ROBOGAZE: EVALUATING ROBOT WORLD MODELS VIA STRUCTURED VISION-LANGUAGE ANALYSIS

Anonymous authors
Paper under double-blind review

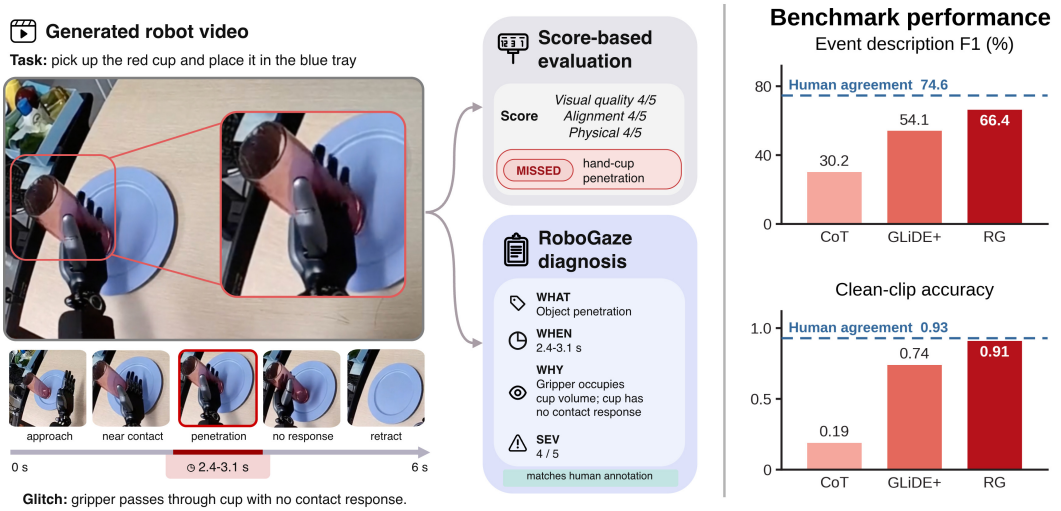


Figure 1: From scalar judgment to grounded diagnosis. *Left:* VideoScore2 (He et al., 2025), a learned score-based judge, rates the clip $4/5$ on visual quality, alignment and physical consistency, with no indication of what failed or when. ROBOGAZE identifies the hand–cup penetration, localizes it to 2.4–3.1 s, explains the supporting evidence and grades its severity. *Right:* against evaluators that do report events (CoT and GLiDE$^+$), on Gemini 3.1 Pro over all 914 clips, ROBOGAZE improves description F_1 and clean-clip accuracy. Dashed lines mark human agreement on the 180-clip gold subset; on that subset, model scores are within 0.9 F_1 of their full-benchmark values.

ABSTRACT

Robot world models now generate synthetic manipulation videos, but evaluating them is hard: realistic-looking outputs often violate physical laws, temporal consistency or task logic, while conventional metrics and monolithic Vision-Language Model (VLM) judges return scores that do not say what failed. We present RoboGaze, a training-free multi-agent VLM evaluator that grounds the task and scene, routes each subtask to dimension-specific specialists and verifies their findings with a critic, producing temporally localized diagnostic reports that state what failed, when, why and how severely, under a schema of 6 dimensions and 30 failure types whose annotation reliability and coverage we measure. To benchmark it, we introduce RoboGazeBench, 914 human-annotated clips from five generators, conditioned on simulated and real-world data and running from 4.9 to 148.9 s. Across eight open-source and proprietary VLM backbones, RoboGaze outperforms every baseline; even a published agentic detector given the same schema and a slightly larger inference budget trails it by 10.1–12.3 description-F_1 points, a margin that blinded human raters confirm. On the strongest backbone it closes 83% of the gap between chain-of-thought prompting and human agreement, and its critic curbs the "cry-wolf" false positives of standard VLMs, lifting mean clean-clip accuracy from 17% to 85%. The advantage is largest on held-out generators and long clips, and carries over to selecting generated demonstrations (80% vs. 71% precision).

Project page: `robogazeproject.github.io/RoboGazeProject`.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Recent advances in generative world models (hereafter *generators*) have made synthetic video a promising interface for embodied AI, enabling robot-centric rollouts for manipulation prediction, demonstration synthesis, and policy evaluation (Ali et al., 2025; Hu et al., 2023; Bruce et al., 2024; Yang et al., 2023; Hong et al., 2023). These rollouts increasingly stand in for teleoperated demonstrations and policy-evaluation episodes, so each individual clip must be correct. For robotics, however, visual realism alone is insufficient: generated videos must also be physically plausible, temporally coherent, and task-executable. We call a video *physically* plausible when the robot body and the objects persist coherently and their contact is supportable (no penetration, unsupported motion or teleportation). Such failures are often brief and local, like a hand passing through a cup for a fraction of a second, so a clip can look convincing overall yet contain exactly the error that matters, a gap between perceptual quality and robotics utility.

Existing evaluation methods do not fully capture these requirements. Perceptual metrics measure low-level visual fidelity, while general video evaluators such as VBench (Huang et al., 2024), DOVER (Wu et al., 2023), and VideoScore (He et al., 2024) primarily assess overall video quality rather than execution correctness. Robotics benchmarks such as RBench (Deng et al., 2026), WorldSimBench (Qin et al., 2024), and PAI-Bench (Zhou et al., 2025) evaluate more relevant physical or task properties, but mainly aggregate scores for model-level comparison rather than diagnose individual clips. Human evaluation (Zhang et al., 2026) is more reliable but costly to scale.

Recent VLM-based evaluators (He et al., 2025; Mou et al., 2025; Inbasekar et al., 2026) and world critics (Zheng et al., 2025; Zhang et al., 2025; Jiang et al., 2026) improve semantic and physical reasoning, yet typically produce holistic judgments and can over-report failures when explicitly prompted to search for them (§5.1). Event-level methods such as GLiDE (Zheng et al., 2026) and Spotlight (Chinchure et al., 2025) localize failures in time, but target game or text-to-video content, use fixed error categories, and do not condition on task instructions. Thus, robotics-specific per-video diagnosis remains underexplored: identifying *what* failed, *when* it failed, *why* it failed, and *how severe* the failure was.

To address this gap, we introduce **RoboGaze**, a training-free multi-agent VLM evaluator for robot video generation (Figure 1). RoboGaze is built around three key innovations: (i) *task- and scene-grounded context construction* that separates what the instruction expects from what the video shows and aligns the video with its subtasks; (ii) *routed, dimension-specific specialists* that localize failures within each subtask; and (iii) *a critic verifier* that validates, merges and reports failures with interpretable evidence. Grounding supplies the task context that event detectors lack, routing localizes each failure in time, and the critic suppresses the false reports of a monolithic judge. All three operate over a *failure schema* whose six dimensions cover the task (*Task Progress*, *Instruction Consistency*), the objects and scene (*Object–Scene Consistency*), the robot (*Robot-Body Consistency*), their interaction (*Physical Plausibility*) and whether the frames can decide at all (*Visual Quality*); they are refined into 30 observable failure types, which also define the annotation of our benchmark. Unlike every evaluator in Table 1, RoboGaze is at once training-free, robotics-specific and temporally localized.

To support this task, we construct **RoboGazeBench**, a human-validated benchmark of 914 generated robot-manipulation videos. Its clips come from five generators under four generation mechanisms, from robot-specific models (Cosmos-Predict2.5, DreamGen, DreamDojo) to general-purpose image-to-video models (Wan 2.2, Veo) that we held out from development. Where prior benchmarks judge a clip as a whole, every failure here is annotated from scratch under the schema above, with a temporal span, a severity score and a free-text description, so the benchmark tests whether an evaluator finds the right failure at the right time, not only whether it rates the clip as good or bad. We also measure how reliably annotators apply the schema and how much of the phenomenon it covers (§4).

Across eight frozen backbones, RoboGaze outperforms every baseline, including a published agentic detector given the same schema, the same rubric and a larger inference budget (§5.1); controls matched in information, compute and agentic design each close part of the gap, but none of them approaches its reliability on failure-free clips (§5.2). The advantage is largest on the two held-out generators and long clips, is confirmed by blinded human raters, and carries over to screening generated demonstrations (§5.3).

In summary, we make three contributions to robot-video evaluation:

- **RoboGaze.** We propose a training-free, model-agnostic multi-agent VLM evaluator that performs temporally grounded failure diagnosis and generates structured diagnostic reports instead of scalar quality scores.
- **RoboGazeBench.** A human-validated benchmark of 914 generated robot-manipulation videos from five generators (two held out from development) under four generation mechanisms, spanning 4.9 to 148.9 seconds: the longest horizon of any comparable evaluation set.
- **Evidence that structure drives the gain.** On eight proprietary and open-source VLMs, controls matched in information, compute and agentic design, a blinded human audit and a demonstration-screening study show that RoboGaze's gains come from how the evaluation is organised and carry over to data selection.

2 Related Work

Robot World Models and Generated Robot Video. Large-scale video diffusion systems such as CogVideo (Hong et al., 2023) enable high-fidelity synthetic videos but prioritize visual quality over physical executability (Wang et al., 2025). Cosmos-Predict (Ali et al., 2025) unifies multi-modal world generation with better instruction following and physical consistency, while DreamGen (Jang et al., 2025) and DreamDojo (Gao et al., 2026) extend robot-centric generation to zero-shot and dexterous tasks. Their rollouts are increasingly consumed downstream, standing in for teleoperated demonstrations and supplying predicted dynamics for policy learning, so the correctness of an individual clip matters. Persistent failure modes such as hallucinated physics and instruction-irrelevant rollouts therefore motivate evaluation beyond visual fidelity.

Evaluating Generated Video. Traditional metrics such as PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), and FVD (Unterthiner et al., 2018) measure perceptual or distributional similarity but largely miss physical plausibility, temporal consistency, and task correctness. More structured benchmarks, including VBench (Huang et al., 2024; Zheng et al., 2025; Wu et al., 2023; Han et al., 2025), introduce multi-dimensional evaluation, while learned judges such as VideoScore and VideoScore2 (He et al., 2024; 2025) rely on large-scale supervision. Recent methods instead prompt frozen VLMs for structured critique (Yang et al., 2025; Inbasekar et al., 2026; Mou et al., 2025; Qi et al., 2025), and physics-oriented benchmarks such as PAI-Bench (Zhou et al., 2025) and Morpheus (Zhang et al., 2025) characterize physical failures for model-level comparison.

However, these approaches primarily score an entire video or a predefined evaluation dimension. They provide limited support for diagnosing *where* a failure occurs, *what* type of failure it is, and *how severe* it is within an individual execution. This distinction is particularly important for robotics, where temporally grounded and physics-sensitive judgments remain challenging even for frontier multimodal models (Zhang et al., 2026).

Localized Diagnosis and Robotics-Specific Evaluation. Two lines approach per-video diagnosis. Generic video evaluators localize hallucinations or glitches at the event level (Li et al., 2025; Taesiri et al., 2024; Zheng et al., 2026; Chinchure et al., 2025), but are not instruction-aware and rely on fixed failure taxonomies whose reliability and coverage are typically unclear. Robot-video benchmarks instead assess structural consistency, action completeness, simulation alignment, or closed-loop success (Deng et al., 2026; Jiang et al., 2026; Qin et al., 2024), but primarily rank generative models through aggregate scores rather than diagnose individual executions, and simulation-based methods do not directly apply to offline real-robot videos.

RoboGaze bridges these directions with a robotics-specific, instruction-aware failure schema derived from prior physical and robot-video evaluation criteria (Zhou et al., 2025; Zhang et al., 2025; Deng et al., 2026; Jiang et al., 2026; Qin et al., 2024). It measures the schema's reliability and coverage and, without training or simulation, produces structured per-video diagnoses with temporal localization and severity-aware evidence.

Table 1: Comparison of related video evaluation methods. RoboGaze (last row) is the only training-free method that provides robotics-specific, temporally localized diagnostic reports with severity, task-progress and physical-interaction checks. Columns are defined below the table.

