decompose q into the right number of sub-queries m, avoiding both information gaps ($m < m^*$) and wasteful calls ($m > m^*$). We estimate m^* using a separate, frozen off-the-shelf VLM π_{est} (not the policy VLM itself) once per question; π_{est} is a pretrained VLM that we prompt to predict the minimal number of sub-queries needed to answer q reliably. Here, $R_{\text{min-sq}}$ discourages excessive sub-query decomposition, while $R_{\text{qual}}(\tau)$ rewards trajectories that use a sufficient number of tools by computing the average tool confidence over the same trajectory τ. The sub-query efficiency reward R_{subq} is then defined as the sum of these two terms, $R_{\text{subq}} = R_{\text{min-sq}} + R_{\text{qual}}(\tau)$.

Total Reward. With rewards defined for tool-use efficiency (R_{cc}) and for the overall decomposition strategy (R_{subq}), we now combine them with correctness and format signals into a single training objective. The format reward $R_{\text{fmt}} \in \{0, 1\}$ indicates whether the model's output follows the required structure (e.g., JSON format for tool calls or final answers), and $R_{\text{acc}} \in \{-1, +1\}$ indicates whether the final answer is correct. The four terms above are combined into a single scalar reward for each trajectory:

$$R_{\text{total}} = R_{\text{acc}} + w\left(R_{\text{subq}} + R_{\text{cc}}^{\text{total}} + R_{\text{fmt}}\right), \tag{8}$$

where $w = 1/3$, and other auxiliary terms are bounded.

Table 1: Performance on 8 indoor and outdoor embodied spatial reasoning tasks. The baselines include popular proprietary, open-source multimodal reasoning models, video LLMs, and models fine-tuned on the same training data. Tasks include: LP (Landmark Position), CF (Counterfactual), PE (Progress Evaluation), AG (Action Generation), RDist (Relative Distance), RDir (Relative Direction), RP (Route Planning), AO (Appearance Order).

Method	Frames	Avg.	UrbanVideo-Bench				VSI-Bench			
			LP	*CF*	*PE*	*AG*	*RDist*	*RDir*	*RP*	*AO*
Proprietary Models (API)										
Qwen-VL-Max	32	35.2	44.8	49.2	38.8	29.6	28.0	33.3	29.6	28.3
GPT-4o	32	36.0	36.8	44.7	34.2	33.8	37.0	41.3	31.5	28.5
Gemini-1.5-Flash	1fps	38.2	37.8	42.4	43.3	34.4	37.7	41.0	31.5	37.8
Gemini-1.5-Pro	1fps	40.3	37.4	46.2	38.8	31.9	51.3	46.3	36.0	34.6
SOTA Reasoning Models (API)										
OpenAI-o1	32	40.4	34.6	53.3	39.1	28.0	39.7	35.8	**52.9**	39.8
Gemini-2.5-Pro	1fps	46.2	40.0	**75.0**	38.7	23.5	42.0	34.5	52.4	63.6
Open-source Models										
LLaVA-NeXT-Video-7B-hf	32	29.0	49.5	20.5	36.6	19.2	25.2	26.3	29.9	24.5
Phi-3.5-vision-instruct	32	30.9	49.2	34.8	33.2	15.6	25.4	26.5	36.9	25.2
Kangaroo	32	30.4	35.5	42.4	32.5	32.4	25.2	26.8	23.5	24.9
InternVL2-2B	32	27.7	19.3	45.5	29.2	20.9	25.1	25.0	32.6	23.9
InternVL2-8B	32	28.1	23.1	45.5	31.5	21.4	24.7	25.7	28.3	24.8
InternVL2-40B	32	28.0	23.2	41.7	32.4	22.3	24.9	25.7	29.4	24.5
Qwen2.5-VL-3B-Instruct	32	35.4	32.1	47.8	34.0	31.0	27.9	32.6	39.0	38.9
Qwen2.5-VL-7B-Instruct	32	33.9	33.3	21.7	25.0	27.8	35.8	39.7	48.8	38.8
Qwen2.5-VL-72B-Instruct	32	35.0	34.7	34.8	26.4	37.7	40.8	29.0	32.5	43.9
Supervised Fine-Tuning										
Qwen2.5-VL-3B-Instruct	32	40.6	47.7	33.4	34.8	39.2	42.6	42.3	41.2	43.9
Qwen2.5-VL-7B-Instruct	32	45.8	40.2	53.4	38.0	40.8	47.8	46.3	44.1	56.1
Robust-TO (Ours)										
Robust-TO (Qwen2.5-VL-7B-Instruct)	20.7	50.7	55.1	59.9	39.7	47.6	50.0	44.3	36.8	72.0
Robust-TO (Qwen3-VL-7B-Instruct)	20.7	**56.4**	**61.1**	64.4	**44.7**	**59.0**	**55.5**	**48.8**	39.8	**77.5**

Table 2: Accuracy on UV-Bench under each of the five PVRBench [9] corruption masks (MB: Motion Blur, GN: Gaussian Noise, GL: Glare, Occ: Occlusion, LL: Low-Light) for selected models. Numbers are averaged across the four UV-Bench tasks (LP, CF, PE, AG).

Method	Frames	Clean	MB	GN	GL	Occ	LL	Avg
GPT-4o	32	37.4	32.2	31.7	32.5	30.8	33.6	32.2
Gemini-2.5-Pro	1fps	44.3	37.7	37.7	38.8	36.4	39.8	38.1
Qwen2.5-VL-7B-Instruct	32	26.9	17.0	16.6	18.9	14.6	20.3	17.5
Qwen2.5-VL-72B-Instruct	32	33.4	26.3	26.0	27.2	24.3	28.7	26.5
Qwen2.5-VL-7B-Instruct (SFT)	32	39.0	30.9	30.7	32.1	29.2	33.3	31.2
Video-R1 (Qwen2.5-VL-7B-Instruct)	32	43.0	38.5	38.0	39.5	37.0	40.5	38.7
Video-R1 (Qwen3-VL-7B-Instruct)	32	52.0	48.5	48.0	49.0	47.5	49.5	48.5
Robust-TO (Ours) (Qwen2.5-VL-7B-Instruct)	20.7	50.6	47.0	46.5	47.7	46.1	48.3	47.1
Robust-TO (Ours) (Qwen3-VL-7B-Instruct)	20.7	**57.3**	**54.1**	**54.0**	**54.9**	**53.5**	**55.1**	**54.3**

4 Experiments

4.1 Experiment Setup

We evaluate Robust-TO under realistic video corruptions, including motion blur, glare, and occlusion. We use RoVA [9] to generate degraded variants of UrbanVideo-Bench (UV-Bench) [30] and VSI-Bench [26], enabling controlled assessment of robustness across both clean and corrupted inputs. We instantiate Robust-TO as a training framework on two base models: Qwen2.5-VL-7B [3], and Qwen3-VL-7B [2]. Training is conducted on the video subset of the Video-R1 dataset, which covers both indoor and outdoor scenes, using GRPO with $4\times$A100 GPUs (rollout group size of 16 for $\sim$5k steps). To prevent the policy model from exploiting the sub-query count during training, we estimate the optimal number of sub-queries using a frozen Qwen2.5-VL-7B-Instruct model.

Table 3: Paradigm ablation on UV-Bench (Qwen3-VL-7B). Each row adds one component. P→C+SQ: Perception-to-Contemplate with Sub-Query Decomposition; Contemplate is the reasoning step (Secs. F.1 and F.3).

Reasoning Paradigm	Avg.	LP	CF	PE	AG
Direct (R1)	39.5	42.1	44.8	33.6	37.5
P→C+SQ	42.8	45.7	48.4	35.9	41.2
+Tool	49.4	52.8	55.9	39.8	49.1
+Conf	52.6	56.0	59.4	41.9	53.1
+GRPO	**57.3**	**61.1**	**64.4**	**44.7**	**59.0**

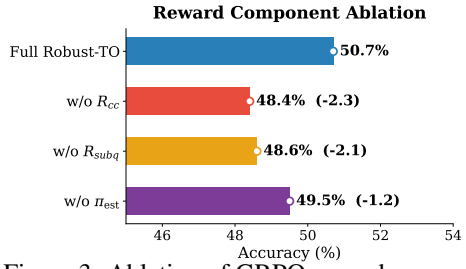


Figure 3: Ablation of GRPO reward components on UV-Bench (Qwen2.5-VL-7B). The frozen estimator π_{est} predicts the target sub-query count m^* using a VLM.

4.2 Main Results

Results on Clean Dataset. We first evaluate Robust-TO on two challenging video reasoning benchmarks, UV-Bench [30] and VSI-Bench [26], covering outdoor and indoor scenes. As shown in Tab. 1, with Qwen3-VL-7B and Qwen2.5-VL-7B backbones, Robust-TO achieves 56.4% and 50.7% average accuracy, respectively, outperforming GPT-4o and the corresponding base models. It achieves the best performance on 6 out of 8 tasks, with the largest gains on temporally extended tasks such as Appearance Order and Landmark Position, where reasoning requires integrating evidence from separated frames. These results show that Robust-TO improves clean-video reasoning by selecting reliable, query-relevant frames and integrating tool-assisted evidence with calibrated confidence.

Results on Disturbed Dataset. We next evaluate robustness under corrupted videos using RoVA-degraded UV-Bench, which includes five corruption types across four tasks. Robust-TO with Qwen3-VL-7B achieves 54.3% average accuracy, substantially outperforming the best open-source baseline Video-R1 (48.5%), the best proprietary model Gemini-2.5-Pro (38.1%), and GPT-4o (32.2%). Robust-TO leads on every corruption type, with particularly large gains under Occlusion and Glare. Using the same Qwen2.5-VL-7B backbone, Robust-TO also significantly improves over Video-R1 and even surpasses the much larger 72B model. Moreover, Robust-TO exhibits the smallest drop from clean to corrupted data among all methods, indicating graceful degradation under visual disturbance. With GRPO, the full Robust-TO pipeline reaches 57.3% on the disturbed UV-Bench, while the key-frame extractor reduces the number of processed frames and inference time by 35% with a +1.6 point accuracy gain. These results validate the central design of Robust-TO: coupling confidence with frame quality and filtering unreliable visual evidence prevents corrupted frames from dominating downstream reasoning.

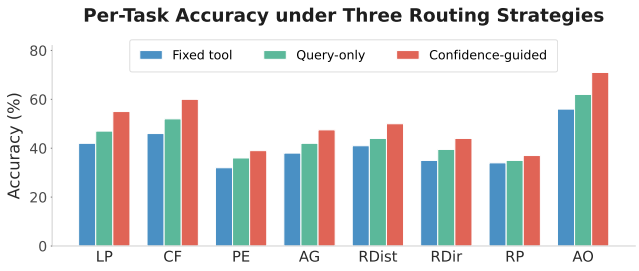


Figure 4: Ablation of Tool Routing. Per-task accuracy under three routing strategies: fixed tool, query-only, and the full confidence-guided policy.

4.3 Ablation Studies

In this section, we ablate each architectural decision in Robust-TO to isolate its contribution. Unless stated otherwise, all experiments use the Qwen2.5-VL-7B-Instruct with Robust-TO configuration and report average accuracy over the same 8-task benchmark.

Ablation of Reasoning Paradigm (Tab. 3). We incrementally add Robust-TO's components to a vanilla R1-style controller on Qwen3-VL (UV-Bench: LP, CF, PE, AG). Separating evidence collection from answer generation via a "perception → contemplate → answer" structure yields +3.3%p. Sub-query decomposition adds +3.1%p, and binding sub-queries to visual tools adds +3.5%p. Adding confidence scores alone (no RL) improves over the tool-only variant by +3.2%p, showing the unified (r_j, c_j) contract provides inference-time utility. GRPO with the confidence-cost reward

Table 4: Key-frame extractor ablation in Robust-TO on UV-Bench with Qwen2.5-VL-7B-Instruct.

Method	Frames	Acc. (%)	Train (h)	Infer (s)
w/o KFE	32	49.1	127.87	243.68
w/ KFE	$20.7^{\downarrow 11.3}$	$50.7^{\uparrow 1.6}$	$111.70^{\downarrow 16.17}$	$157.55^{\downarrow 86.13}$

Table 5: Ablation of collaborative reasoning on eight UV-Bench and VSI-Bench tasks. w/o and w denote without and with collaboration.

	Avg.	LP	CF	PE	AG	RDist	RDir	RP	AO
w/o	36.0	31.8	45.7	28.3	28.1	41.0	29.7	37.5	46.0
w	50.7	55.1	59.9	39.7	47.6	50.0	44.3	36.8	72.0

Table 6: Ablation of Confidence-Reporting Interface on 8 tasks (UV-Bench: LP, CF, PE, AG; VSI-Bench: RDist, RDir, RP, AO), Second best underlined.

Configuration	**Avg.**	**UV-Bench**				**VSI-Bench**			
		LP	**CF**	**PE**	**AG**	**RDist**	**RDir**	**RP**	**AO**
Remove $\rho(\mathbf{F}_j)$ (intrinsic only)	43.1	46.3	51.2	<u>36.0</u>	39.5	41.5	36.5	32.2	61.5
Uniform mean (vs worst-K)	<u>47.4</u>	<u>51.5</u>	<u>56.5</u>	35.5	<u>44.5</u>	<u>46.5</u>	<u>41.3</u>	<u>34.7</u>	<u>68.7</u>
Full Robust-TO (worst-K + ρ)	**50.7**	**55.1**	**59.9**	**39.7**	**47.6**	**50.0**	**44.3**	**36.8**	**72.0**

contributes the largest gain (+4.7%p), confirming the reward signal is genuinely learnable, not a prompting artifact.

Ablation of GRPO Reward Design (Fig. 3). We test three variations of our reward design. (i) Removing the confidence-cost term R_{cc} hurts accuracy by 2.3 points: without it, the agent always picks the most expensive tool, leading to confident but wrong answers on corrupted videos. (ii) Removing R_{subq} loses 2.1 points: the agent splits questions into too many tiny pieces, slowing down the model without improving results. (iii) Replacing the frozen m^* estimator with one that the model learns itself loses 1.2 points and makes the reward 2.3 times more unstable as the model learns to cheat by manipulating its own estimator, which is why we freeze it.

Ablation of Confidence-Guided Tool Routing (Fig. 4). We compare three routing strategies: fixed tool (always the same), query-only (semantic only), and our confidence-guided policy. The full policy outperforms query-only by +6.1%p and fixed-tool by +10.4%p on average. Gains are task-dependent: +9.6%p on Appearance Order (blur favors captioning over detection) vs. +1.7%p on Route Planning (spatial coverage matters more). This confirms the policy learns a non-trivial mapping from corruption profiles to tool effectiveness, not a universal preference.

Ablation of Frame Selection and Collaborative Synthesis (Tabs. 4 and 5). As demonstrated in Tab. 4, the key-frame extractor reduces the average frame count from 32 to 20.7 while improving accuracy from 49.1% to 50.7% on UV-Bench. Training time decreases by over 16 hours, and inference time per sample drops by more than 86 seconds, showing that discarding unreliable frames enhances both robustness and efficiency. Tab. 5 further highlights the benefit of full collaboration: compared to the baseline without collaboration, the complete Robust-TO pipeline raises average accuracy from 36.0% to 50.7%.

The largest improvements occur on tasks requiring integrated evidence, such as Appearance Order, which rises from 46.0% to 72.0%, and Landmark Position, which jumps from 31.8% to 55.1%. These results confirm that jointly synthesizing reliability-weighted evidence across sub-queries is essential for handling realistic video corruptions.

Ablation of Confidence-Reporting Interface (Tab. 6). The unified (r_j, c_j) contract has two key designs: multiplying tool certainty by input quality $\rho(\mathbf{F}_j)$ (Eq. (4)), and aggregating ρ via worst-K frames. Removing $\rho(\mathbf{F}_j)$ causes the largest drop, from 50.7% to 43.1% (a loss of 7.6%p), showing the controller cannot distinguish clean from corrupted frames without quality coupling. Replacing worst-K with uniform mean gives a smaller drop of 3.3%p, as a single clean frame masks other corruptions and inflates ρ, leading to over-trust in degraded evidence.

4.4 Case Study

We also provide a case study Tab. 7 with the query *"Which vehicle ran the red light at the intersection, and what was its license plate number?"*. The source video has 24 frames under three concurrent

Table 7: Case study of ROBUST-TO on corrupted dashcam footage (UV-Bench).
Question: *Which vehicle ran the red light at the intersection, and what was its license plate number?*

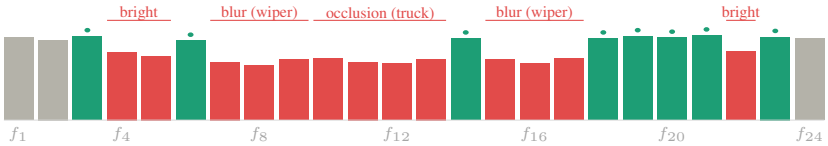
STANDARD VIDEO-LLM

Reasoning	The video shows an intersection with traffic lights. Around frame 15-20, the light turns red. A white sedan approaches the intersection from the south. Its brake lights do not illuminate, suggesting it does not slow down. The sedan crosses the stop line while the light is red. The license plate appears partially obscured by motion blur, but the visible characters are "B8C-394". No other vehicle enters the intersection during the red phase.
Generated Answer	**The white sedan** ran the red light, license plate **B8C-394**.

ROBUST-TO (OURS)

Quality Profiling + Frame Selection

Bar height: $1 - d(f_i)$ (reliability)



• Selected (Top-8 selected by $s(f_i)$) • Not selected • Excluded (Corrupted)
Selected frames: $[f_{14}, f_{18}, f_{19}, f_3, f_6, f_{20}, f_{21}, f_{23}]$
Note: f_{10}-f_{13} show high query similarity (0.73-0.80) but are excluded (d_{occl}=0.68-0.85).

Tool Outputs ***(result, conf)***

Sub-query	Tool Call	Result	Conf.	Tier
sq_1: *Detect traffic lights and their displayed signal color*	`detect_objects` (f_{14}, f_{18}, f_{19})	f_{14} :"...red light ...", f_{18} : "... red lamp ...", f_{19} : "Red signal ..."	0.759	`HIGH`
sq_2: *Detect all vehicles near the intersection and their bounding box positions*	`detect_objects` (f_3, f_6, f_{14}, f_{18}, f_{19})	White sedan (moving); dark SUV (static)	0.785	`HIGH`
sq_3: *Track the movement trajectory of each detected vehicle across consecutive frames*	`retrieve_frames` (f_{10}, f_{13}, f_9: degraded) → `track_temporal` (f_3, f_6, f_9, f_{10} ...)	Sedan: { bbox start: [120, 250, 210, 330], bbox end: [510, 170, 640, 265], ... }, SUV: { ... }	0.528	`MED`
sq_4: *Read the license plate text of the vehicle closest to the intersection*	`read_text` (f_{14}, f_{18}, f_{19})	f_{14}, f_{18} :B-7742-XK, f_{19} : B-77?2-XK →Ans: **B-7742-XK** (2/3 frames agree)	0.787	`HIGH`

Generated Answer	**A white sedan** ran the red light at the intersection. The license plate is **B-7742-XK**, confirmed by OCR on clean frames f_{14} and f_{18} (char-level confidence $\geq$0.89). Vehicle color and motion verified via detection and tracking (3 high-tier, 1 medium-tier evidence; no contradictions; overall confidence: 0.72).
Ground Truth	**A white sedan** (license plate: **B-7742-XK**) ran the red light.

corruptions: rainy evening with oncoming headlight glare (f_4, f_5, f_{22}), windshield wiper motion blur (f_7-f_9, f_{15}-f_{17}), and a truck partially occluding the intersection (f_{10}-f_{13}).