Method	Focus	Eval. clip length	Temp. loc.	Diag.	Sev.	Task prog.	Phys. inter.	Robot.	Train-free	Output
VBench (Huang et al., 2024)	Gen. quality	2.0 s	✗	✗	✗	✗	✗	✗	✓	Visual score
VideoScore (He et al., 2024)	Gen. quality	1.0–3.0 s	✗	◑	✗	✗	◑	✗	✗	Multi-dim. score
WORLDJEN (Inbasekar et al., 2026)	Multi-dim. eval.	∼5–8 s	✗	◑	✗	◑	✓	✗	✓	Multi-dim. score
GRADEO (Mou et al., 2025)	Human-like eval.	1.0–5.1 s	✗	◑	◑	◑	◑	✗	✗	Semantic score
VideoGen-Eval (Yang et al., 2025)	Structured eval.	–	◑	◑	✗	◑	◑	✗	✓	Multi-dim. score
PAI-Bench (Zhou et al., 2025)	Physical realism	∼2–8 s	✗	◑	✗	◑	✓	◑	✓	Physics score
VideoHallu (Li et al., 2025)	Hallucination	5.3 s avg.	✗	✓	✗	✗	✓	✗	✓	Hallu. report
GlitchBench (Taesiri et al., 2024)	Glitch detection	–	✗	✓	✗	✗	◑	✗	✓	Glitch report
GLiDE (Zheng et al., 2026)	Game glitch (open)	19.1/60 s†	✓	✓	✗	✗	◑	✗	✓	Glitch report
Spotlight (Chinchure et al., 2025)	Gen. error loc.	∼5–10 s	✓	✓	✓	◑	✓	✗	✓	Error report
WorldSimBench (Qin et al., 2024)	World model eval.	–	✗	◑	✗	✓	◑	✓	✗	Benchmark score
RBench (Deng et al., 2026)	Robot video eval.	4–8 s*	✗	◑	✗	✓	✓	✓	✓	Robotic score
RoboWM-Bench (Jiang et al., 2026)	Robot exec. eval.	5–25 s	◑	◑	✗	✓	◑	✓	◑	Exec. score
Morpheus (Zhang et al., 2025)	Physical reasoning	–	✗	◑	✗	✗	✓	✗	✓	Physics score
RoboGaze	**Robot diagnosis**	**4.9–148.9 s**	✓	✓	✓	✓	✓	✓	✓	**Diagnostic report**

†Average/maximum. *RBench range covers configurations with stated durations; several robotics-specific configurations report frame rate only. Among benchmarks that report duration, RoboGazeBench has the longest maximum horizon (148.9 s); GlitchBench is image-only. *Temp. loc.*: localizes failures to time spans. *Diag.*: names what failed, beyond a score. *Sev.*: grades how serious each failure is. *Task prog.*: checks whether the instructed task or subtask is achieved, beyond matching the prompt's content. *Phys. inter.*: checks physical laws, contact or object–robot interaction. *Robot.*: designed for, or evaluated on, robot-manipulation video. ✓: yes; ✗: no; ◑: limited, i.e., covered partially or only as one term of a holistic score. –: not reported.

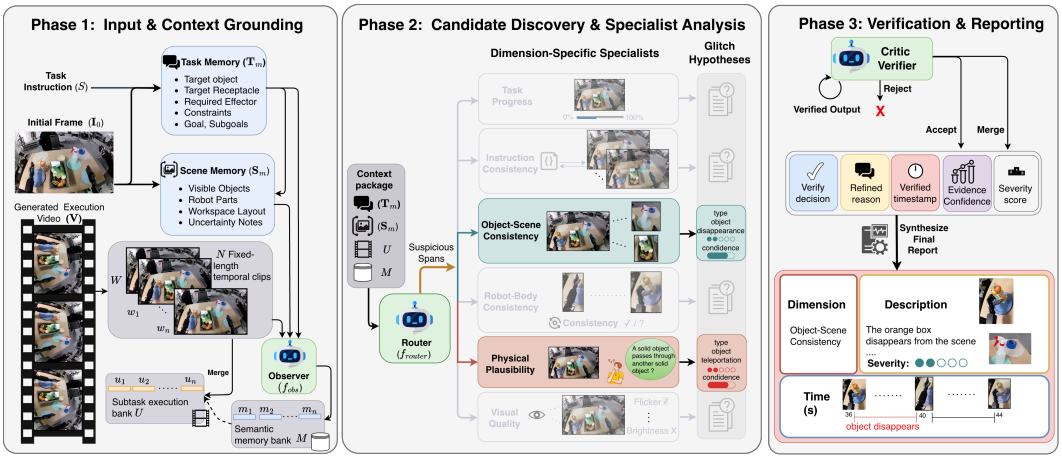


Figure 2: RoboGaze Overview. A three-phase pipeline for video generation diagnosis: (1) extracting task and scene context memories; (2) routing suspicious temporal spans to six dimension-specific specialists to generate failure hypotheses; and (3) verifying and synthesizing hypotheses into a final structured diagnostic report.

3 Method

This section presents **RoboGaze**, a multi-agent framework for fine-grained evaluation of robot video generation. Given a task instruction, an initial scene observation, and a generated robot execution video, RoboGaze detects execution failures and produces structured diagnostic reports with failure types, supporting evidence, and likely causes. As shown in Figure 2 and pseudo code in appendix A, RoboGaze decomposes anomaly reasoning into three phases, *(1) Input and Context Grounding* (§3.2), *(2) Candidate Discovery and Specialist Analysis* (§3.3) and *(3) Verification and Reporting* (§3.4), all organised around the failure schema of §3.1.

3.1 Failure Schema

Scope. We use *failure* for any observable deviation of a generated video from a correct, physically plausible execution of the instructed task, including rendering conditions that make the execution visually undecidable. This is a property of the video, not of a deployed policy: an empty report means

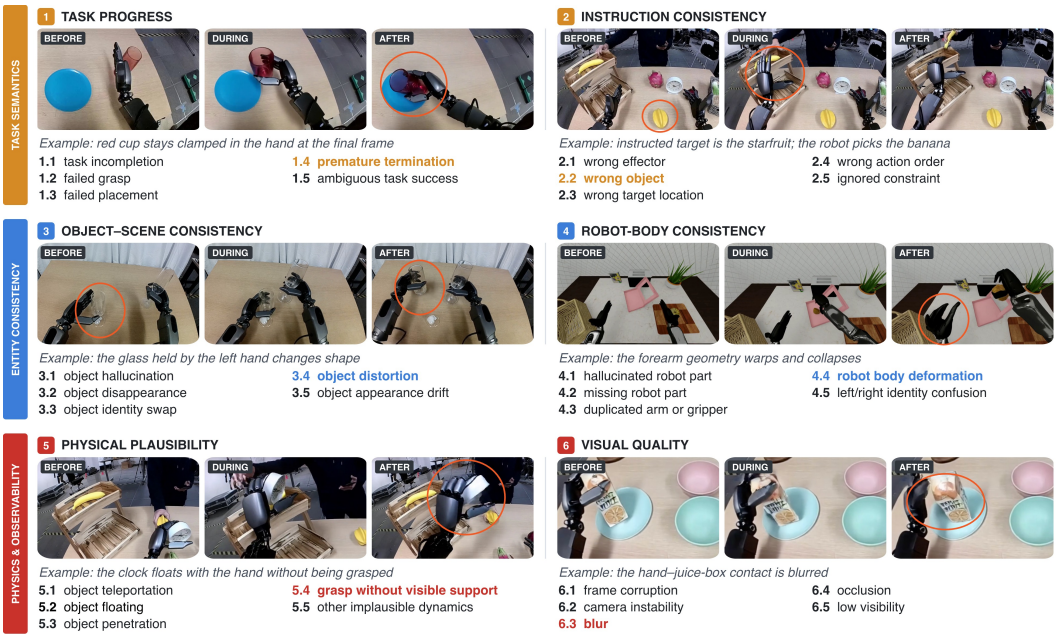


Figure 3: The RoboGazeBench failure schema. Three groups (rows), each holding two dimensions, and each dimension refined into five failure types. One unmodified benchmark event per dimension is shown before, during and after the failure, and its type is highlighted. Definitions, per-type counts and the severity rubric are in Appendix B.

that no failure is visible in the sampled frames, not that the depicted behaviour is executable on a robot. Safety, action feasibility and downstream policy harm are outside the schema.

Structure. Failures are organised in three levels (Figure 3). Three *groups*, the rows of the figure, state the question being asked: whether the video performs the right task (*task semantics*), whether objects and the robot persist coherently (*entity consistency*), and whether the result is physically supportable and visually decidable (*physics and observability*). Each group holds two *dimensions*, the unit of evaluation: one specialist handles each (§3.3), and results are aggregated over them because that is where annotators agree (§4). Each dimension is refined into five *types*, the unit of explanation, which a report names for every failure; three of them record that the frames cannot decide rather than asserting a defect, so abstention is explicit. The schema was built top-down from prior taxonomies and bottom-up from open-coded generated clips; its construction, type definitions and exclusion rules are in Appendix B.

3.2 Input and Context Grounding

Given a task instruction S, an initial observation I_0, and a generated execution video V, RoboGaze first converts raw multimodal inputs into structured contextual memories for anomaly diagnosis, because directly prompting a VLM on long-horizon robot videos often yields weak temporal localization and ambiguous task reasoning amid large amounts of irrelevant visual context.

Specifically, RoboGaze parses S and I_0 into a *task memory* T_m, encoding the global task objective o and an ordered list of K subtask specifications, $T_m = (o, \{t_1, \ldots, t_K\})$, where each t_k lists the manipulated objects, target state, expected action outcome and completion criterion of subtask k. RoboGaze also constructs a *scene memory* S_m conditioned on T_m and I_0, representing the initial workspace configuration, visible objects, robot body parts, spatial relationships, and uncertainty cues from occlusion or ambiguous object states. Together, $T_m = f_{\text{task}}(S, I_0)$ and $S_m = f_{\text{scene}}(I_0, T_m)$, where f_{task} and f_{scene} denote VLM-based semantic parsers, provide semantic grounding for task intent and physical grounding for scene feasibility.

To capture localized execution dynamics, RoboGaze decomposes V into N non-overlapping fixed-length temporal windows $\mathcal{W} = \{w_1, \ldots, w_N\}$. Instead of directly localizing subtasks over the full

video, RoboGaze converts localization into window-level task-progress reasoning. Each window is jointly analysed with T_m and S_m to produce a structured semantic observation:

$$\mathcal{M} = \{m_i\}_{i=1}^{N}, \quad m_i = f_{\text{obs}}(w_i, T_m, S_m), \tag{1}$$

where m_i summarizes robot actions, object state transitions, task progress signals, anomalies, and uncertainty cues within window w_i. Using these progress signals, RoboGaze groups temporally related windows into subtask-specific execution segments:

$$\mathcal{U} = \{u_k\}_{k=1}^{K}, \quad u_k = \text{Merge}(\{w_i \mid i \in \mathcal{I}_k\}), \tag{2}$$

where $\mathcal{I}_k$ denotes the windows whose progress signals assign them to subtask t_k, so segment u_k and specification t_k refer to the same subtask, and $\text{Merge}(\cdot)$ combines temporally contiguous windows into a unified sub-video. Together these form the context package $\mathcal{C} = \{T_m, S_m, \mathcal{M}, \mathcal{U}\}$.

3.3 Candidate Discovery and Specialist Analysis

Given the context package $\mathcal{C}$, RoboGaze performs targeted diagnosis with six *specialists*, one per failure dimension, each reasoning only over its dimension's five types (§3.1). Since only a subset of failure dimensions is typically relevant to each subtask, running all six specialists on every segment introduces noisy or weakly grounded diagnoses; RoboGaze therefore first routes each segment to the relevant specialists.

For each subtask execution segment $u_k \in \mathcal{U}$, RoboGaze retrieves the corresponding subtask specification t_k from T_m (§3.2). A candidate router VLM then scores how relevant each dimension is to the segment and keeps the most relevant ones:

$$\mathbf{p}_k = f_{\text{router}}(u_k, t_k) \in [0,1]^6, \qquad G_k^{\text{cand}} = \text{TopK}(\mathbf{p}_k, 3) \subseteq \mathcal{G}, \tag{3}$$

where $\mathcal{G}$ denotes the six failure dimensions, so G_k^{cand} is a set of at most three dimensions rather than a vector. Routing lets each specialist examine a segment closely without being asked about failures the segment cannot exhibit (e.g., a grasp segment is not checked for action order); removing the router costs 5.1 description F_1 (§5.2).