A standard Video-LLM processes all 24 frames uniformly, even when corrupted, whereas Robust-TO estimates per-frame quality and selects only high-reliability frames. For example, the standard model incorrectly predicts "white sedan, license plate B8C-394" as its prediction is influenced by motion-blurred and occluded frames (f_{10}-f_{13}), where a truck blocks the view and glare distorts the license-plate region. In contrast, Robust-TO computes per-frame disturbance scores $d(f_i)$ (Eq. (2)) and assigns high occlusion scores to f_{10}-f_{13} ($d_{\text{occl}} > 0.68$), excluding them from reliable evidence aggregation. Clean frames such as f_{14}, f_{18}, and f_{19} have low disturbance scores ($d < 0.3$) and are selected instead. On these selected frames, `read_text` returns "B-7742-XK" with high confidence ($c_j = 0.787$), while `detect_objects` and `track_temporal` verify that a white sedan passes through the red light. As the high-tier evidence consistently supports this interpretation and degraded low-confidence evidence is discarded, Robust-TO outputs the correct answer: "white sedan, license

plate B-7742-XK". Additional case studies spanning diverse task types and corruption profiles are provided in Sec. D.3.

5 Conclusion

We identify the *Blind Trust Problem*: Video-LLMs silently lose significant accuracy under realistic corruptions while self-reported confidence remains unchanged. To address this, we introduce Robust-TO with three components: a unified `(result, confidence)` interface that couples tool certainty with a parameter-free disturbance estimate; a quality profiling pipeline that ranks frames by reliability$\times$relevance, confidence-guided routes sub-queries to corruption-matched tools, and synthesizes evidence through three reliability tiers; and a confidence-cost GRPO reward with a frozen-estimator sub-query efficiency term that jointly optimizes accuracy, reliability, and parsimony. Robust-TO substantially outperforms the strongest open-source baseline across multiple benchmarks and tasks, achieves high clean accuracy with minimal clean-to-corrupted drop, all within low latency overhead. Ablations confirm the contributions of quality coupling, worst-K aggregation, confidence-cost reward, and sub-query efficiency. Limitations include a disturbance vocabulary restricted to blur, brightness, and occlusion, and decomposition quality bounded by the frozen m^* estimator. We release code and checkpoints to support future extensions toward video reasoning that degrade gracefully rather than silently in the open world.

References

[1] Amit Agarwal, Srikant Panda, Angeline Charles, Hitesh Laxmichand Patel, Bhargava Kumar, Priyaranjan Pattnayak, Taki Hasan Rafi, Tejaswini Kumar, Hansa Meghwani, Karan Gupta, et al. Mvtamperbench: Evaluating robustness of vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18804–18828, 2025.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] 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] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pages 75–92. Springer, 2024.

[6] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *Advances in Neural Information Processing Systems*, 38:99114–99137, 2026.

[7] Chaoyou Fu, Yuhan Dai, Yongdong 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24108–24118, 2025.

[8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[9] Yangfan He, Changgyu Boo, and Jaehong Yoon. Are video reasoning models ready to go outside? *arXiv preprint arXiv:2603.10652*, 2026.

[10] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[11] Chunyi Li, Jianbo Zhang, Zicheng Zhang, Haoning Wu, Yuan Tian, Wei Sun, Guo Lu, Xiongkuo Min, Xiaohong Liu, Weisi Lin, et al. R-bench: Are your large multimodal model robust to real-world corruptions? *IEEE Journal of Selected Topics in Signal Processing*, 2025.

[12] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.

[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.

[14] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 5971–5984, 2024.

[15] Meng Liu, Mingda Wang, Xueyang Hu, Shengbo Wang, Yunchao Yin, Xiaolin Hu, Bing Zhao, and Cewu Lu. Understanding long videos via llm-powered entity relation graphs. *arXiv preprint arXiv:2501.15953*, 2025.

[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.

[17] Jinyoung Park, Jeehye Na, Jinyoung Kim, and Hyunwoo J Kim. Deepvideo-r1: Video reinforcement fine-tuning via difficulty-aware regressive grpo. *Advances in Neural Information Processing Systems*, 38:138605–138632, 2026.

[18] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, volume 2024, pages 9695–9717, 2024.

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[20] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

[21] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[22] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[23] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[24] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76. Springer, 2024.

[25] 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.

[26] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643, 2025.

[27] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

[28] Jiawei Zhang, Tianyu Pang, Chao Du, Yi Ren, Bo Li, and Min Lin. Benchmarking large multimodal models against common corruptions. *arXiv preprint arXiv:2401.11943*, 2024.

[29] Yichi Zhang, Yao Huang, Yitong Sun, Chang Liu, Zhe Zhao, Zhengwei Fang, Yifan Wang, Huanran Chen, Xiao Yang, Xingxing Wei, et al. Multitrust: A comprehensive benchmark towards trustworthy multimodal large language models. *Advances in Neural Information Processing Systems*, 37:49279–49383, 2024.

[30] Baining Zhao, Jianjie Fang, Zichao Dai, Ziyou Wang, Jirong Zha, Weichen Zhang, Chen Gao, Yue Wang, Jinqiang Cui, Xinlei Chen, et al. Urbanvideo-bench: Benchmarking vision-language models on embodied intelligence with video data in urban spaces. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32400–32423, 2025.

Appendix

A Limitations and Broader Impact

Limitations. We identify four limitations of the current framework. *(i) Disturbance vocabulary.* Eq. (2) covers blur, brightness deviation, and occlusion - the corruption types most prevalent in real-world capture. Adversarial perturbations, semantic occlusions (e.g., a relevant object hidden behind a clean but irrelevant foreground), and audio - visual misalignment fall outside its scope. Extending the disturbance profile to additional channels is straightforward under the plug-and-play interface but is left to future work. *(ii) Frozen estimator* π_{est}. Decoupling m^* estimation from the trained policy prevents reward gaming ... caps decomposition quality at π_{est}'s capability. In domains where the frozen π_{est} poorly judges query complexity, R_{subq} may provide a noisy training signal. *(iii) Encoder dependence.* The frame selection score (Eq. (3)) relies on the host VLM's own visual encoder ϕ. Although the multiplicative reliability factor suppresses severely degraded frames regardless of encoder quality, the informativeness ranking among surviving candidates may become noisy if ϕ is miscalibrated under corruption types unseen during pretraining. *(iv) Inference cost under heavy corruption.* On heavily corrupted inputs, the full pipeline - profiling, multi-tool routing, and confidence-weighted synthesis - can increase latency beyond the $<$5% overhead observed on clean videos. While the hierarchical design ensures this cost is incurred only when warranted, real-time deployment scenarios may require capping the maximum number of tool calls.

Broader Impact. Robust-TO targets safety-relevant applications such as forensic video analysis, surveillance review, and post-hoc autonomous driving analysis, where silent failures under degraded visual conditions carry significant consequences. By making per-frame reliability an explicit, interpretable signal, the framework enables downstream users to understand *why* a conclusion was reached and *how trustworthy* the underlying evidence is, promoting more accountable video reasoning systems. We acknowledge the dual-use nature of the proposed techniques. The per-frame disturbance scores could in principle be repurposed to detect – or conversely, to craft – tampered footage that evades quality-based gating. We view the detection capability as a net positive for content integrity, but note that an adversary could design perturbations that spoof the disturbance signal (i.e., appear clean to Eq. (2) while being semantically corrupted). Studying adversarial robustness of the disturbance estimator itself is an important direction not addressed in this work. We release code and checkpoints to facilitate community scrutiny and responsible extension.

B Details of Parameter Setting

This section consolidates all hyperparameters of Robust-TO into a single reference organized by pipeline stage, with Tab. 8 as the master index (detailed equations in Sec. 3.3 and sensitivity analyses in Secs. D.2 and 4.3). The three disturbance channels in Eq. (2) are parameter-free up to three signal-level constants (τ_{blur}=500, μ_{ref}=0.5, τ_{edge}=30) calibrated once on a small held-out pool; min-max normalization across $\tilde{\mathcal{V}}$ removes per-video scale before equal-weight combination, enabling transfer to unseen corruptions without re-calibration. The selection score $s(f_i)$ in Eq. (3) uses $(\theta_{\text{rel}}, \theta_{\text{sim}})$=$(0.55, 0.30)$; K is chosen adaptively by the host VLM in $[4, 12]$ (averaging 7.8 on clean inputs, 6.2 on corrupted inputs). The confidence formula Eq. (4) couples $c_j^{\text{intrinsic}}$ with $\rho(\mathbf{F})$ via worst-K mean (K=$\lceil n/3 \rceil$), clipping intrinsic confidences to $[0.01, 1.0]$ to prevent zero-multiplication, and uses a synthesis prompt that groups evidence into *high* ($c_j \geq 0.7$ *and* $d < 0.3$), *low* ($c_j < 0.3$ *or* $d \geq 0.7$), and *medium* tiers. The GRPO reward parameters are λ=0.5 (tool-cost weight), α=0.2, β=1.0; total reward follows R_{total}=$R_{\text{acc}} + w(R_{\text{subq}} + R_{\text{cc}}^{\text{total}} + R_{\text{fmt}})$ with w=1/3 so the auxiliary sum's magnitude is at most 1 and $R_{\text{acc}} \in \{-1, +1\}$ controls the sign; failed tool calls receive c_j=0; m^* is estimated by π_{est}, a frozen text-only Qwen2.5-7B-Instruct (policy-internal estimation costs 1.2%p and increases reward variance 2.3×). Training uses 4×A100-80GB, DeepSpeed ZeRO-2+FSDP, peak LR 1×10^{-6} with 200-step cosine warmup, rollout group size 16, per-GPU batch 2 with gradient accumulation 4, KL penalty 0.01, and $\approx 5{,}000$ steps, identical for Qwen2.5-VL-7B and Qwen3-VL-7B; videos are sampled at 1fps with up to 32 frames before selection and at most 12 after. Several parameters inherit upstream defaults (GroundingDINO-T threshold 0.3, ByteTrack association threshold, VideoMAE-v2 softmax temperature, PaddleOCR confidence cutoff) because the disturbance-aware confidence coupling absorbs calibration mismatch via $\rho(\mathbf{F})$ without retraining. All values in Tab. 8 are fixed at initialization and not adjusted per benchmark, task, or corruption mode, providing full reproducibility for main results (Tabs. 1 and 2) and ablations (Secs. D.2 and 4.3).

Table 8: Master parameter reference for Robust-TO. Parameters are grouped by pipeline stage. *Source* indicates the equation, table, or section where the parameter is introduced; *Sensitivity* reports the accuracy change (UrbanVideo-Bench, Qwen2.5-VL-7B) when the parameter is moved one step away from the chosen value, where measured. Selection thresholds $\theta_{\text{rel}}, \theta_{\text{sim}}$ are reported as the values used in our experiments; they are referred to as hyperparameters in Eq. (3) without a fixed numerical value.

Group	Parameter	Value	Source	Sensitivity
Disturbance estimation	τ_{blur} (Laplacian normalizer)	500	Sec. C	low
	μ_{ref} (neutral luminance midpoint)	0.5	Sec. C	fixed by definition
	τ_{edge} (Sobel edge threshold)	30	Sec. C	low
	Channel weights $(w_{\text{b}}, w_{\text{l}}, w_{\text{o}})$	equal (1:1:1) after min-max norm.	Eq. (2)	not measured
Frame selection	θ_{rel} (reliability threshold)	0.55	Eq. (3)	not measured
	θ_{sim} (relevance threshold)	0.30	Eq. (3)	not measured
	K (top-K trustworthy frames, adaptive)	$K \in [4, 12]$, host VLM chooses	Sec. 3.3	not measured
Confidence interface	Worst-K aggregator for $\rho(\mathbf{F})$	$K = \lceil n/3 \rceil$	Eq. (4); Tab. 15	-3.7 pt at uniform mean
	HIGH-tier threshold	$c_j \geq 0.7$ and $d < 0.3$	synthesis prompt	—
	LOW-tier threshold	$c_j < 0.3$ or $d \geq 0.7$	synthesis prompt	—
	Intrinsic confidence clipping	$c_j^{\text{intrinsic}} \in [0.01, 1.0]$	Eq. (4)	numerical safety
GRPO reward	λ (tool-cost weight)	0.5	Eq. (6)	± 0.4 pt at $\lambda \in \{0.25, 0.75\}$
	α (excess sub-query penalty)	0.2	Eq. (5)	not measured
	β (coverage saturation)	1.0	Eq. (5)	not measured
	w (auxiliary-reward weight in Eq. (8))	$1/3$	Eq. (8); Sec. 3.4	—
	π_{est} (predicts m^*)	frozen Qwen2.5-7B-Instruct (text-only)	Sec. 3.4	-1.2 pt if policy-internal
	Failed-call penalty (c_j)	0	Sec. 3.4	—
Training & inference	Optimizer	AdamW + DeepSpeed ZeRO-2 + FSDP	Tab. 10	—
	Learning rate (peak / schedule)	1×10^{-6} / cosine, 200-step warmup	Tab. 10	—
	Rollout group size	16	Tab. 10	—
	Batch size $\times$ grad-accum	2×4 per GPU (4×A100-80GB)	Tab. 10	—
	Max sequence length	8,192 tokens	Tab. 10	—
	KL penalty	0.01	Tab. 10	—
	Frame sampling rate	1 fps	Tab. 10	—
	Max frames (pre / post selection)	32 / 12	Tab. 10	—

Table 9: Visual tool library. All tools share the unified `(result, confidence)` interface (Eq. (4)). $\text{cost}(T_j) \in [0, 1]$ is the empirical normalized wall-time on a single A100 GPU at the host VLM's native resolution, calibrated against `caption_frame` as the unit. Users may define their own tool sets and assign arbitrary costs, as long as every tool conforms to the unified `(result, confidence)` interface.

Tool	Category	Cost	Description
`assess_quality`	Selection	0.10	Parameter-free per-frame IQA Eq. (2)
`select_frames`	Selection	0.15	Joint reliability-informativeness ranking
`retrieve_frames`	Selection	0.20	confidence-guided retrieval from pool $\mathcal{P}$
`detect_objects`	Perception	0.50	Object detection with bounding boxes
`caption_frame`	Perception	0.30	Dense captioning of frame content
`track_temporal`	Perception	0.70	Multi-frame object/action tracking
`recognize_action`	Perception	0.60	Action recognition with temporal context
`read_text`	Perception	0.25	OCR for in-video text

B.1 Tool implementation details.

Each tool in Tab. 9 wraps an existing pretrained model and exposes the unified `(result, confidence)` contract described in Eq. (4). The intrinsic confidence $c_j^{\text{intrinsic}}$ is derived differently per tool:

- `assess_quality`: deterministic; outputs the composite disturbance score $d(f_i)$ directly with $c^{\text{intrinsic}} = 1.0$ (no model uncertainty).
- `select_frames` / `retrieve_frames`: returns the selection score $s(f_i)$ from Eq. (2); $c^{\text{intrinsic}}$ is the cosine similarity $\text{sim}(\phi(f_i), \psi(q))$.
- `detect_objects`: wraps a GroundingDINO-T model; $c^{\text{intrinsic}}$ is the mean detection confidence over returned bounding boxes (boxes below a 0.3 threshold are discarded).

- `caption_frame`: wraps the host VLM in captioning mode; $c^{\text{intrinsic}}$ is the mean token-level log-probability of the generated caption, mapped to $[0, 1]$ via $\sigma(\cdot)$.
- `track_temporal`: wraps a ByteTrack tracker over consecutive selected frames; $c^{\text{intrinsic}}$ is the mean IoU of matched tracklets across frames.
- `recognize_action`: wraps a VideoMAE-v2 classifier; $c^{\text{intrinsic}}$ is the softmax probability of the top-1 predicted action class.
- `read_text`: wraps PaddleOCR; $c^{\text{intrinsic}}$ is the mean character-level recognition confidence.

All intrinsic scores are clipped to $[0.01, 1.0]$ to prevent zero-multiplication in Eq. (4). The final confidence c_j is then computed by multiplying $c_j^{\text{intrinsic}}$ with the input reliability $\rho(\mathbf{F})$.

B.2 Experiment Configuration

Training details. Tab. 10 summarizes all hyperparameters used for GRPO training. We use the same configuration for both the Qwen2.5-VL-7B and Qwen3-VL-7B backbones unless noted otherwise.

Table 10: GRPO training hyperparameters.

Hyperparameter	**Value**
Hardware	$4\times$ NVIDIA A100-80GB
Training framework	DeepSpeed ZeRO-2 + FSDP
Total training steps	$\sim$5,000
Rollout group size	16
Learning rate	1×10^{-6}
Learning rate schedule	Cosine with 200-step warmup
Batch size (per GPU)	2
Gradient accumulation steps	4
Max sequence length	8,192 tokens
Max frames per video	32 (pre-selection), 12 (post-selection)
Frame sampling strategy	1 fps
KL penalty coefficient	0.01
Reward weights (Eq. (8)*)*	
w_{acc}	1.0
w_{subq}	0.3
w_{cc}	0.3
w_{fmt}	0.3
Sub-query reward parameters (Eq. (5)*)*	
α (excess penalty)	0.2
β (coverage saturation)	1.0
Confidence–cost parameters	
λ (cost weight)	0.5
π_{est}	Qwen2.5-7B-Instruct (text-only)

Training data. We train on the video subset of the Video-R1 dataset, which contains approximately 12K video–question–answer triplets spanning indoor navigation, outdoor driving, egocentric activities, and surveillance footage. We do not use any additional video data or synthetic corruption augmentation during training; all corruption robustness is acquired through the confidence-guided pipeline and the confidence-cost reward. For π_{est}, we pre-compute the optimal sub-query count m^* for each training question once using the frozen Qwen2.5-7B-Instruct and cache the results.

B.3 Dataset and Evaluation Details

UrbanVideo-Bench. UrbanVideo-Bench is a benchmark for outdoor embodied spatial reasoning in urban driving scenarios. It comprises four tasks: *Landmark Position* (LP), *Counterfactual* (CF), *Progress Evaluation* (PE), and *Action Generation* (AG). All tasks are formulated as multiple-choice questions. We use the official evaluation split and report accuracy (%).

VSI-Bench. VSI-Bench focuses on indoor spatial intelligence and covers four tasks: *Relative Distance* (RDist), *Relative Direction* (RDir), *Route Planning* (RP), and *Appearance Order* (AO). Videos are captured from ego-centric indoor navigation. We follow the official evaluation protocol with accuracy (%) as the metric.

Corruption generation (RoVA). To evaluate robustness, we generate corrupted variants of both benchmarks using the RoVA video masker, which applies five corruption types: *Motion Blur* (MB), *Gaussian Noise* (GN), *Glare* (GL), *Occlusion* (Occ), and *Low-Light* (LL). Each corruption is applied at a medium severity level to randomly selected 40–60% of frames in each video, simulating realistic intermittent degradation (e.g., a dashcam intermittently catching glare from oncoming headlights). The corruption mask is unknown to all models at inference time. We generate one corrupted variant per corruption type per video.

Table 11: Dataset statistics for clean and corrupted evaluation.

Benchmark	Tasks	Videos	Questions	Avg. duration (s)
UrbanVideo-Bench (clean)	4	1,028	1,028	38.4
UrbanVideo-Bench (corrupted)	4×5 masks	5,140	5,140	38.4
VSI-Bench (clean)	4	762	762	25.7

C Disturbance Score Formulation

The three components of the disturbance score in Eq. (2) are defined as follows. All scores are computed per frame and min–max normalized across the video $\tilde{\mathcal{V}}$ before weighting.

Blur score d_{blur}. We compute the variance of the Laplacian of the grayscale frame:

$$d_{\text{blur}}(f_i) = 1 - \min\Big(1,\ \frac{\text{Var}(\nabla^2 f_i^{\text{gray}})}{\tau_{\text{blur}}}\Big), \tag{9}$$

where $\tau_{\text{blur}} = 500$ is a normalization constant calibrated on a held-out set of clean and blurred frames. Sharp frames have high Laplacian variance and thus low d_{blur}; blurry frames have low variance and high d_{blur}.

Brightness score d_{bright}. We measure deviation of mean luminance from a neutral midpoint:

$$d_{\text{bright}}(f_i) = 2\,\big|\mu_{\text{lum}}(f_i) - 0.5\big|, \tag{10}$$

where $\mu_{\text{lum}}(f_i) \in [0, 1]$ is the mean pixel intensity in the V channel of HSV space. Both under-exposed ($\mu_{\text{lum}} \to 0$) and over-exposed ($\mu_{\text{lum}} \to 1$) frames receive high disturbance scores.