For each candidate dimension $g \in G_k^{\text{cand}}$, the corresponding specialist f_g returns a possibly empty set of hypotheses, one for each failure of its dimension it finds in the segment:

$$\mathcal{D}_{k,g} = f_g(u_k, t_k, S_m) = \{d_j\}, \qquad d_j = (y_j, \tau_j, r_j, e_j, \tilde{s}_j, c_j), \tag{4}$$

where $y_j \in \mathcal{T}_g$ is one of the five types of dimension g, τ_j localizes the failure temporally, r_j provides diagnostic reasoning, e_j summarizes supporting evidence, $\tilde{s}_j \in \{1, \ldots, 5\}$ is a proposed severity, and $c_j \in [0,1]$ is the specialist's confidence. The hypotheses of all specialists, $\mathcal{D} = \bigcup_k \bigcup_{g \in G_k^{\text{cand}}} \mathcal{D}_{k,g}$, are passed to verification.

3.4 Verification and Reporting

The hypotheses $\mathcal{D}$ produced in Phase 2 may still be uncertain, weakly grounded, or redundant across dimensions, so a critic verifier re-examines each hypothesis $d \in \mathcal{D}$ from segment u_k against the execution segment, diagnostic reasoning, temporal localization, supporting evidence, and task–scene grounding, and assigns one of three actions together with its own confidence $\hat{c}$:

$$f_{\text{verify}}(d, u_k, t_k, S_m) \;\rightarrow\; (a, \hat{c}), \quad a \in \{\textsc{accept}, \textsc{reject}, \textsc{merge}\}, \; \hat{c} \in [0,1]. \tag{5}$$

A hypothesis is ACCEPTed if it is visually supported, temporally consistent, semantically compatible with the task and scene, and its confidence exceeds the fixed threshold $\tau_c = 0.50$ selected on development data (Appendix E.6). It is REJECTed when the evidence is better explained by normal execution (e.g., occlusion rather than disappearance), and MERGEd when it refers to the same physical event as another hypothesis, based on matching objects or actions, overlapping spans, and consistent evidence. Independent failures remain separate even when their spans overlap.

Each surviving hypothesis is represented as $\hat{d} = (y, \hat{\tau}, \hat{r}, \hat{e}, s)$, containing its type, verified span, refined explanation, validated evidence, and severity. The accepted and merged hypotheses form the final report $R = f_{\text{report}}(\hat{\mathcal{D}})$, a structured list of failure events with their dimension, type, time span, severity, description, and evidence (Figure 1, left; Appendix C.6). This verification stage filters spurious specialist predictions before reporting and is key to RoboGaze's reliability on clean clips (Section 5.1).

4 EXPERIMENTAL SETUP

Benchmark. ROBOGAZEBENCH is a set of generated robot-manipulation videos in which every visible failure is annotated from scratch under the schema of Section 3.1, with a temporal span, a severity and a free-text description (Table 2, Figure 4). Two of its generators, Wan 2.2 and Veo (160 clips), were held out from all development and differ in architecture from the Cosmos-family backbone the others share, and its clips reach the longest horizon of any comparable evaluation set (Table 1).

Table 2: Key statistics.

Clips (failure-free)	914 (210)
Annotated failure events	1,847
Total duration	5.0 h
Instructions	194
Generators (mechanisms)	5 (4)
Embodiments	3
Duration, median (mean)	10.4 (19.8) s
Duration, min–max	4.9–148.9 s
Clips $>$ 40 s	130 (14.2%)

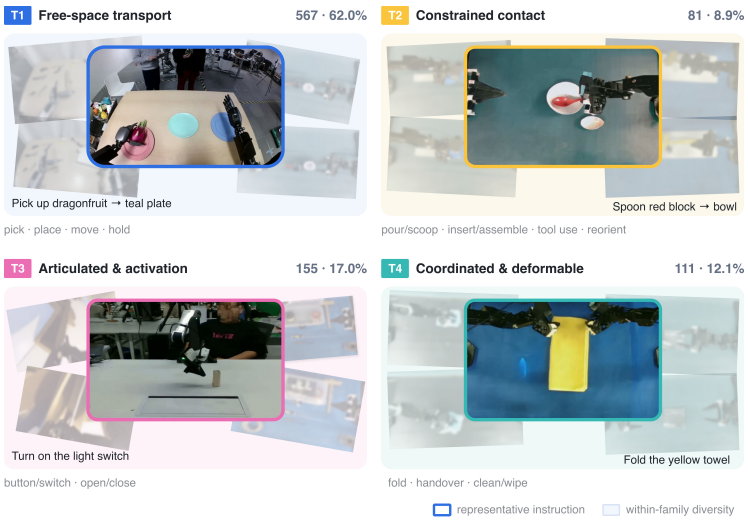


Figure 4: Task families, clip counts and example clips (Appendix C.2).

Annotation and reliability. Each clip is labelled independently by two robotics-trained annotators and merged by a senior annotator; a 180-clip *gold subset*, labelled by three senior annotators, supplies the human reference. Annotators agree on whether a clip contains a failure at Cohen's $\kappa = .74$, and more on the dimension of a failure than on its exact type, so results are aggregated by dimension (Appendices C and B).

Backbones and baselines. All methods use the same eight frozen VLMs, four proprietary and four open-weight (8–38B; Table 3), with identical visual inputs within each backbone, enabling controlled within-backbone comparisons (Appendix D.1).

To rule out simpler explanations for gains over direct prompting, we control for additional information, inference budget, and generic agentic reasoning. B1 receives the full RoboGaze context (schema, severity rubric, initial scene, and admissible normal behaviour) in a single call; B2 and B3 increase inference through self-consistency and generic propose-and-verify, respectively. B4a uses GLiDE (Zheng et al., 2026), the closest published agentic detector, while B4b (GLiDE$^+$) augments GLiDE with our schema and rubric. As the strongest and most expensive control, GLiDE$^+$ is our primary comparator in Sec. 5 (Appendix E.1). Learned scalar evaluators such as VideoScore2 (He et al., 2025) do not output localized events, so we evaluate them separately on candidate-clip ranking.

Evaluation protocol. A diagnostic report is a set of typed, time-stamped events, so we score it event by event with the open-ended protocol of GLiDE (Zheng et al., 2026), whose glitch reports have the same form. A fixed LLM judge (Gemini 3.1 Pro, temperature 0; $r = .79$ with human ratings) rates the similarity $S_{ij} \in [0, 1]$ of a predicted and a reference description, and Hungarian assignment (Kuhn, 1955) pairs events one-to-one by similarity times temporal overlap IoU_t (Appendix D.2). Description F_1 asks *what* failed: precision and recall sum S_{ij} over matched pairs and divide by the number of predicted and of reference events, so an invented event lowers precision and a missed one recall. Temporal mIoU asks *when*, as the mean overlap of matched spans, and $F_1 \times$IoU asks both at once, crediting each match by $S_{ij} \cdot \text{IoU}_t$. Two judge-free metrics score what our schema adds: clean accuracy, the fraction of the 210 failure-free clips given an empty report, and severe-event recall (Sev-R), the fraction of severity-4/5 reference events recovered. All are averaged over clips; intervals are paired 95% bootstrap intervals over source conditions.

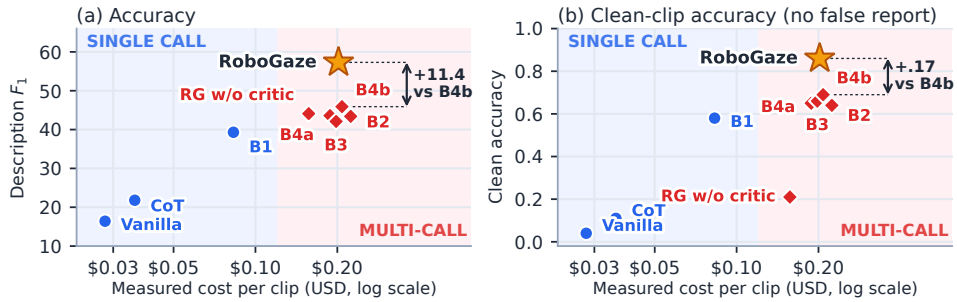


Figure 5: Structure, not information, compute or agentic design, separates the methods (Gemma4-31B, all 914 clips; controls in §4; full values in Appendix E.1).

5 Results

5.1 RQ1: Does Structured Diagnosis Beat Strong Evaluators?

Table 3: Event diagnosis on RoboGazeBench (914 clips). RG is RoboGaze; GLiDE$^+$ is the schema-equipped GLiDE control (B4b). Blue marks the best score overall and orange the best among open-weight backbones; ♦ ♦ ♦ rank the top three. F_1 and $F_1 \times$IoU are percentages; mIoU and Clean are fractions.

Backbone	Size	Desc. F_1 ↑			mIoU ↑			$F_1 \times$IoU ↑			Clean ↑		
		CoT	GLiDE$^+$	**RG**	CoT	GLiDE$^+$	**RG**	CoT	GLiDE$^+$	**RG**	CoT	GLiDE$^+$	**RG**
Proprietary backbones													
♦ Gemini 3.1 Pro	–	30.2	54.1	**66.4**	.41	.53	**.63**	12.9	30.1	**42.4**	.19	.74	**.91**
♦ GPT-5.5	–	28.6	52.0	**64.1**	.40	.52	**.62**	12.0	28.4	**40.3**	.16	.72	**.89**
♦ Claude Sonnet 4.6	–	25.7	48.4	**60.2**	.39	.50	**.60**	10.5	25.6	**36.7**	.18	.70	**.88**
Gemini 3.1 Flash-Lite	–	24.1	46.3	**57.8**	.38	.49	**.58**	9.6	23.7	**34.1**	.14	.67	**.85**
Open-weight backbones													
Gemma4-31B	31B	21.8	45.9	**57.3**	.36	.51	**.62**	8.2	24.8	**36.1**	.11	.69	**.86**
Qwen3.6-35B	35B	20.6	43.1	**53.9**	.35	.47	**.57**	7.6	21.6	**31.3**	.21	.66	**.84**
InternVL3.5-38B	38B	19.2	40.2	**50.6**	.34	.45	**.55**	6.8	19.0	**28.4**	.12	.63	**.81**
LLaVA-OV-2-8B	8B	17.9	38.0	**48.1**	.32	.43	**.52**	6.0	17.2	**25.6**	.23	.60	**.79**
Mean		*23.5*	*46.0*	**57.3**	*.37*	*.49*	**.59**	*9.2*	*23.8*	**34.4**	*.17*	*.68*	**.85**
Human agreement†	–		*74.6*			*.69*			*52.2*			*.93*	

†Senior annotators against each other, 180-clip gold subset; model scores there are within 0.9 F_1 of full-benchmark values (Appendix E.4).

RoboGazeBench is hard, and the gain is consistent across backbones. With chain-of-thought prompting (Table 3), no backbone exceeds 30.2 description F_1 or returns an empty report on more than 23% of failure-free clips, and even GLiDE$^+$ stays 20.5 F_1 below human agreement on the strongest backbone (54.1 against 74.6). RoboGaze is best on every metric and every backbone: averaged over the eight backbones, description F_1 rises from 23.5 with CoT and 46.0 with GLiDE$^+$ to 57.3, a paired gain of $+11.3$ $[10.2, 12.4]$ that stays within $+10.1$ to $+12.3$ although the backbones span 18.3 points. It is largest on the strongest backbone, so the structure adds to what a stronger model knows rather than compensating for a weaker one.

The widest separation is on clean clips; the residual gap sits where annotators disagree. With no judge involved, CoT returns an empty report on 17% of failure-free clips, GLiDE$^+$ on 68% and RoboGaze on 85% (91% on Gemini 3.1 Pro). On the gold subset, RoboGaze on Gemini 3.1 Pro recovers 83% of the distance between CoT and human agreement in description F_1, against 55% for GLiDE$^+$, and 99% of it in clean accuracy (Appendix E.4). The residual concentrates in Physical Plausibility and Visual Quality and tracks inter-annotator agreement across the six dimensions ($\rho = -.94$; Appendix E.3): part of the ceiling reflects what sampled RGB frames can settle, not the evaluator (error atlas in Appendix F).