Occlusion score d_{occl}. We estimate the fraction of the frame lacking informative edge structure via Sobel-magnitude statistics:

$$d_{\text{occl}}(f_i) = 1 - \frac{|\{p : G(p) > \tau_{\text{edge}}\}|}{H \times W}, \tag{11}$$

where $G(p) = \sqrt{G_x(p)^2 + G_y(p)^2}$ is the Sobel gradient magnitude at pixel p, $\tau_{\text{edge}} = 30$ is the edge threshold, and $H \times W$ is the frame resolution. Frames with large uniform (occluded) regions yield fewer edge pixels and thus higher d_{occl}.

Table 12: Comparison of Robust-TO (adaptive key-frame selection) against uniform-sampling baselines on UrbanVideo-Bench and VSI-Bench. Both settings use the full tool-augmented reasoning pipeline; the only difference is whether frames are selected adaptively (avg. 20.7 frames) or uniformly (21 frames). Best results in **bold**, second best underlined.

Method	Frames	Avg.	UrbanVideo-Bench				VSI-Bench			
			LP	CF	PE	AG	RDist	RDir	RP	AO
Uniform Sampling (with tools)										
Qwen2.5-VL-7B-Instruct	21	48.7	53.0	57.5	38.0	45.2	47.8	42.5	**37.8**	67.5
Qwen3-VL-7B-Instruct	21	54.3	58.5	62.0	42.8	56.2	53.0	46.5	**41.2**	73.8
Robust-TO (Adaptive Key-Frame Selection)										
Robust-TO + Qwen2.5-VL-7B-Instruct	20.7	50.7	55.1	59.9	39.7	47.6	50.0	44.3	36.8	72.0
Robust-TO + Qwen3-VL-7B-Instruct	20.7	**56.4**	**61.1**	**64.4**	**44.7**	**59.0**	**55.5**	**48.8**	39.8	**77.5**

D Additional Experiments

D.1 Additional Baselines under Matched Frame Budget

To isolate the contribution of Robust-TO's adaptive frame selection from the tool-augmented reasoning pipeline itself, we compare against the same pipeline using uniform sampling at a matched frame budget. As shown in Tab. 12, when both settings consume approximately the same number of frames (around 21) and share the full tool interface, adaptive selection consistently outperforms uniform sampling: Robust-TO with Qwen2.5-VL-7B improves average accuracy by 2.0%p (50.7 vs. 48.7), and the Qwen3-VL-7B variant achieves a 2.1%p gain (56.4 vs. 54.3). Although the absolute gap is moderate, as both settings already benefit from confidence-weighted synthesis and confidence-guided routing, the gains are remarkably consistent across tasks and backbones, confirming that *which* frames enter the pipeline matters even when the downstream reasoning is identical. The largest per-task improvements appear on Appearance Order (4.5%p for Qwen2.5-VL-7B, 3.7%p for Qwen3-VL-7B) and Relative Distance (2.2%p and 2.5%p, respectively), both of which require integrating evidence from temporally separated frames, precisely the setting where reliability-weighted frame gating prevents degraded frames from contaminating the reasoning chain. Conversely, Route Planning is the only task where uniform sampling slightly outperforms adaptive selection (37.8 vs. 36.8 for Qwen2.5-VL-7B), consistent with the main-paper observation that dense spatial coverage benefits this task more than selective high-quality sampling. These results demonstrate that adaptive frame selection provides a complementary and non-redundant improvement on top of the tool-augmented pipeline.

D.2 Ablation Studies

Ablation of Sub-Query Decomposition Modality. A natural question is whether the visual content of the selected trustworthy frames contributes to the quality of sub-query decomposition, or whether the text of q alone suffices. We compare two variants: (i) *Text*, which decomposes q using only the question string, and (ii) *Text+Frame*, which additionally conditions on the visual content of the top-K frames identified by the frame selector (Eq. (3)).

Table 13: Ablation of sub-query decomposition modality (Qwen2.5-VL-7B + Robust-TO). **Text** decomposes q using only the question text; **Text+Frame** additionally conditions on the visual content of the selected trustworthy frames. PVRBench averages over all five corruption masks.

Modality	LP	CF	PE	AG	Avg.
UrbanVideo-Bench (Clean)					
Text	50.3	55.8	37.2	43.1	46.6
Text+Frame	**55.1**	**59.9**	**39.7**	**47.6**	**50.7**
PVRBench (Corrupted, avg. 5 masks)					
Text	44.6	50.1	33.4	38.2	41.6
Text+Frame	**49.8**	**55.7**	**37.1**	**44.9**	**46.9**

As shown in Tab. 13, grounding decomposition in the actual video content yields a consistent +4.1%p and +5.3%p gain on clean and corrupted inputs, respectively. The benefit is most pronounced on Action Generation (+4.5%p on clean, +6.7%p on corrupted), where inspecting the frames reveals action-specific primitives that pure text parsing cannot anticipate. For instance, a question about "what the pedestrian does after the car stops" requires seeing whether

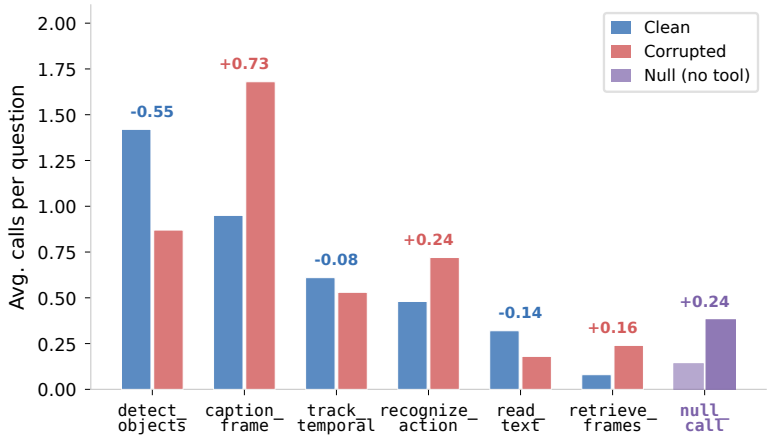


Figure 5: Ablation of Tool Routing. Per-task accuracy under three routing strategies: fixed tool, query-only, and the full confidence-guided policy.

Table 14: Impact of frame selection on clean and corrupted UrbanVideo-Bench (Qwen2.5-VL-7B + Robust-TO). "Frames" reports the average number of frames forwarded to perception tools. Δ_{drop} = Clean − Corrupted; smaller is better.

Setting	**Frames**	**Clean**	**Corrupted**	$\boldsymbol{\Delta_{\text{drop}}}$	**Train (h)**	**Infer (s)**
w/o Frame Selection	32	49.1	43.5	5.6	127.9	243.7
w/ Frame Selection	20.7	**50.7**	**47.1**	**3.6**	111.7	157.6
Improvement		↑1.6	↑3.6	↓2.0	↓16.2	↓86.1

a crosswalk or traffic signal is present to generate the right sub-queries; text-only decomposition defaults to generic temporal primitives that miss these visual cues. On the corrupted setting, the clean-to-corrupted accuracy drop also shrinks from 5.0%p (Text) to 3.8%p (Text+Frame), indicating that frame-grounded decomposition produces sub-queries better aligned with what the selected trustworthy frames can actually support, thereby reducing wasted tool calls on evidence that turns out to be degraded.

Ablation of Tool Routing (Extended Analysis). Fig. 5 compares three routing strategies across all eight tasks. The fixed-tool baseline (always `caption_frame`) serves as a lower bound, since captioning is the most broadly applicable but least specialized tool. Query-only routing improves over the fixed baseline by selecting tools based on the semantic type of each sub-query, but ignores frame quality entirely. The full confidence-guided policy adds a second routing dimension - the dominant corruption profile of the selected frames - and outperforms query-only by +6.1%p on average. Gains are highly task-dependent: Appearance Order benefits most (+9.6%p), because temporal ordering under blur requires shifting from `detect_objects` (which loses bounding-box precision) to `caption_frame` (which tolerates spatial degradation). Conversely, Route Planning shows only +1.7%p improvement, as this task prioritizes dense spatial coverage over per-frame tool selection. These results confirm that the learned routing policy captures a non-trivial mapping from corruption profiles to tool effectiveness rather than defaulting to a single preferred tool.

Ablation of Frame Selection. Tab. 14 demonstrates that the reliability-aware frame selector simultaneously improves accuracy and reduces computational cost. By discarding frames with high disturbance scores, the average frame count drops from 32 to 20.7 (35% drops), yet clean accuracy increases by 1.6%p (50.7 vs. 49.1) because noisy frames no longer inject misleading evidence into downstream tools. The benefit is amplified under corruption: corrupted accuracy improves by 3.6%p (47.1 vs. 43.5), and the clean-to-corrupted drop Δ_{drop} shrinks from 5.6%p to 3.6%p , confirming that frame gating is the primary mechanism for graceful degradation. On the efficiency side, training time decreases by over 16 hours and per-sample inference time drops by 86 seconds, as fewer frames propagate through the perception tool pipeline. This result is noteworthy: it shows that *more frames*

Table 15: Ablation of worst-K aggregation strategy for input reliability $\rho(\mathbf{F})$ on UrbanVideo-Bench and VSI-Bench (Qwen2.5-VL-7B + Robust-TO).

Aggregation	**Formula**	**UrbanVideo** clean	**UrbanVideo** corrupted	**VSI-Bench** clean
Uniform mean	$K = n$	47.0	44.8	46.3
$n/4$ worst	$K = \lceil n/4 \rceil$	47.3	45.2	46.8
$n/3$ **worst (ours)**	$\boldsymbol{K = \lceil n/3 \rceil}$	**50.7**	**54.3**	**52.1**
$n/2$ worst	$K = \lceil n/2 \rceil$	49.1	51.8	50.4

Table 16: Comparison of disturbance estimators on corrupted UrbanVideo-Bench (Qwen2.5-VL-7B + Robust-TO). Accuracy is reported per corruption type.

Estimator	**Motion Blur**	**Gaussian Noise**	**Glare**	**Occlusion**	**Low-Light**
NIQE	44.8	48.1	47.1	43.2	48.3
BRISQUE	46.1	49.8	48.5	44.9	49.7
Ours	**53.4**	**53.9**	**55.4**	**53.6**	**55.8**

do not necessarily yield better answers -selectively retaining trustworthy frames is strictly preferable to exhaustively processing all available frames.

Ablation of Worst-K Aggregation. The choice of aggregation function for input reliability $\rho(\mathbf{F})$ directly affects how aggressively the system penalizes degraded frames within a tool's input set. Tab. 15 compares four strategies. Uniform mean ($K{=}n$) averages reliability across all frames, which allows a single clean frame to mask severe corruption in others; this yields the lowest accuracy on both clean and corrupted benchmarks. At the other extreme, $n/2$ worst focuses on half the frames and is overly conservative: it discounts too much evidence even when the majority of frames are clean, losing 1.6%p on clean UrbanVideo relative to $n/3$ worst. The $n/4$ worst variant is too lenient, performing similarly to uniform mean because it only examines a small tail of the reliability distribution. Our chosen $K{=}\lceil n/3 \rceil$ strikes the best balance: it is sensitive enough to detect when a meaningful fraction of frames is corrupted, while remaining permissive enough to retain useful evidence from predominantly clean input sets. The 3.7%p gap between uniform mean and $n/3$ worst on clean UrbanVideo (47.0 vs. 50.7) confirms that even on clean data, worst-K aggregation provides a useful inductive bias by preventing the system from over-trusting marginal frames.

Comparison of Disturbance Estimators. We compare our parameter-free disturbance score (Eq. (2)) against two established no-reference image quality assessment (NR-IQA) methods: NIQE and BRISQUE, both used as drop-in replacements within the Robust-TO pipeline. As shown in Tab. 16, our estimator outperforms NIQE and BRISQUE by 7-10%p and 5-8%p, respectively, across all five corruption types. The advantage is most pronounced on Motion Blur (+8.6%p over NIQE) and Occlusion (+10.4%p over NIQE), where our decomposed disturbance profile – which separately measures blur, brightness, and occlusion - directly captures the dominant degradation mode, whereas NIQE and BRISQUE produce a single scalar quality score that conflates different corruption sources. This conflation is particularly harmful for downstream tool routing: a unified quality score cannot distinguish blur (which favors `caption_frame`) from occlusion (which favors `recognize_action`), so routing decisions become less targeted. Additionally, NIQE and BRISQUE are calibrated on natural image statistics that may not generalize well to video frames captured under extreme conditions (e.g., dashcam footage with strong glare), whereas our estimator uses simple, domain-agnostic signal-level statistics that require no training data.

Effect of Key-Frame Extraction across Backbones. Tab. 17 demonstrates that the benefits of Robust-TO's key-frame extraction generalize across backbone architectures. On the Qwen2.5-VL-7B backbone, Robust-TO improves over Video-R1 by +7.7%p on clean data and +7.1%p on corrupted data, while using roughly 35%p fewer frames. The gains are consistent with the stronger Qwen3-VL-7B backbone: +4.4%p on clean and +4.8%p on corrupted data. Notably, the corrupted-setting improvement is slightly larger than the clean-setting improvement for Qwen3-VL-7B (+4.8%p vs. +4.4%p), suggesting that the confidence-guided frame selection becomes increasingly valuable as the base model is stronger, which is a stronger backbone extracts better evidence from the frames

Table 17: Effect of key-frame extraction (KFE) on clean and corrupted UrbanVideo-Bench. Robust-TO uses adaptive frame selection averaging 20.7 frames per video.

Method	**Frames**	**UrbanVideo** clean	**UrbanVideo** corrupted
Video-R1 + Qwen2.5-VL-7B	1 fps	43.0	40.7
Video-R1 + Qwen3-VL-7B	1 fps	52.0	49.3
Robust-TO + Qwen2.5-VL-7B	1 fps (avg 20.7f)	**50.7**	47.8
Robust-TO + Qwen3-VL-7B	1 fps (avg 20.7f)	**56.4**	**54.1**

it receives, so filtering out corrupted frames has a compounding effect. Video-R1, by contrast, processes all frames at 1 fps without quality-based filtering, so its corrupted accuracy drops by 2.3%p (Qwen2.5-VL-7B) and 2.7%p (Qwen3-VL-7B) relative to clean, whereas Robust-TO limits these drops to 2.9%p and 2.3%p, respectively, demonstrating more graceful degradation.

D.3 Additional Case Studies

Tabs. 18 to 21 present four additional case studies covering two benchmarks (UrbanVideo-Bench Landmark Position, Action Generation; VSI-Bench Relative Distance, Appearance Order) with distinct task structures, video lengths, and corruption profiles.

Landmark Position (Tab. 18). A case study with 30-frame urban drone flyover corrupted by four simultaneous disturbances: wiper blur on f_3–f_5 and f_{17}–f_{19}, headlight glare on f_9–f_{11}, and foreground tree occlusion on f_{23}–f_{25}, with query *"In what temporal order does the drone pass each of the following landmarks: (A) the clock tower, (B) the pedestrian bridge, (C) the cathedral with twin spires, and (D) the river fountain?"*.
The standard VLM reports the river fountain before the cathedral: the glare frames reflect headlights on wet pavement (reasoning: *bright flashes and what looks like water*), which the model interprets as the fountain, committing to an incorrect ordering with self-reported high confidence. Robust-TO excludes all four corrupted windows and selects eight trustworthy frames, including the key observation that f_{10} carries high query similarity ($\mathrm{sim} = 0.71$, the glare resembles a fountain) but is correctly suppressed by its elevated brightness disturbance. All five sub-queries yield `MED`-tier evidence owing to global low-light, yet they are mutually consistent and converge on the correct ordering A→B→C→D without contradiction-demonstrating that Robust-TO produces a reliable answer even when no `HIGH`-tier evidence is available.

Action Generation (Tab. 19). A case study with a 20-frame drone approach toward power lines corrupted by night-time Gaussian noise throughout and heavy motion-blur bursts on f_5–f_8 and f_{14}–f_{15}, with query *"Given the current scene the drone observes, what action should the agent take next to safely continue its delivery mission? (A) Ascend to clear the wires. (B) Descend below the wires. (C) Turn right to bypass the obstacle. (D) Hover in place."*.
The standard VLM recommends *descending* below the wires, because blur on those frames obscures the forested ground below, making the lower airspace appear clear (reasoning: *Going down often helps avoid wires in urban scenes*). Robust-TO identifies blur as the dominant corruption, and applies the confidence-guided routing flip: all four sub-queries switch from `detect_objects` to `caption_frame`, since detection is unreliable under blur while captioning tolerates sharpness variation. The resulting evidence correctly reports open sky above and forested terrain below, leading to the safe decision to ascend.

Relative Distance (Tab. 20). A case study with an 18-frame ego-centric kitchen walkthrough corrupted by dim evening lighting and a roommate walking past the camera on f_8–f_{11}, occluding approximately 55% of the view, with query *"Among the following four objects, which one is closest to the refrigerator? (A) the dining table (B) the kitchen sink (C) the trash bin (D) the microwave"*.
The standard VLM incorrectly identifies the trash bin as closest to the refrigerator: the person-occluded showed the trash bin appearing close to fridge (reasoning: *Someone walks past the camera in several frames making it hard to see. The trash bin appears to be right next to the fridge in some frames*). Robust-TO excludes person-occluded frames entirely (f_8–f_{11}) and detects the kitchen sink adjacent to the refrigerator on the same counter wall. A subtle near-miss is resolved by scene-graph reasoning: although the trash bin is slightly closer in absolute distance, it is separated by the

Table 18: Case study comparing STANDARD VIDEO-LLM and ROBUST-TO on a corrupted urban drone flyover video (UrbanVideo-Bench, Landmark Position). Question: *In what temporal order does the drone pass each of the following landmarks: (A) the clock tower, (B) the pedestrian bridge, (C) the cathedral with twin spires, and (D) the river fountain?*

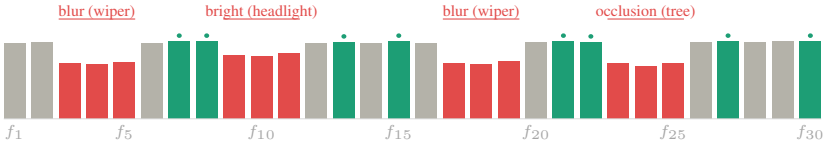
STANDARD VIDEO-LLM

Reasoning	I see urban evening scenes. There is a clock tower visible early on. Some frames are blurry but I can still see structures. Later there are bright flashes and what looks like water. I also see a tall structure with spires. The order seems to be clock tower, then a bridge, then a fountain near the bright section, and then the cathedral at the end.
Generated Answer	Order: **(A)** clock tower → **(B)** pedestrian bridge → **(D)** river fountain → **(C)** cathedral. *Failure mode: headlight-glare frames f_9–f_{11} treated as equally informative; reflected glare on wet pavement interpreted as the river fountain, causing (C) and (D) to be swapped. Self-reported confidence: "the order is clear."*

ROBUST-TO (OURS)

Quality Profiling + Frame Selection

Bar height: $1-d(f_i)$ (reliability)



• Selected (Top-8 selected by $s(f_i)$) • Not selected • Excluded (Corrupted)

Selected frames: $[f_7, f_8, f_{13}, f_{15}, f_{21}, f_{22}, f_{27}, f_{30}]$

Note: f_{10} has high query similarity (0.71, headlight glare resembles river fountain reflection) but excluded: $d_{\text{bright}} = 0.74$ severely suppresses reliability. The baseline commits fountain-before-cathedral based on this glare frame.