5.2 RQ2: Where Does the Gain Come From?

Each control closes part of the gap, but none matches it on clean clips. On Gemma4-31B (Figure 5; same ordering on Gemini 3.1 Pro, Appendix E.1), a single call given everything RoboGaze is told (B1) closes 49% of the F_1 distance from CoT to RoboGaze, but its clean accuracy stays at

.58 against .86: the schema tells the model what counts as a failure, not whether the frames show one. Spending more on the same evidence adds little: nine self-consistent samples (B2) gain 4.1 F_1 over B1, and a propose-and-verify loop (B3) does no better. GLiDE given our schema and rubric (B4b) is the strongest control, closing 68%, yet at comparable cost RoboGaze still leads it by 11.4 F_1 and .17 clean accuracy.

Grounding and routing find events; the critic removes false ones. Removing task and scene memory or the candidate router costs 6.8 and 5.1 F_1 and .05 and .04 severe-event recall (Table 4). Removing the critic costs 13.2 F_1 and collapses clean accuracy from .86 to .21, yet is the only removal that *raises* severe-event recall (+.03); the ordering holds over thresholds .30–.70 (Appendix E.6).

Table 4: Component ablation, Gemma4-31B; all eight configurations in Appendix E.3.

Configuration	F_1	Sev-R	Clean
RoboGaze	**57.3**	.91	**.86**
w/o task & scene mem.	50.5	.86	.79
w/o candidate router	52.2	.87	.82
w/o critic verifier	44.1	**.94**	.21

5.3 RQ3: Does It Hold Beyond the Development Conditions?

The margin grows on held-out generators and long clips. Both methods score lower on Wan 2.2 and Veo, held out when prompts and thresholds were chosen, but GLiDE$^+$ loses more: the gain is +12.4 and +13.0 F_1 against +11.3 overall (Figure 6a). It also rises monotonically with clip length, from +10.2 below fifteen seconds to +16.6 above eighty (Figure 6b), and all eighteen slice intervals exclude zero (Appendix E.2). On long clips sixteen uniform frames skip 68% of reference events, which RoboGaze reads window by window; more frames barely help GLiDE$^+$ (Appendix E.6).

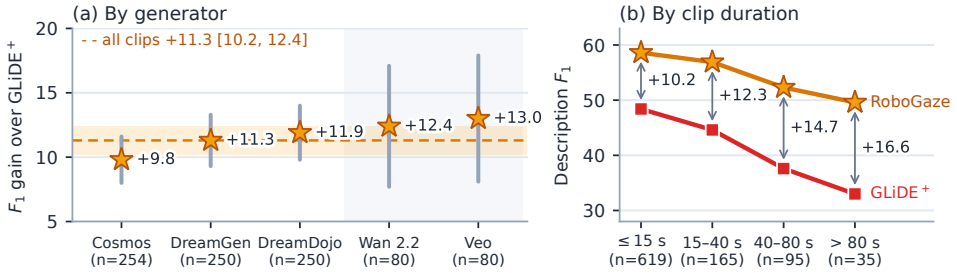


Figure 6: The margin is largest where the benchmark is hardest (pooled over eight backbones). (a) Paired F_1 gain over GLiDE$^+$ by generator, 95% intervals; shaded: held out. (b) F_1 by clip duration (Appendix E.2).

The scores do not depend on the language judge. Three blind senior annotators re-scoring RoboGaze and GLiDE$^+$ (Gemma4-31B) on the gold subset give 71.6 against 61.4 event F_1, a margin of +10.2 close to the automatic +11.4 on the same clips (Appendix C.12). Re-scoring every cell with GPT-5.5 preserves the ordering ($\rho = .95$, largest change 1.9 points; Appendix E.6).

The reports also select better demonstrations. Ranking six renderings of each of 40 conditions outside the benchmark, with usability labelled by blind annotators, RoboGaze keeps usable clips in its top half at .80 precision, against .71 for GLiDE$^+$, .65 for VideoScore2 (He et al., 2025) and .62 for RBench, and halves surviving severe failures (.05 against .11; Appendix E.5).

6 Conclusion and Limitations

We presented RoboGaze, a training-free evaluator that turns frozen VLM judges into temporally localized robot-video diagnosticians, and RoboGazeBench, the longest-horizon benchmark of its kind. Evaluator *structure*, not information, compute or agentic design, drives a consistent gain that grows on held-out generators and long clips.

Limitations. RoboGaze reports *visible* failures: an empty report is not evidence of physical executability and does not replace human oversight before real-robot use. The residual gap to humans lies in *Physical Plausibility* and *Visual Quality*, where annotators also disagree most, suggesting that 3D or pixel-level signals are needed beyond sampled RGB frames. All clips are single-view, whereas recent robot world models also render multi-camera rollouts; judging cross-view consistency, which remains hard for current VLMs, is left to future work. Finally, the precision-oriented critic can suppress rare transient failures, and the multi-call design costs more than a single judge.

AI USE STATEMENT

Generative video models were used to create the synthetic videos that constitute ROBOGAZEBENCH, and vision–language models were used as the experimental systems and evaluators studied in this work, as described in Section 4 and Appendices C and D. Generative AI tools were also used to assist with literature review and to polish the wording of the manuscript. The authors checked cited works against their primary sources, verified numerical claims against the experiment records, and reviewed all AI-assisted text. The authors take responsibility for the complete content of this work.

ETHICS STATEMENT

This work studies generated robot videos and does not evaluate people or deploy an autonomous system in the physical world. Seven annotators with relevant technical backgrounds labelled the videos after training and calibration; they were paid above local hourly research-assistant rates. The activity was reviewed internally as non-human-subject work. Annotators saw only generated videos and task instructions and were blinded to generator and evaluator identity. Released annotation records use pseudonymous annotator identifiers and exclude directly identifying information. Source datasets are redistributed only under their original licences. The benchmark may contain generator- and task-dependent biases and does not certify physical executability or safety: false negatives remain possible, and its predictions should not replace human oversight in a real-robot setting. Further details on annotation, release contents, and intended use appear in Appendices C.7 and F.

REPRODUCIBILITY STATEMENT

Section 4 specifies benchmark construction, annotation, baselines, metrics, and statistical testing. The appendix records the generation and filtering procedure (Appendix C), exact model checkpoints and API access dates, frame budgets, decoding settings, retry policy, prompt versions and hashes, and compute configuration (Appendix D.1), as well as metric definitions and robustness analyses (Appendices D.2 and E.6). The release contains the benchmark manifest, prompts, configurations, model outputs, annotations, scoring code, and figure-generation scripts needed to reproduce the reported results; it will be made public upon acceptance.

REFERENCES

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical AI. *arXiv preprint arXiv:2511.00062*, 2025.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, et al. Why do multi-agent LLM systems fail? *arXiv preprint arXiv:2503.13657*, 2025.

Aditya Chinchure, Sahithya Ravi, Pushkar Shukla, Vered Shwartz, and Leonid Sigal. Spotlight: Identifying and localizing video generation errors using VLMs. *arXiv preprint arXiv:2511.18102*, 2025.

Yufan Deng, Zilin Pan, Hongyu Zhang, Xiaojie Li, Ruoqing Hu, Yufei Ding, Yiming Zou, Yan Zeng, and Daquan Zhou. Rethinking video generation model for the embodied world. *arXiv preprint arXiv:2601.15282*, 2026.

Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. DreamDojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

Hui Han, Siyuan Li, Jiaqi Chen, Yiwen Yuan, Yuling Wu, Yufan Deng, Chak Tou Leong, Hanwen Du, Junchen Fu, Youhua Li, et al. Video-Bench: Human-aligned video generation benchmark. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18858–18868, 2025.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, et al. VideoScore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2105–2123, 2024.

Xuan He, Dongfu Jiang, Ping Nie, Minghao Liu, Zhengxuan Jiang, Mingyi Su, Wentao Ma, Junru Lin, Chun Ye, Yi Lu, et al. VideoScore2: Think before you score in generative video evaluation. *arXiv preprint arXiv:2509.22799*, 2025.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. CogVideo: Large-scale pretraining for text-to-video generation via transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Karthik Inbasekar, Guy Rom, and Omer Shlomovits. WorldJen: An end-to-end multi-dimensional benchmark for generative video models. *arXiv preprint arXiv:2605.03475*, 2026.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. DreamGen: Unlocking generalization in robot learning through video world models. In *Conference on Robot Learning*, pp. 5170–5194. PMLR, 2025.

Feng Jiang, Yang Chen, Kyle Xu, Yuchen Liu, Haifeng Wang, Zhenhao Shen, Jasper Lu, Shengze Huang, Yuanfei Wang, Chen Xie, et al. RoboWM-Bench: A benchmark for evaluating world models in robotic manipulation. *arXiv preprint arXiv:2604.19092*, 2026.

Harold W Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Zongxia Li, Xiyang Wu, Guangyao Shi, Yubin Qin, Hongyang Du, Tianyi Zhou, Dinesh Manocha, and Jordan Boyd-Graber. VideoHallu: Evaluating and mitigating multi-modal hallucinations on synthetic video understanding. *Advances in Neural Information Processing Systems*, 38: 76046–76078, 2025.

Zhun Mou, Bin Xia, Zhengchao Huang, Wenming Yang, and Jiaya Jia. GRADEO: Towards human-like evaluation for text-to-video generation via multi-step reasoning. In *International Conference on Machine Learning*, pp. 44971–44996. PMLR, 2025.

Robert C. Nickerson, Upkar Varshney, and Jan Muntermann. A method for taxonomy development and its application in information systems. *European Journal of Information Systems*, 22(3): 336–359, 2013.

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

Yiran Qin, Zhelun Shi, Jiwen Yu, Xijun Wang, Enshen Zhou, Lijun Li, Zhenfei Yin, Xihui Liu, Lu Sheng, Jing Shao, et al. WorldSimBench: Towards video generation models as world simulators. *arXiv preprint arXiv:2410.18072*, 2024.

Mohammad Reza Taesiri, Tianjun Feng, Cor-Paul Bezemer, and Anh Nguyen. GlitchBench: Can large multimodal models detect video game glitches? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22444–22455, 2024.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Yimu Wang, Xuye Liu, Wei Pang, Li Ma, Shuai Yuan, Paul Debevec, and Ning Yu. Survey of video diffusion models: Foundations, implementations, and applications. *Transactions on Machine Learning Research*, 2025.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 20144–20154, 2023.

Yuhang Yang, Ke Fan, Shangkun Sun, Hongxiang Li, Ailing Zeng, FeiLin Han, Wei Zhai, Wei Liu, Yang Cao, and Zheng-Jun Zha. VideoGen-Eval: Agent-based system for video generation evaluation. *arXiv preprint arXiv:2503.23452*, 2025.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. UniSim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1389–1399, 2023.

Chenyu Zhang, Daniil Cherniavskii, Antonios Tragoudaras, Antonios Vozikis, Thijmen Nijdam, Derck WE Prinzhorn, Mark Bodracska, Nicu Sebe, Andrii Zadaianchuk, and Efstratios Gavves. Morpheus: Benchmarking physical reasoning of video generative models with real physical experiments. *arXiv preprint arXiv:2504.02918*, 2025.

Qin Zhang, Peiyu Jing, Hong-Xing Yu, Fangqiang Ding, Fan Nie, Weimin Wang, Yilun Du, James Zou, Jiajun Wu, and Bing Shuai. Physion-Eval: Evaluating physical realism in generated video via human reasoning. *arXiv preprint arXiv:2603.19607*, 2026.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Muyang Zheng, Tong Zhou, Geyang Wu, Zihao Lin, Haibo Wang, and Lifu Huang. Open-ended video game glitch detection with agentic reasoning and temporal grounding. *arXiv preprint arXiv:2604.07818*, 2026.

Fengzhe Zhou, Jiannan Huang, Jialuo Li, Deva Ramanan, and Humphrey Shi. PAI-Bench: A comprehensive benchmark for physical AI, 2025. URL https://arxiv.org/abs/2512.01989.

Appendix

This supplement provides additional details on the failure schema, benchmark construction and annotation, implementation and metrics, complete and robustness results, and further qualitative examples for RoboGaze.

(A) RoboGaze Pseudocode **14**

(B) Failure Schema **14**

(B.1) Scope, Construction and Status 14

(B.2) Type Definitions and Exclusion Rules 16

(B.3) Severity, Cascading Events and Primary Dimension 17

(B.4) Per-Type Event Counts 17

(B.5) Events the Schema Did Not Cover 18

(B.6) Coverage Check on Held-Out Generators 18

(C) Benchmark Construction and Annotation **19**

(C.1) Construction Overview 19

(C.2) Composition 19

(C.3) Generation Settings 20

(C.4) Sampling and Matched Design 21

(C.5) Filtering 21

(C.6) Released Record Format 22

(C.7) Annotation Protocol and Interface 22

(C.8) Annotation Effort 22

(C.9) Contamination 23

(C.10) Agreement 24

(C.11) Agreement at Three Granularities 24

(C.12) Blinded Human Audit 25

(D) Implementation and Metrics **26**

(D.1) Implementation 26

(D.2) Metrics 27

(E) Complete Results and Robustness **28**

(E.1) Baselines and Measured Cost 28

(E.2) Results by Slice 28

(E.3) Complete Ablations, Per-Dimension Scores, and Judge-Free Metrics 29

(E.4) Gold Subset 30

(E.5) Offline Candidate Screening 31

(E.6) Robustness Checks 31

(F) Qualitative Atlas, Release, and Ethics **32**

Algorithm 1 Input and Context Grounding

Input: Task instruction S, initial observation I_0, video V
Output: Context package $\mathcal{C} = \{T_m, S_m, M, U\}$

$T_m \leftarrow f_{\text{task}}(S, I_0)$
$S_m \leftarrow f_{\text{scene}}(I_0, T_m)$
$W \leftarrow \textsc{SplitWindows}(V, \Delta = 4\,\text{s})$ ▷ Non-overlapping windows
$M \leftarrow \emptyset$; $U \leftarrow \emptyset$
for each $w_i \in W$ **do**
    $m_i \leftarrow f_{\text{obs}}(w_i, T_m, S_m)$
    $M \leftarrow M \cup \{m_i\}$
end for
for each subtask t_k in T_m **do**
    $I_k \leftarrow \textsc{AssignWindows}(M, t_k)$ ▷ Use task-progress signals
    $u_k \leftarrow \textsc{MergeContiguous}(\{w_i \mid i \in I_k\})$
    $U \leftarrow U \cup \{u_k\}$
end for
return $\mathcal{C} = \{T_m, S_m, M, U\}$

Algorithm 2 Candidate Discovery and Specialist Analysis

Input: Context $\mathcal{C} = \{T_m, S_m, M, U\}$ from Algorithm 1; failure schema $\mathcal{G}$
Output: Failure hypotheses $\mathcal{D}$ with source contexts

$\mathcal{D} \leftarrow \emptyset$
for each subtask segment $u_k \in U$ **do**
    $t_k \leftarrow \textsc{RetrieveSubtask}(T_m, k)$
    $\mathbf{p}_k \leftarrow f_{\text{router}}(u_k, t_k)$
    $\mathcal{G}_k^{\text{cand}} \leftarrow \textsc{TopKDimensions}(\mathbf{p}_k, 3)$ ▷ At most three dimensions
    for each $g \in \mathcal{G}_k^{\text{cand}}$ **do**
        $\mathcal{D}_{k,g} \leftarrow f_g(u_k, t_k, S_m)$ ▷ Possibly empty
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(d, k, g) \mid d \in \mathcal{D}_{k,g}\}$ ▷ Retain provenance
    end for
end for
return $\mathcal{D}$

A RoboGaze Pseudocode

This section presents the pseudocode for the three phases of RoboGaze: input and context grounding (Algorithm 1), candidate discovery and specialist analysis (Algorithm 2), and verification and reporting (Algorithm 3). The phases communicate through a context package $\mathcal{C}$ and a set of failure hypotheses $\mathcal{D}$, producing a structured diagnostic report R. All VLM components use the same frozen backbone without task-specific training.

B Failure Schema

B.1 Scope, Construction and Status

Scope. The schema covers *visible* failures in generated RGB robot-manipulation video: any observable deviation of the video from a correct, physically plausible execution of the instructed task, including rendering conditions that make the execution visually undecidable. It deliberately does not cover action executability on hardware, safety, or downstream policy harm, and an empty report is a statement about the sampled frames rather than about the depicted behaviour. Severity is likewise defined on observable video properties, not on unmeasured downstream outcomes.

Construction. We assembled the schema before production annotation in three passes. First, top-down: we collected the evaluation criteria used by twelve prior benchmarks and taxonomies: general video-quality suites (Huang et al., 2024; Zheng et al., 2025; He et al., 2024), physical-plausibility and physical-reasoning benchmarks (Zhou et al., 2025; Zhang et al., 2025; 2026), event-level artifact detectors (Taesiri et al., 2024; Zheng et al., 2026; Chinchure et al., 2025), and robot-video

Algorithm 3 Verification and Reporting

Input: Hypotheses $\mathcal{D}$ from Algorithm 2; context $\mathcal{C}$ from Algorithm 1; thresholds $\tau_c = 0.50$, $\tau_{\text{IoU}} = 0.60$
Output: Structured diagnostic report R

$\mathcal{A} \leftarrow \emptyset$; $\mathcal{Q} \leftarrow \emptyset$
for each $(d, k, g) \in \mathcal{D}$ **do**
  $(u_k, t_k, S_m) \leftarrow \textsc{RetrieveContext}(\mathcal{C}, k)$
  $(a, \hat{c}) \leftarrow f_{\text{verify}}(d, u_k, t_k, S_m)$
  if $a = \textsc{Accept}$ **and** $\hat{c} \geq \tau_c$ **then**
    $\hat{d} \leftarrow \textsc{RefineVerifiedEvent}(d, u_k, t_k, S_m)$
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{\hat{d}\}$
  else if $a = \textsc{Merge}$ **then**
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(d, k, g)\}$ ▷ Resolve jointly
  else
    continue ▷ Discard rejected or low-confidence hypotheses
  end if
end for
$\widehat{\mathcal{D}} \leftarrow \textsc{ConsolidateVerifiedEvents}(\mathcal{A}, \mathcal{Q}, \mathcal{D}, \mathcal{C}, \tau_{\text{IoU}})$
$R \leftarrow f_{\text{report}}(\widehat{\mathcal{D}})$
return R

evaluators (Deng et al., 2026; Jiang et al., 2026; Qin et al., 2024), and kept every criterion expressible as an observable property of a single video. Second, bottom-up: two authors open-coded 120 generated clips drawn from the three development generators but held out of the benchmark, writing free-text descriptions of every failure and grouping them by the mechanism they named. Third, reconciliation: over three rounds we merged criteria that no annotator could separate in practice, split those that named two mechanisms, and wrote an exclusion rule for every pair that had been confused during the 30-clip calibration pass (Appendix C.7). The result, comprising definitions, exclusion rules and severity rubric, was frozen after calibration and before any production clip was annotated, and was not revised afterwards.

Design criteria. Three requirements follow from the setting rather than from taste, and they explain most of the schema's shape. Every category must be decidable from sampled RGB frames alone, because that is all an offline evaluator sees; where the frames do not decide, the schema has to be able to say so instead of forcing a label, which is why three types are evidence-limiting rather than assertions of defect. Neighbouring categories must be separable in practice and not merely in principle, which is why every type carries an exclusion rule naming the type that applies instead (Appendix B.2), written for the pairs annotators confused during calibration. And the vocabulary must be closed enough to aggregate over, yet open enough to measure its own gaps: free-text description alone, as used by open-ended detectors (Zheng et al., 2026), gives no category to report per-category reliability over and no way to state what the vocabulary fails to cover.

On the uniform width. Each dimension holding exactly five types is a property of the reconciliation pass, not a finding about robot-video failure. Merging criteria that no annotator could separate in calibration, and splitting those that named two mechanisms, drove the dimensions toward a similar granularity, and we fixed the width at that point so that the codebook could be read and applied uniformly. The types are not equally natural and we do not present them as such: per-type frequencies span 22 to 138 annotated events (Table 7), and one dimension, Visual Quality, carries most of the evidence-limiting mass (82 of 113 events). A schema chosen for balance rather than for the phenomenon would show the opposite.

Status. The schema is the annotation instrument of RoboGazeBench, not a claim to an exhaustive account of robot-video failure. We therefore report what an instrument should report: how often each type is used (Table 7), how reliably annotators apply it at each level (Appendix C.11), and what fraction of annotated events it failed to cover (Appendix B.5). Following standard practice for taxonomy development (Nickerson et al., 2013) and for empirically grounded failure taxonomies (Cemri et al., 2025), we treat coverage as a measured quantity rather than an assertion.

Abstention and the residual type. Three types are *evidence-limiting*: `ambiguous_task_success`, `occlusion` and `low_visibility` record that the sampled frames do not decide, rather than asserting that the generator produced a defect. They account for 113 events (6.1% of all annotated events). Keeping them inside the schema makes abstention reportable instead of silent, and it is part of why *Visual Quality* has the lowest inter-annotator agreement of the six dimensions (Appendix C.11). One type is a *residual*: `other_implausible_dynamics` applies only when none of the four named Physical Plausibility types does, and absorbs 13.7% of Physical Plausibility events. Neither category is scored differently; both are marked so that a reader can separate what the schema asserts from what it declines to decide.

B.2 Type Definitions and Exclusion Rules

Tables 5 and 6 give the released codebook. Each type carries a positive definition and an explicit rule naming the neighbouring type that applies instead; the pairs listed were the ones most often confused during calibration. Annotators worked from this text verbatim.

Table 5: Codebook, Groups A and B (part 1): Task Semantics and the object half of Entity Consistency.

Type	Applies when	Does not apply when
1. Task Progress (TP): *Does the task reach the instructed end state?*		
1.1 `task_incompletion`	The task is attempted but the required final state is never reached.	The episode stops early with the action sequence unfinished (→1.4), or a specific place action fails (→1.3).
1.2 `failed_grasp`	The robot attempts to grasp an object but never establishes a stable grasp.	The object follows the gripper with no contact (→5.4), or a grasp succeeds and the placement fails (→1.3).
1.3 `failed_placement`	An object is transported or manipulated but not brought to its intended final state.	No stable grasp was formed (→1.2), or the target itself is the wrong one (→2.3).
1.4 `premature_termination`	The episode ends before the task-relevant action sequence is complete.	The episode runs its full length without reaching the final state (→1.1).
1.5 `ambiguous_task_success`	*Evidence-limiting.* The final state is visually ambiguous, so success cannot be determined from the frames.	The frames do decide; then record the asserted type. If a nameable visual condition causes the ambiguity, also record 6.4 or 6.5.
2. Instruction Consistency (IC): *Does the execution follow the instruction?*		
2.1 `wrong_effector`	The robot uses an effector inconsistent with the instruction or expected embodiment.	Extra, absent or duplicated limbs are rendered (→4.1–4.3).
2.2 `wrong_object`	The robot interacts with an object other than the one specified.	The correct object changes identity mid-clip (→3.3).
2.3 `wrong_target_location`	The object is moved to a location inconsistent with the instruction.	The object reaches the correct target but is not stably placed (→1.3).
2.4 `wrong_action_order`	Required actions occur in an order inconsistent with the instruction.	A completed action repeats with no ordering constraint stated (→1.1).
2.5 `ignored_constraint`	A stated constraint, precondition or task-specific requirement is ignored.	The violated constraint names an effector, object, location or order; use the specific type (→2.1–2.4).
3. Object–Scene Consistency (OSC): *Do objects and the scene persist coherently?*		
3.1 `object_hallucination`	A task-relevant object appears without a physically supported cause.	The object enters from outside the frame (normal), or a robot part appears (→4.1).
3.2 `object_disappearance`	A visible object vanishes without being moved out of view; requires visible evidence that occlusion cannot explain.	Occlusion is the better explanation (→6.4).
3.3 `object_identity_swap`	An object changes semantic identity or is replaced by another object.	Only colour or texture changes (→3.5); only geometry changes (→3.4).
3.4 `object_distortion`	Geometry deforms inconsistently with the object's expected rigidity or articulation.	Surface colour or texture changes with stable geometry (→3.5); the deforming entity is the robot (→4.4).
3.5 `object_appearance_drift`	Colour or texture drifts over time without task-relevant cause, geometry remaining stable.	Geometry changes (→3.4); the whole frame degrades (→6.1, 6.3).