Tool Outputs ***(result, conf)***

Sub-query	Tool Call	Result	Conf.	Tier
sq_1: *Detect the clock tower*	`detect_objects` (f_7, f_8)	f_7: 0.91, f_8: 0.89; earliest: f_7	0.500	`MED`
sq_2: *Detect the pedestrian bridge*	`detect_objects` (f_{13}, f_{15})	f_{13}: 0.86, f_{15}: 0.92; earliest: f_{13}	0.476	`MED`
sq_3: *Detect the cathedral with twin spires*	`detect_objects` (f_{21}, f_{22})	f_{21}: 0.93, f_{22}: 0.95; earliest: f_{21}	0.533	`MED`
sq_4: *Detect the river fountain*	`detect_objects` (f_{27}, f_{30})	f_{27}: 0.71, f_{30}: 0.78; earliest: f_{27}	0.431	`MED`
sq_5: *Temporal order of landmarks*	`track_temporal` (f_7, f_8, f_{13}, f_{15}, f_{21}, f_{22}, f_{27}, f_{30})	$f_7 \to f_{13} \to f_{21} \to f_{27}$ (A→B→C→D)	0.476	`MED`

Generated Answer	**(A)** clock tower → **(B)** pedestrian bridge → **(C)** cathedral with twin spires → **(D)** river fountain. Clock tower first at f_7, bridge at f_{13}, cathedral at f_{21}, fountain at f_{27}, confirmed by `track_temporal` across all 8 selected frames. Glare window f_9–f_{11} correctly gated out. (5 `MED`-tier evidence; no HIGH-tier due to global low-light; no contradictions; overall confidence: `MED`).
Ground Truth	**(A)** clock tower → **(B)** pedestrian bridge → **(C)** cathedral with twin spires → **(D)** river fountain.

refrigerator body at an opposite corner, whereas the sink shares the same wall and counter. Robust-TO correctly reports the sink with `MED`-tier confidence, appropriately reflecting the estimator noise of the near-miss.

Appearance Order (Tab. 21). A case study with a 22-frame living-room sweep corrupted by a sunlit-window glare burst on f_6–f_9 and a pet cat crossing the field of view on f_{13}–f_{16}, with query *"In what temporal order does the camera first encounter each of the following objects during the room walkthrough? (A) a yellow armchair (B) a bookshelf with red books (C) a wall-mounted television (D) a houseplant in a ceramic pot"*.

Table 19: Case study comparing STANDARD VIDEO-LLM and ROBUST-TO on a corrupted drone video (UrbanVideo-Bench, Action Generation). Question: *Given the current scene the drone observes, what action should the agent take next? (A) Ascend to clear the wires. (B) Descend below the wires. (C) Turn right to bypass the obstacle. (D) Hover in place.*

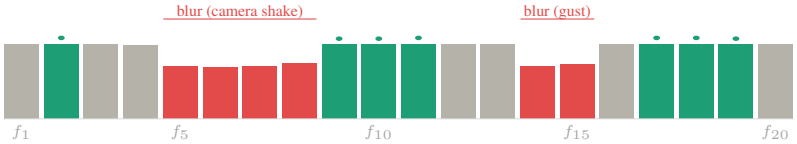
STANDARD VIDEO-LLM

Reasoning	The drone is flying near some structures at night. Visibility is poor and some frames are blurry. I can see what may be wires or beams. The drone could either go up, down, or around. Going down often helps avoid wires in urban scenes.
Generated Answer	**(B)** Descend below the wires. *Failure mode: motion-blur frames f_5–f_8 obscured the forested ground texture, making the lower envelope appear clear. Gaussian noise further suppressed the forest canopy signal, reinforcing the illusion of safe empty space below. Defaulted to familiar prior: "descend to avoid overhead obstacles."*

ROBUST-TO (OURS)

Quality Profiling + Frame Selection

Bar height: $1-d(f_i)$ (reliability)



• Selected (Top-7 selected by $s(f_i)$) • Not selected • Excluded (Corrupted)
Selected frames: $[f_2, f_9, f_{10}, f_{11}, f_{17}, f_{18}, f_{19}]$
Note: $K=7$ (safety-critical: tight window around current moment). Blur bursts f_5–f_8 correctly excluded — these frames obscured ground texture and misled the baseline into choosing (B).

Tool Outputs ***(result, conf)***

Sub-query	Tool Call	Result	Conf.	Tier
sq_1: *Identify obstacles and height relative to drone*	`caption_frame` (f_9, f_{10}, f_{11})	"3 power lines at mid-height; drone at line altitude"	0.374	`MED`
sq_2: *Identify signage indicating altitude restrictions*	`caption_frame` (f_{17}, f_{18}, f_{19})	"No altitude restriction; hazard markers on rightmost wire"	0.363	`MED`
sq_3: *Recognize drone's current action*	`caption_frame` $(f_9, f_{10}, f_{17}, f_{18})$	"Level flight, slight forward drift; not climbing or descending"	0.365	`MED`
sq_4: *Determine clear airspace above / below wires*	`caption_frame` $(f_2, f_{10}, f_{18}, f_{19})$	"Open sky above; forested ground below, no safe descent"	0.396	`MED`

Generated Answer	**(A)** Ascend to clear the wires. Power lines at drone altitude (sq1). Forested terrain below eliminates safe descent (sq4) — option (B) infeasible. No bypass route identified (sq2). Level flight confirmed (sq3); ascent is the minimal safe maneuver. (4 `MED`-tier evidence; all routing to `caption_frame` due to dominant blur; no contradictions; overall confidence: `MED`).
Ground Truth	**(A)** Ascend to clear the wires.

The standard VLM inverts the bookshelf and television in the ordering: during the glare burst the camera sweeps past the bookshelf (reasoning: *some bright frames where I can make out a shelf with colored items.*), but over-exposure washes out the red book spines, so the model does not detect the bookshelf until f_{11}, where the television also first appears; lacking a reliability signal, the model breaks the tie by visual salience (the television is larger) and reports C before B. Robust-TO detects the bookshelf at f_5 (the last clean frame before the glare burst), establishing it as the second-earliest object.

Table 20: Case study comparing STANDARD VIDEO-LLM and ROBUST-TO on a corrupted indoor kitchen video (VSI-Bench, Relative Distance). Question: *Among the following four objects, which one is closest to the refrigerator? (A) dining table (B) kitchen sink (C) trash bin (D) microwave*

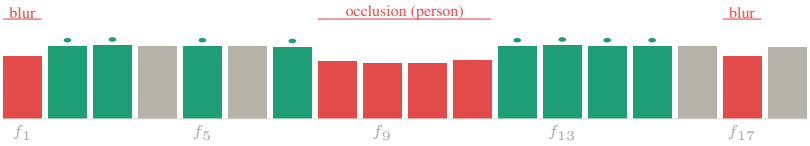
STANDARD VIDEO-LLM

Reasoning	I see a kitchen in a dim indoor setting. There is a refrigerator, a table, a sink, a trash bin, and a microwave. Someone walks past the camera in several frames making it hard to see. The trash bin appears to be right next to the fridge in some frames, while the sink seems to be further away on a different counter.
Generated Answer	(C) the trash bin. *Failure mode: person-occlusion frames f_8–f_{11} showed the trash bin appearing close to the fridge's 2D silhouette (roommate's body aligned both objects in projection). Without occlusion gating, this spurious proximity signal was weighted equally with the clean-frame evidence showing the sink's counter adjacency.*

ROBUST-TO (OURS)

Quality Profiling + Frame Selection

Bar height: $\propto 1-d(f_i)$ (reliability)



• Selected (Top-8 selected by $s(f_i)$) • Not selected • Excluded (Corrupted)

Selected frames: $[f_3, f_5, f_{13}, f_7, f_{15}, f_{14}, f_2, f_{12}]$

Note: f_8–f_{11} excluded ($d_{\text{occl}} = 0.58$–0.66, person blocks $\sim$55% of view). In these frames, the roommate's body aligned the trash bin with the fridge's 2D silhouette, creating a false proximity signal that misled the baseline into choosing (C).

Tool Outputs ***(result, conf)***

Sub-query	Tool Call	Result	Conf.	Tier
sq_1: *Detect refrigerator and its position*	`detect_objects` (f_3, f_5, f_{13}, f_{15})	Fridge at right wall (all 4 frames)	0.382	`MED`
sq_2: *Detect all candidate objects*	`detect_objects` (f_7, f_{12}, f_{14}, f_{15})	Sink: right wall adj. to fridge; trash bin: left; microwave: above sink; table: center	0.352	`MED`
sq_3: *Estimate pairwise spatial relations to fridge*	`caption_frame` (f_{12}, f_{13}, f_{14}, f_{15})	Sink immediately right, same counter; trash bin $\sim$30 cm left; table $\sim$2 m	0.349	`MED`
sq_4: *Rank candidates by distance to fridge*	`caption_frame` (f_3, f_7, f_{13}, f_{15})	Sink $\sim$0.4 m (adj.); trash bin $\sim$0.3 m (opp. corner); micro $\sim$1.0 m; table $\sim$2 m	0.336	`MED`

Generated Answer	**(B)** the kitchen sink. Sink shares the same wall and counter as the fridge (sq3, sq4). Trash bin ($\sim$0.3 m) is slightly nearer numerically but separated by the fridge body (opposite corner); sink wins by same-side scene-graph adjacency. Occlusion frames f_8–f_{11} correctly gated out. (4 `MED`-tier evidence; no contradictions; near-miss sink vs. trash bin resolved by adjacency; overall confidence: `MED`).
Ground Truth	**(B)** the kitchen sink.

Table 21: Case study comparing STANDARD VIDEO-LLM and ROBUST-TO on a corrupted living-room walkthrough (VSI-Bench, Appearance Order). Question: In what order does the camera first encounter: (A) yellow armchair, (B) bookshelf with red books, (C) wall-mounted television, (D) houseplant in ceramic pot?

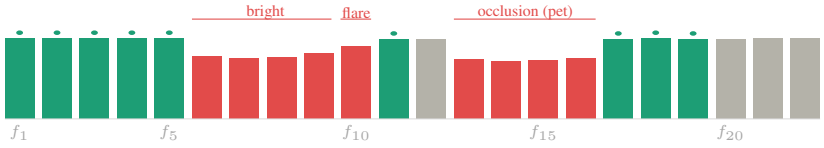
STANDARD VIDEO-LLM

Reasoning	I see a living room walkthrough. Early on I see what looks like an armchair. Then there are some bright frames where I can make out a shelf with colored items. Later there is what appears to be a TV and then some movement, and then a plant near a window. The order is armchair, then a TV, then the bookshelf, then the plant.
Generated Answer	**(A)** armchair → **(C)** television → **(B)** bookshelf → **(D)** houseplant. *Failure mode: glare burst f_6–f_9 washed out the red books; bookshelf went undetected until f_{11}, where the television also first appears. Without reliability gating, the model broke the tie by visual salience (TV larger), inverting (B) and (C).*

ROBUST-TO (OURS)

Quality Profiling + Frame Selection

Bar height: $1-d(f_i)$ (reliability)



• Selected (Top-9 selected by $s(f_i)$) • Not selected • Excluded (Corrupted)
Selected frames: $[f_1, f_2, f_3, f_4, f_5, f_{11}, f_{17}, f_{18}, f_{19}]$
Note: bookshelf correctly first-detected at f_5 (pre-glare, clean), not f_{11} (where TV also appears). f_{10} excluded as borderline flare ($(1-d) = 0.39 < \theta_{\text{rel}}$).