Table 6: Codebook (part 2): the robot half of Entity Consistency, and Physics & Observability.

Type	Applies when	Does not apply when
4. Robot-Body Consistency (RBC): *Does the robot keep its morphology and kinematics?*		
4.1 `hallucinated_robot_part`	A limb, gripper, finger or body segment appears with no physical continuity and no counterpart on the platform.	The part is an additional copy of one the platform does have (→4.3).
4.2 `missing_robot_part`	A visible robot part becomes absent when it should remain visible.	The part leaves the frame or is occluded (not a failure; record 6.4 if it blocks diagnosis).
4.3 `duplicated_arm_or_gripper`	An extra copy of a part the platform does have is rendered.	The part has no counterpart on the platform (→4.1).
4.4 `robot_body_deformation`	The robot bends, stretches, melts or changes shape beyond its kinematic structure.	The deforming entity is an object (→3.4).
4.5 `left_right_identity_confusion`	Left/right arms, grippers or body sides swap identity or become inconsistent over time.	A side disappears rather than swapping (→4.2).
5. Physical Plausibility (PP): *Is the depicted interaction physically supportable?*		
5.1 `object_teleportation`	An object changes position discontinuously with no visible transport or causal interaction.	The jump occurs across an occluder and is consistent with some occluded path (→6.4).
5.2 `object_floating`	An object remains suspended with no visible support, contact or plausible dynamics.	The object follows the gripper without contact (→5.4).
5.3 `object_penetration`	A solid object passes through another solid object, robot part or support surface; requires an apparent intersection.	Contact is merely occluded and ambiguous (→6.4).
5.4 `grasp_without_visible_support`	An object follows or moves with the robot despite no visible grasp, contact or support.	The robot attempted a grasp and the object stayed put (→1.2).
5.5 `other_implausible_dynamics`	*Residual.* Motion violates continuity, gravity, contact or inertia in a way none of 5.1–5.4 names.	Any of 5.1–5.4 applies. Use only after all four have been excluded.
6. Visual Quality (VQ): *Is the video artifact-free and readable?*		
6.1 `frame_corruption`	Frames contain rendering corruption, tearing, severe artifacts or missing visual content.	A localized geometry change affects one entity (→3.4, 4.4).
6.2 `camera_instability`	Camera shake, viewpoint jumps or unstable framing disrupt temporal interpretation.	The camera is stable and the scene layout itself changes (→3.x).
6.3 `blur`	Motion or defocus blur prevents reliable interpretation of task-relevant state.	Frames are globally dark or low-contrast (→6.5).
6.4 `occlusion`	*Evidence-limiting.* Occlusion or masking prevents reliable observation of the relevant robot–object interaction.	The frames do decide; record the asserted failure type instead.
6.5 `low_visibility`	*Evidence-limiting.* Lighting, resolution or contrast is too poor for reliable diagnosis.	Blur is localized and lighting is adequate (→6.3).

B.3 Severity, Cascading Events and Primary Dimension

Severity rubric. Severity is an ordinal score in $\{1, \ldots, 5\}$ defined on observable task-video properties rather than unmeasured downstream outcomes: 1 cosmetic (task state and object interaction remain interpretable); 2 minor (localized, little ambiguity about the intended execution); 3 moderate (affects an object, robot part or subtask transition relevant to the task); 4 severe (obscures whether the task executed correctly, or makes the clip unsuitable as a clean demonstration); 5 catastrophic (a physically or semantically impossible event that invalidates the execution). Severe-event recall counts events at severity 4 or 5; these are 558 events, 30.2% of the total (Table 7).

Cascading events. An event may implicate several dimensions: a gripper passing through a cup is both physical-plausibility and robot-body. Annotators assign all applicable dimensions and promote a primary one from the most directly observable evidence; matching uses the primary dimension and the manifest retains the rest. Cascading events are kept rather than discarded: 163 events across 114 clips (12.5%) implicate more than one dimension. All per-type counts below are over primary labels, so they sum to the event total. The label-agnostic setting $\lambda_{\text{dim}} = 0$ (Appendix E.6) discards the dimension entirely and is reported as a robustness check.

B.4 Per-Type Event Counts

Across the 914 clips, annotators recorded 1,847 failure events on 704 clips (77.0%); the remaining 210 clips (23.0%) carry no annotated event and form the failure-free set used for clean accuracy. A clip with at least one event carries 2.6 on average and 2 at the median. Table 7 gives the distribution over the thirty types. No type is unused, and the most frequent type (`task_incompletion`, 138) is roughly six times the least frequent (`ignored_constraint`, 22); we report the raw counts

rather than a balanced sample because the distribution is a property of the generators, not a design choice.

Table 7: Annotated failure events per type, over primary labels, all 914 clips. Dimension totals sum to 1,847.

Type		n	Type		n
1. Task Progress	412		**4. Robot-Body Cons.**	286	
1.1	task incompletion	138	4.1	hallucinated robot part	74
1.2	failed grasp	104	4.2	missing robot part	58
1.3	failed placement	87	4.3	duplicated arm or gripper	47
1.4	premature termination	52	4.4	robot body deformation	81
1.5	ambiguous task success	31	4.5	left/right identity confusion	26
2. Instruction Cons.	198		**5. Physical Plausibility**	379	
2.1	wrong effector	28	5.1	object teleportation	86
2.2	wrong object	61	5.2	object floating	71
2.3	wrong target location	54	5.3	object penetration	112
2.4	wrong action order	33	5.4	grasp without visible support	58
2.5	ignored constraint	22	5.5	other implausible dynamics	52
3. Object–Scene Cons.	341		**6. Visual Quality**	231	
3.1	object hallucination	63	6.1	frame corruption	44
3.2	object disappearance	98	6.2	camera instability	38
3.3	object identity swap	41	6.3	blur	67
3.4	object distortion	79	6.4	occlusion	49
3.5	object appearance drift	60	6.5	low visibility	33
Total events					**1,847**
Severity 1 / 2 / 3					329 / 448 / 512
Severity 4 / 5 (30.2% of events)					388 / 170

B.5 Events the Schema Did Not Cover

Annotators could mark an event `unresolved` when no type applied, so that a failure outside the schema would be recorded rather than forced into the nearest label. This occurred on 73 events (4.0% of all annotated events) across 61 clips (6.7%).

A senior adjudication pass re-examined every unresolved event against the codebook and sorted it into one of three outcomes. 52 of them (71%) resolved into an existing type once the existing boundary rule between the two candidates was applied explicitly, almost all on the `object_distortion` / `object_appearance_drift` and `failed_grasp` / `grasp_without_visible_support` pairs. These were application errors rather than coverage gaps: the exclusion rules in Tables 5 and 6 already decide them, and the frozen codebook was not changed. 9 were re-labelled as normal execution or annotator error and removed. 12 (16% of unresolved, 0.6% relative to the 1,847 typed events) remained outside the schema; they are released separately under an `other` category together with their free-text descriptions and are not counted in Table 7 or in any score. Their recurring kinds are illumination and shadow inconsistency not attached to any entity (5), behaviour of background agents who are not the manipulating robot (4), and 3 singletons.

We report this residual rather than suppressing it. A schema with no escape hatch cannot measure its own coverage, so a rate of exactly zero would be evidence about the annotation interface rather than about the schema; a measured 0.6% is what a reader needs in order to know how much of the phenomenon the thirty types account for. Illumination and shadow inconsistency is the clearest candidate for a type in a future revision, and it recurs in the held-out coverage check below.

B.6 Coverage Check on Held-Out Generators

The residual above is measured by annotators trained on the schema, so it is anchored by construction: a failure the schema does not name is also one they were not taught to look for. We therefore ran a second, schema-blind check on the held-out block. On 40 clips from Wan 2.2 and Veo, neither of which was used when the schema was frozen or the evaluator configured (Appendix C.9), annotators described every failure in free text without seeing the schema; a separate annotator then mapped each description to the thirty types.

Of 96 described failures, 81 (84%) mapped to exactly one type, 11 (11%) required two or more (cascading events, which the primary-dimension rule already handles), and 4 (4%) could not be

mapped, giving 96% coverage on generators the schema was not built against. One candidate type recurred across the unmapped descriptions, the same illumination and shadow inconsistency found in the residual analysis. We read this as the schema transferring to unfamiliar generators with a small, named and reproducible gap, rather than as completeness.

C Benchmark Construction and Annotation

C.1 Construction Overview

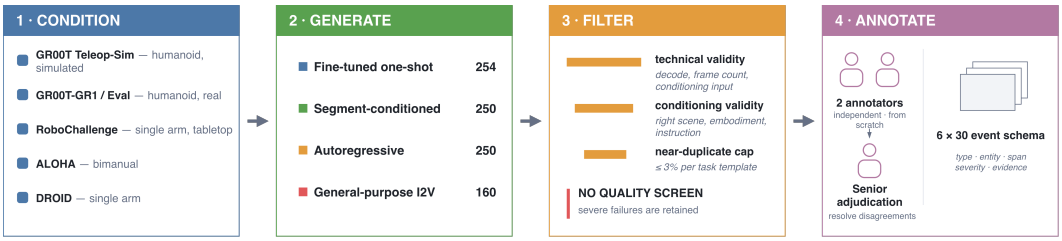


Figure 7: Construction of RoboGazeBench. Conditioning instructions and initial observations are drawn from five source datasets, rendered by five generators under four generation mechanisms, filtered for technical and conditioning validity but never for output quality, and annotated from scratch under the frozen event schema.

C.2 Composition

Table 8 gives the counts behind Figure 4, together with camera view and resolution, which the figure does not show. Example frames for the three embodiment and camera-view groups are shown in Figure 8. Humanoid clips are the largest embodiment group because the fine-tuned one-shot block is entirely GR1; simulation-derived conditioning comes from a single source inside that block, which is why the environment axis is unbalanced.

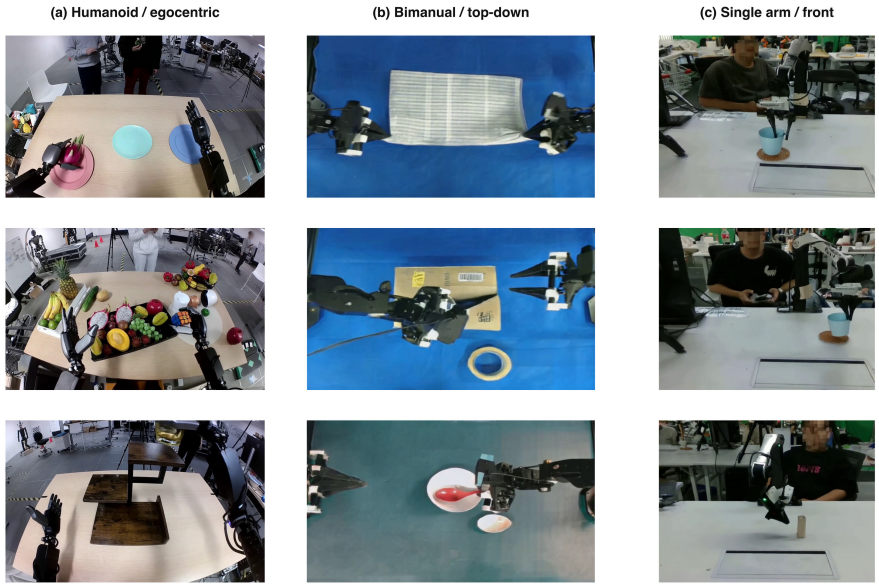


Figure 8: Example frames from RoboGazeBench, three per group: (a) humanoid with egocentric view, (b) bimanual with top-down view, and (c) single-arm with front view. Camera view is fixed by each source's capture rig and therefore coincides with embodiment.

Table 8: Counts along every reported axis. Each axis sums to 914. *Embodiment* is defined by the arms a clip uses, not by the platform, so a two-armed platform performing a one-armed task counts as single-arm. *Environment* describes the data a clip's conditioning came from; every clip is generated. *Camera view* is fixed by each source's capture rig and therefore coincides with embodiment.

Axis	Group	n
Mechanism	Fine-tuned one-shot	254
	Segment-conditioned	250
	Autoregressive	250
	General-purpose I2V	160
Generator	Cosmos-Predict2.5	254
	DreamGen	250
	DreamDojo	250
	Wan 2.2	80
	Veo	80
Embodiment	Humanoid	414
	Bimanual	250
	Single arm	250
Environment	Real-derived	760
	Simulation-derived	154
Horizon	≤ 15 s	619
	15–40 s	165
	40–80 s	95
	> 80 s	35
Task family	T1 free-space transport	567
	T2 constrained contact	81
	T3 articulated & activ.	155
	T4 coord. & deformable	111
Camera view	Egocentric	414
	Top-down	250
	Front	250
Resolution	1280×800	254
	1280×720	80
	832×480	80
	768×432	155
	640×480	345

C.3 Generation Settings

Three of the four mechanisms are realised on Cosmos-family backbones: Cosmos-Predict2.5-14B fine-tuned per task, Cosmos-Predict2 2B driven by DreamGen, and a 2B action-conditioned DiT for DreamDojo. RoboGazeBench therefore varies *generation mechanism* more than it varies backbone architecture, and Wan 2.2 and Veo are the only architecturally independent systems in it. We state this rather than let the pipeline names imply otherwise; the mechanism contrast is partly controlled because DreamGen and DreamDojo share source datasets and a backbone family, and architectural independence is what the held-out block supplies. Clip duration is set by each generator's rollout mechanism rather than by the condition alone: segment-conditioned DreamGen renders bounded segments, whereas autoregressive DreamDojo continues for as long as its conditioning actions run, which is why the two blocks differ in horizon (Table 9). Resolution and frame rate also differ by source, so a fixed frame budget corresponds to a different real-time sampling interval in each block; Appendix E.6 reports the result at half, the default, and double each backbone's frame budget.

Table 9: Per-source generation settings, grouped by generation mechanism. Each shaded header names the backbone and the sampler setting (classifier-free guidance / denoising steps) shared by every source under it. Durations are mean [min–max] where the horizon varies; frame counts are duration × FPS and are given per clip in the release. †Aggregate over the two sources above it.

Source	Embodiment	Resolution	FPS	Duration (s)	n
Fine-tuned one-shot Cosmos-Predict2.5-14B 7 / 35					
ft. GR00T Teleop-Sim	humanoid	1280×800	16	8.69	154
ft. GR00T-GR1 + Eval	humanoid	1280×800	16	5.80	100
Segment-conditioned (DreamGen) Cosmos-Predict2 2B 7 / 35, seed 0					
`sa_static_v2/90k`	single arm	640×480	16	23.06 [11.6–40.3]	95
`bimanual_16fps/110k`	bimanual	768×432	16	63.31 [11.6–103.6]	
`humanoid_16fps/90k`	humanoid	768×432	16	14.31 [11.6–17.3]	155†
Autoregressive (DreamDojo) action-conditioned DiT 2B 1.5 / 35					
`single_arm_sv/30k`	single arm	640×480	15	12.47 [8.9–38.5]	100
`aloha_resume70k_bimanual_sv/100k`	bimanual	640×480	10	49.30 [12.1–148.9]	
`2B_GR1_post-train/50k`	humanoid	640×480	10	9.80 [4.9–142.0]	150†
General-purpose I2V independent backbones, no robot-specific training					
Wan2.2-I2V-14B	all three	832×480	16	5.06	80
Veo	all three	1280×720	24	8.00	80
Total					**914**

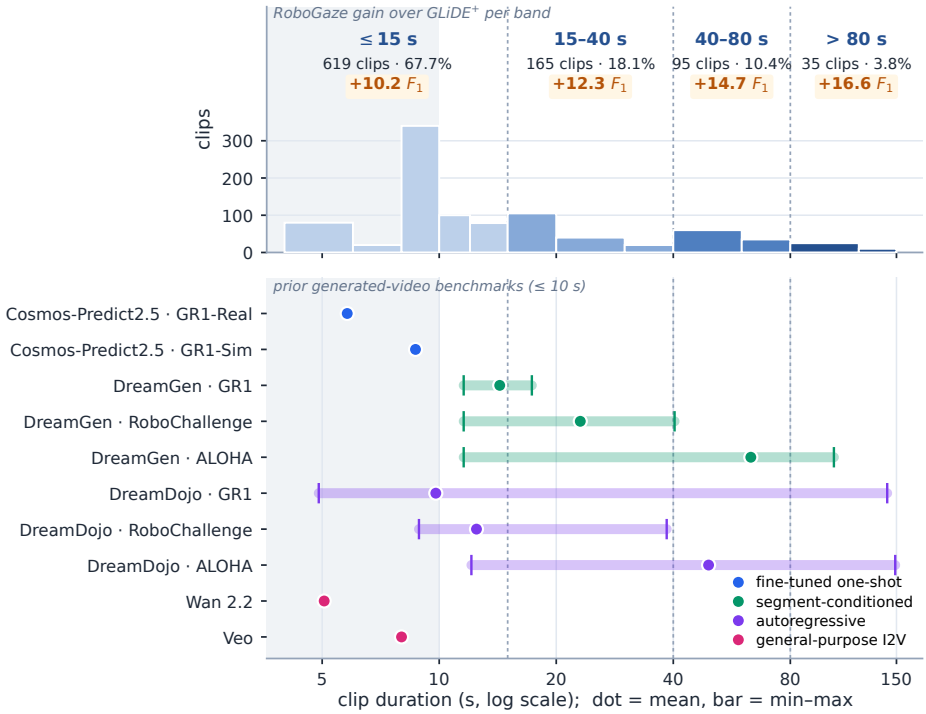


Figure 9: Horizon of RoboGazeBench. *Top:* clip-duration distribution coloured by horizon band, with each band's exact clip count and RoboGaze's gain over GLiDE$^+$ in that band printed above it. *Bottom:* the range and mean duration of each source, coloured by generation mechanism, on the same axis. The shaded region is the horizon at which prior generated-video benchmarks evaluate; only the robot-specific world models produce clips beyond it by design, which is why the held-out-generator and long-horizon analyses in Section 5 are reported separately. Within-bin shape reproduces the reported quantiles.

C.4 Sampling and Matched Design

Conditions were sampled before any generated output was viewed, so the benchmark cannot have been selected for failure density. Within each generator the sample is stratified over the source's task families and over its frame-count range. The second of these matters more than it appears: the bimanual sources are intrinsically the longest (Figure 9), so without stratifying on duration inside each source, the horizon analysis in Section 5.3 would be indistinguishable from an embodiment effect.

Wan 2.2 and Veo render the same 80 conditions. The other generators draw their conditions from overlapping source datasets but were sampled separately, by task family and duration, so their conditions are not paired one-to-one. That 914 clips draw on only 194 instructions reflects that one instruction is realised under several distinct initial observations and by several generators. Clips sharing a condition share a resampling group in the bootstrap (Appendix D.2), so repeated conditions do not inflate the effective sample size.

C.5 Filtering

A clip is rejected on two grounds only. It is technically invalid (a decode failure, a truncated file, a frame count outside the declared range, a missing conditioning input), in which case it is regenerated once with the same seed and dropped if it fails again. Or its conditioning is wrong: the clip does not depict the instructed scene, embodiment or task, judged by an annotator who sees the instruction and the first frame only. Near-duplicate conditions are detected by instruction and initial-frame similarity, and no task template exceeds 3% of the benchmark.

Nothing is removed for poor output. A generated video that teleports an object, deforms the gripper or abandons the task is what the benchmark exists to contain, and screening on quality would yield a benchmark on which every evaluator looks better than it is. One exception is historical: the 254

fine-tuned one-shot clips were quality-screened when first collected, before this policy was fixed. The screen makes them easier than the rest, so every result is reported per generator group (Appendix E.2) where its effect is visible.

C.6 Released Record Format

Clip fields are fixed at construction and event fields are produced by annotation and adjudication; Table 10 shows one record of each. A clip carrying no event is a *clean* clip and contributes only to clean accuracy.

Table 10: A single RoboGazeBench record, abbreviated. Each clip carries zero or more events; a clip with none is a *clean* clip and contributes to clean accuracy.

Clip	
clip_id	rgb_00417
instruction	*"pick up the glass and place it on the right tray"*
condition_id	aloha_sv_0132
generator	dreamgen | mechanism: segment_conditioned
checkpoint	bimanual_16fps/110k
source_dataset	aloha | environment: real_derived
embodiment	bimanual
task_family	T1_free_space_transport | fine_task: object_transport
camera_view	top_down
resolution / fps	768x432 / 16
frames / duration_s	712 / 44.50 | duration_band: long_40_80s
seed	0
filter_status	passed | regenerated: false
Event (1 of 2 on this clip)	
event_id	rgb_00417_e1
dimension	physical_plausibility | secondary: [object_scene]
type	object_penetration
entity	glass
span_frames / span_s	[498, 546] / [31.1, 34.1]
severity	4
description	*"the left gripper closes through the glass rim instead of around it; the glass follows the gripper with no contact"*
evidence	*"frames 502–538, left gripper and glass rim"*
flags	open_world: false | ambiguous: false
annotation	annotators: [a3, a5] | adjudicated_by: s1 | double_labelled: true

C.7 Annotation Protocol and Interface

Seven annotators with bachelor-level robotics, computer-vision or mechanical-engineering backgrounds completed a four-hour rubric training session, a 30-clip calibration pass and a senior review before production; three were designated senior on calibration agreement above $\kappa = 0.78$. Each clip is annotated independently from scratch by two annotators and merged by a senior annotator under a written adjudication rubric: detection disagreements are re-watched and resolved, dimension disagreements default to the cascade rule, spans with pairwise IoU ≥ 0.6 are merged by union and otherwise treated as separate events, and severity disagreements within ± 1 are averaged, otherwise re-rated from scratch; the manifest keeps the averaged value, and event counts and severe-event recall use it rounded up to the more severe level. Annotators see only the generated video and the task instruction, never a dataset identifier, a generator checkpoint or a model output, and the schema, rubric and guide were frozen before production annotation. Annotators were paid above local hourly research-assistant rates; the study uses generated video only and was reviewed internally as non-human-subject work.

C.8 Annotation Effort

Event annotation is slow because a clip must be watched more than once before a span can be fixed: an annotator views it for gist, again for candidate events, then scrubs each candidate at reduced speed to set boundaries at 0.1 s precision before writing the description and severity. A pass therefore costs roughly twenty-four times real time, and it scales with clip length rather than with clip count: a minute-long clip takes five to six times as long as a ten-second one. Table 11 accounts for the whole labelling effort.

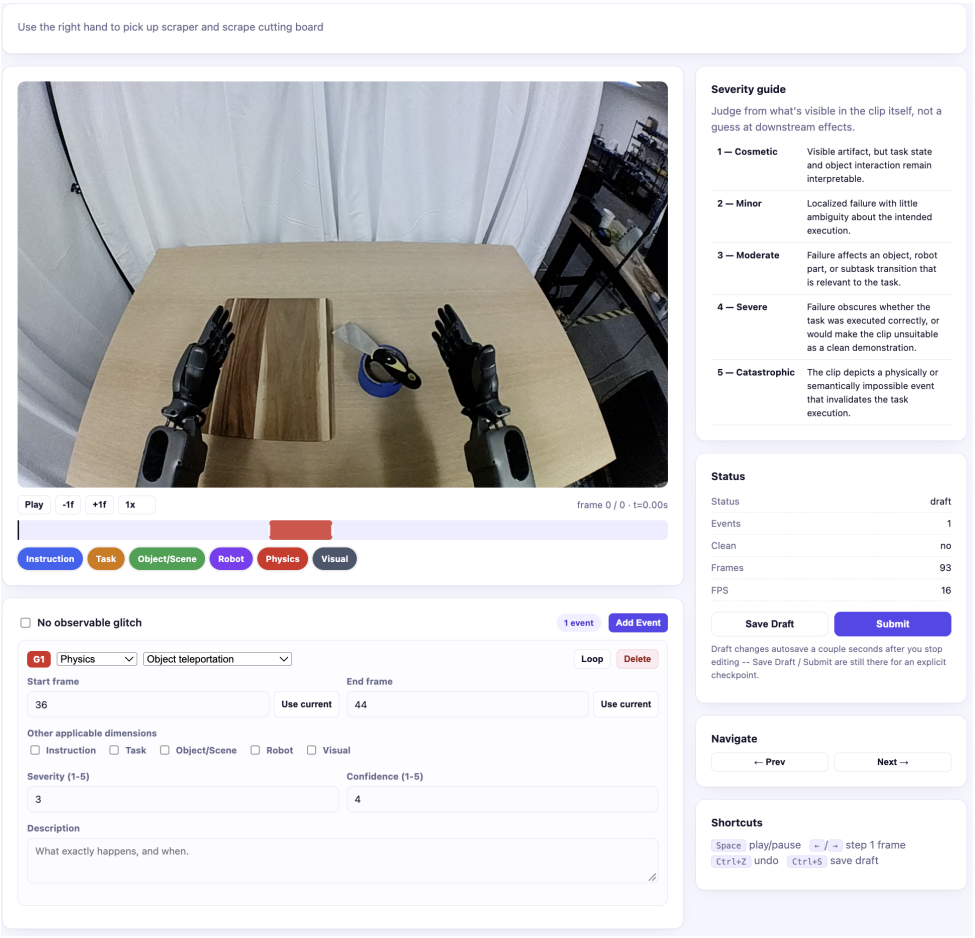


Figure 10: The annotation interface. Annotators see the generated video and the task instruction only.