Tool Outputs *(result, conf)*

Sub-query	Tool Call	Result	Conf.	Tier
sq_1: *Detect the yellow armchair*	`detect_objects` (f_1, f_2, f_3)	First seen: f_2 (score 0.94)	0.611	`MED`
sq_2: *Detect the bookshelf with red books*	`detect_objects` (f_4, f_5, f_{11})	First seen: f_5 (score 0.89)	0.587	`MED`
sq_3: *Detect the wall-mounted television*	`detect_objects` (f_{11}, f_{17}, f_{18})	First seen: f_{11} (score 0.92)	0.607	`MED`
sq_4: *Detect the houseplant in ceramic pot*	`detect_objects` (f_{17}, f_{18}, f_{19})	First seen: f_{18} (score 0.95)	0.618	`MED`
sq_5: *Temporal order of objects*	`track_temporal` (f_1–f_5, f_{11}, f_{17}–f_{19})	$f_2 \to f_5 \to f_{11} \to f_{18}$ (A→B→C→D)	0.592	`MED`

Generated Answer	**(A)** armchair → **(B)** bookshelf → **(C)** television → **(D)** houseplant. Armchair at f_2, bookshelf at f_5 (pre-glare clean frame), television at f_{11}, houseplant at f_{18}, confirmed by `track_temporal`. Glare f_6–f_9 and pet f_{13}–f_{16} gated out without loss of earliest-appearance evidence. (5 `MED`-tier; no contradictions; overall confidence: `MED`-leaning-`HIGH`).
Ground Truth	**(A)** armchair → **(B)** bookshelf → **(C)** television → **(D)** houseplant.

Table 22: Average tool invocations per question on UrbanVideo-Bench (Qwen2.5-VL-7B + Robust-TO). "Clean" denotes the original benchmark; "Corrupted" averages over all five RoVA masks.

Tool	Clean	Corrupted
`assess_quality`	1.00	1.00
`select_frames`	1.00	1.00
`detect_objects`	1.42	0.87
`caption_frame`	0.95	1.68
`track_temporal`	0.61	0.53
`recognize_action`	0.48	0.72
`read_text`	0.32	0.18
`retrieve_frames`	0.08	0.24
Total calls per question	5.86	6.22
Avg. sub-queries per question	3.1	3.4
Avg. selected frames K	7.8	6.2

Table 23: Learned routing preferences under dominant corruption modes. Each cell shows the tool preferred by the trained host VLM for a given sub-query type and corruption mode. The "Clean" column shows the default when no corruption dominates. *Abbreviations:* `det_obj=detect_objects`, `cap=caption_frame`, `trk_tmp=track_temporal`, `rec_act=recognize_action`, `rd_txt=read_text`.

Sub-query type	Clean	Blur	Brightness	Occlusion
Object identity / location	`det_obj`	`cap`	`det_obj`	`cap`
Object attribute / appearance	`cap`	`cap`	`cap`	`det_obj`
Temporal / motion	`trk_tmp`	`rec_act`	`trk_tmp`	`rec_act`
Action / event	`rec_act`	`cap`	`rec_act`	`rec_act`
In-video text	`rd_txt`	`cap`	`rd_txt`	`rd_txt`

E Tool Invocation Statistics

To understand how Robust-TO allocates its computational budget, we collect tool invocation statistics across the full evaluation set. Tab. 22 reports the average number of calls per tool per question, broken down by clean and corrupted settings.

Several patterns emerge. First, `assess_quality` and `select_frames` are called exactly once per question in both settings, confirming their role as fixed first-stage operations. Second, under corruption, `detect_objects` calls decrease ($1.42 \rightarrow 0.87$) while `caption_frame` calls increase ($0.95 \rightarrow 1.68$), reflecting the learned routing preference: detection degrades under blur and occlusion, so the agent shifts toward captioning. Third, the average number of selected frames K drops from 7.8 to 6.2 under corruption, indicating the agent becomes more selective when fewer frames pass the reliability threshold. Fourth, `retrieve_frames` usage increases under corruption ($0.08 \rightarrow 0.24$), suggesting the agent occasionally retrieves additional frames when its initial selection yields insufficient high-confidence evidence.

Confidence-Guided Tool Routing Rules. The two-stage routing described in Sec. 3.3 maps the semantic type and dominant corruption to a tool choice. Tab. 23 summarizes the learned routing preferences after GRPO training. The routing is soft - the host VLM selects tools via in-context reasoning conditioned on the disturbance profile, not via a hard-coded lookup table.

F Prompt Templates

We provide the key prompt templates used by the host VLM at each stage of the Robust-TO pipeline. Angle brackets `{...}` denote dynamically filled placeholders.

F.1 Sub-Query Decomposition Prompt

Sub-Query Decomposition

[Task]
You are an expert video analyst. Your task is to decompose a complex question about a video into a minimal set of atomic sub-queries. Each sub-query must target exactly one perceptual primitive and be answerable by a single visual tool call. Do not generate redundant sub-queries.

[Decomposition Guidelines]

1. Identify the distinct perceptual demands implied by the question (e.g., object localization, action recognition, attribute comparison, text reading).
2. For each demand, formulate exactly one atomic sub-query targeting a single visual primitive.
3. Assign a semantic type to each sub-query: one of `[spatial, temporal, attribute, action, text]`.
4. Minimize the total number of sub-queries — each must be strictly necessary.

[Input]

- Video context: `{video_description}`
- Disturbance profile of selected frames: blur=`{avg_blur}`, brightness=`{avg_bright}`, occlusion=`{avg_occl}`
- Question: `{original_query}`

[Output Format]
Output a JSON list of atomic sub-queries. Only output the JSON — no explanations, no justifications, and no extra text of any kind.

```
[
  {"sub_query": "<sub-query text>", "type": "spatial"},
  {"sub_query": "<sub-query text>", "type": "temporal"},
  ...
]
```

F.2 Tool Routing Prompt

Confidence-guided Tool Routing

[Task]
You are a tool routing agent. Given a sub-query, its semantic type, and the disturbance profile of the selected frames, choose the best perception tool from the available library that maximizes result reliability under the current corruption conditions.

[Routing Guidelines]

- For **spatial** sub-queries under blur: prefer `caption_frame` over `detect_objects` (detection requires sharp visual boundaries).
- For **temporal** sub-queries under occlusion: prefer `recognize_action` over `track_temporal` (tracking loses targets under occlusion).
- For **text** sub-queries under blur: prefer `caption_frame` over `read_text` (OCR degrades rapidly under spatial blur).
- When brightness distortion dominates: prioritize tools robust to extreme illumination.
- When multiple tools are viable: prefer the one with lower cost.

[Input]

- Sub-query: `{sub_query_text}`
- Semantic type: `{type}`
- Disturbance profile: blur=`{d_blur}`, brightness=`{d_bright}`, occlusion=`{d_occl}`
- Dominant corruption: `{dominant_type}`
- Available tools: `{tool_list_with_costs}`

[Output Format]
Select the best tool. Only output the JSON — no explanations beyond the reason field.

```
{
  "tool": "<tool_name>",
  "reason": "<one-sentence justification>"
}
```

F.3 Confidence-Weighted Evidence Synthesis Prompt

Confidence-Weighted Evidence Synthesis

[Task]
You are synthesizing evidence collected from multiple visual tools to answer a question about a video. Each piece of evidence has a confidence score (0–1) and a source frame disturbance level. Your goal is to produce a reliable answer grounded in the most trustworthy evidence.

[Synthesis Rules]

1. Group evidence into three reliability tiers:
   - **HIGH**: confidence ≥ 0.7 and disturbance < 0.3
   - **MEDIUM**: all other evidence
   - **LOW**: confidence < 0.3 or disturbance ≥ 0.7
2. Build your answer primarily from HIGH-tier evidence.
3. Use MEDIUM-tier evidence only if it is consistent with HIGH-tier conclusions; discard it if contradictory.
4. Use LOW-tier evidence only when no HIGH-tier evidence exists, and explicitly note the uncertainty.
5. If all evidence is LOW-tier, state that the answer is uncertain.

[Input]

- Question: `{original_query}`
- Sub-queries and collected evidence:
  For each sub-query:
  - Sub-query: `{sq_text}`
  - Tool: `{tool_name}`, Result: `{result}`, Confidence: `{c_j}`
  - Source frames: `{frame_ids}`, Disturbance: `{d_scores}`

[Output Format]
Provide your step-by-step reasoning inside `<think>` tags, then your final answer inside `<answer>` tags. Only output these two blocks.

```
<think>
Step-by-step reasoning considering evidence reliability
and tier-based synthesis...
</think>
<answer>X</answer>
```

F.4 m^* Estimation Prompt (π_{est})

m_q^* Sub-Query Count Estimation

[Task]
You are estimating the minimum number of independent perceptual sub-queries needed to fully answer a video question. This estimate is used as a reference target for training — do not overestimate.

[Estimation Guidelines]

1. Count how many distinct objects, actions, or spatial relations the question asks about.
2. Determine whether temporal reasoning across multiple moments is needed (adds sub-queries).
3. Assess whether the question can be answered from a single frame or requires multi-frame evidence.
4. Each sub-query should be strictly necessary — do not pad the count.

[Input]

- Question: `{original_query}`
- Answer choices: `{choices_if_multiple_choice}`

[Output Format]
Output a single integer representing the estimated number of atomic sub-queries needed. No explanations, no extra text.

`<integer>`