RoboGazeBench took approximately 456 annotator-hours, of which 245 were production passes and 49 adjudication. The median clip cost 5 minutes per pass and the mean 8, the gap between the two being entirely the long tail of the horizon distribution (Figure 9). We report this because the cost of the benchmark is dominated by the property that distinguishes it: nearly half of the total labelling time went to the 14% of clips that run beyond forty seconds, because those clips carry close to half of the benchmark's total duration.

Table 11: Labelling effort. Passes are independent annotations of the same clips.

Stage	Scope	Passes	Hours
Training and calibration	7 annotators, 30-clip calibration	–	49
Production annotation	all 914 clips	2	245
Adjudication	all 914 clips	1	49
Gold subset	180 clips, senior annotators	3	72
Blinded human audit	180 clips, two methods	3	41
Total			**456**

C.9 Contamination

A benchmark assembled from existing video has to argue that the evaluated models have not already seen it. RoboGazeBench does not have that problem in its usual form, because every clip was generated for this work and none has been published: no evaluated backbone can have encountered a benchmark clip, and the failure events are properties of those clips, annotated here for the first time.

Two weaker channels remain. The conditioning instructions and initial observations are drawn from public datasets, so a backbone may recognise a scene layout or a task phrasing; this could help it describe what a clip is attempting but not whether the attempt went wrong, which is what the benchmark scores. And the schema and annotation guide are released with the benchmark, so models trained after release may have seen the vocabulary. We report exact API snapshot dates and checkpoint identifiers (Table 15) so that either channel can be checked against any model a later reader evaluates.

One block settles the question by construction. The Wan 2.2 and Veo clips were generated after every model in Table 3 was released and after the first access dates in Table 15, so no model could have been trained on them, their conditioning renderings, or their annotations. Those are also the two generators on which RoboGaze's margin is largest (Appendix E.2). Whatever residual contamination the public conditioning sources introduce, it does not account for the result. For the closed APIs we have no visibility into training data and do not claim any; this is a limitation of evaluating proprietary models rather than of the benchmark.

C.10 Agreement

Because every clip receives two independent passes, agreement is measured over the whole benchmark rather than a subset. Deciding *whether* a clip contains a failure is markedly more reliable than agreeing on the span of a physical-plausibility event ($\kappa = .74$ against a temporal IoU of .53), and agreement falls monotonically with horizon, reaching on the longest clips (> 80 s) about the level of the hardest dimensions. Across dimensions, the ordering closely follows the one the models produce (Figure 13).

Table 12: Inter-annotator agreement over all 914 clips, each independently annotated twice. κ and α count an event that the other annotator did not mark as a disagreement; temporal IoU is between the spans of temporally matched events of the two production passes. These are raw agreement statistics and are not the *human agreement* reference of Section 5, which scores each gold-subset senior annotator under the model matching protocol (Section 4).

	Detection κ	Dimension κ	Temporal IoU	Severity α	Desc. sim.
Overall	**.74**	**.66**	**.61**	**.58**	**.73**
≤ 15 s	.77	.69	.64	.61	.75
15–40 s	.71	.62	.56	.55	.70
40–80 s	.67	.60	.54	.51	.68
> 80 s	.63	.56	.50	.46	.64
Task Progress	.79	–	.64	.61	.76
Instruction Consistency	.81	–	.67	.63	.79
Object–Scene Cons.	.72	–	.60	.57	.72
Robot-Body Cons.	.70	–	.59	.56	.70
Physical Plausibility	.63	–	.53	.50	.65
Visual Quality	.58	–	.49	.47	.62

C.11 Agreement at Three Granularities

Table 12 measures agreement on whether a clip contains a failure and on the dimension of an event. Because the schema also has a type level, we report agreement there as well, on the 180-clip gold subset where three senior annotators labelled every clip independently. Events are first matched temporally between annotators. The κ and α columns are then computed over all annotated events, with an event that no other annotator matched counted as a disagreement; the two hierarchical columns are computed over matched pairs only. The same-dimension rate (84%) is therefore higher than κ_{dim} alone would imply, because it conditions on the event having been found by both annotators.

Reliability falls monotonically with granularity: .76 for detection, .68 for the dimension, and .58 for the exact type (Krippendorff's $\alpha = .60$ over the thirty categories). Read hierarchically, 84% of matched pairs agree on the dimension and 60% on the exact type, so most type-level disagreement is a choice between siblings rather than a different reading of the event. This ordering is the reason the main paper aggregates at the dimension level (Table 3) and uses the type level only inside the specialists and the released manifest: the dimension level is where annotators agree well enough for a benchmark score, and the type level is where a diagnostic report is informative enough to act on.

Three sibling pairs account for about half of all type-level disagreement: `object_distortion` against `object_appearance_drift` (31 disagreeing matched pairs), `hallucinated_robot_part` against `duplicated_arm_or_gripper` (24), and `object_teleportation` against `other_implausible_dynamics` (19). All three are the pairs whose exclusion rules were written during calibration (Appendix B.2); they are boundary choices within a dimension rather than disagreements about what happened in the clip.

Per dimension, exact-type agreement follows the same ordering as detection agreement (*Instruction Consistency* .66, *Task Progress* .61, *Object–Scene* .59, *Robot-Body* .56, *Physical Plausibility* .51, *Visual Quality* .49), and the two weakest are Visual Quality, which holds two of the three evidence-limiting types, and Physical Plausibility, which holds the residual type. What sampled RGB frames can settle bounds the schema in the same place it bounds the evaluator (Figure 13).

Table 13: Agreement at three granularities, 180-clip gold subset, three independent senior annotations per clip. The κ and α columns are computed over all annotated events, an unmatched event counting as a disagreement; the hierarchical columns are computed over temporally matched event pairs only.

Granularity	κ	α	Hierarchical (matched pairs)	
			same dimension	same type
Clip-level detection	.76	–	–	–
Event dimension (6)	.68	–	84%	–
Event type (30)	.58	.60	84%	60%
Exact-type κ by dimension				
Task Progress	.61			
Instruction Consistency	.66			
Object–Scene Cons.	.59			
Robot-Body Cons.	.56			
Physical Plausibility	.51			
Visual Quality	.49			

C.12 Blinded Human Audit

The three senior annotators who produced the gold labels, blind to method identity, judged the Gemma4-31B predicted and unrecovered events of RoboGaze and GLiDE+ (B4b) on the 180-clip gold subset using the annotation rubric, and sorted every residual error into one of five kinds (Figure 11). Verification redistributes errors rather than removing them evenly: it more than halves the unsupported claims that reach the report, and the single kind RoboGaze produces more of than the control is the true event its critic rejects. Unsupported physical claims remain the largest kind for both methods, consistent with the per-dimension gap in Figure 13. Figure 16 complements this outcome-level view with representative cases grouped by the pipeline stage that caused each error.

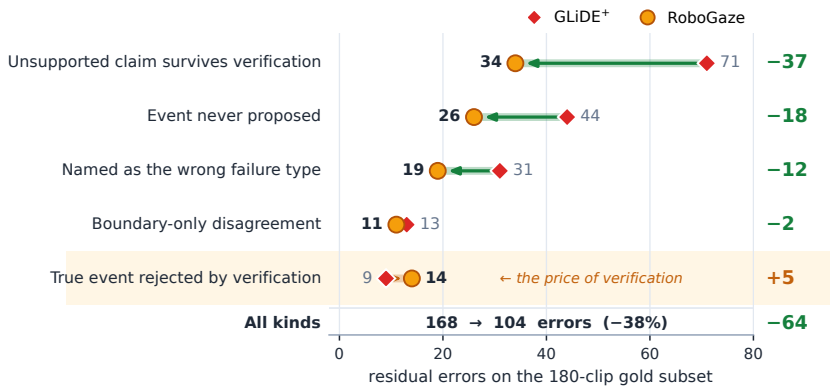


Figure 11: Residual errors by kind on the gold subset, GLiDE+ against RoboGaze. Counts rather than shares, because the totals differ.

Table 14: Human validation.

Check	Condition	Value	95% CI
	B4b control	61.4	[57.0, 65.7]
	RoboGaze	**71.6**	[67.5, 75.4]
	paired difference	+10.2	[6.3, 14.0]
Pairwise preference	RoboGaze over B4b	148/180 (82.2%)	[75.9, 87.2]
Judge–human correlation	text-only judge	$r = .79$	[.72, .85]
	video-grounded judge	$r = .83$	[.77, .88]

D Implementation and Metrics

D.1 Implementation

Table 15 records the API snapshots and open-weight checkpoints, Table 16 the per-backbone visual-input regime, and Table 17 the configuration of every method compared. Because the input policy differs between backbones, method effects are estimated within each backbone; the cross-backbone spread establishes breadth only.

Table 15: API access dates and checkpoint identifiers for the eight evaluated backbones. For closed models the date is when the API endpoint was first queried for our runs; providers may update an endpoint without notice. Open-source checkpoints were served using `vLLM 0.20.2` on 4×NVIDIA L40 (48 GB) with tensor parallelism = 4, batch size 1 per request, and BF16 weights unless otherwise noted.

Backbone	Family	Access date / checkpoint	Context length
Gemini 3.1 Pro	closed	API, accessed 2026-04-15	1M tokens
GPT-5.5	closed	API, accessed 2026-04-23	1M tokens
Gemini 3.1 Flash-Lite	closed	API, accessed 2026-04-15	1M tokens
Claude Sonnet 4.6	closed	API, accessed 2026-04-15	200k tokens (API default)
Gemma4-31B	open	`google/gemma-4-31b-it`	256k tokens
Qwen3.6-35B	open	`Qwen/Qwen3.6-35B-A3B`	262k tokens
LLaVA-OV-2-8B	open	`lmms-lab/llava-ov-2-8b-it`	32k tokens
InternVL3.5-38B	open	`OpenGVLab/InternVL3.5-38B`	32k tokens

Table 16: Per-backbone visual-input interface, held fixed across all methods. Closed APIs accept 16 frames per call; open-weight interfaces accept up to 32 frames at 4 FPS. The latter is an interface cap. The default RoboGaze observer uses 16 frames per non-overlapping 4 s window; the 32-frame window is evaluated only in the window-length sweep. Multi-call methods may therefore inspect more distinct frames per clip than a single pass. Sensitivity to the regime is reported in Appendix E.6.

Backbone	Mode	Effective rate	Per-call cap
Gemini 3.1 Pro	16 uniformly sampled	0.11–3.3 FPS	16 frames
Gemini 3.1 Flash-Lite	16 uniformly sampled	0.11–3.3 FPS	16 frames
GPT-5.5	16 uniformly sampled	0.11–3.3 FPS	16 frames
Claude Sonnet 4.6	16 uniformly sampled	0.11–3.3 FPS	16 frames
Gemma4-31B	full video @ 4 FPS	0.21–4 FPS	32 frames cap
Qwen3.6-35B	full video @ 4 FPS	0.21–4 FPS	32 frames cap
LLaVA-OV-2-8B	full video @ 4 FPS	0.21–4 FPS	32 frames cap
InternVL3.5-38B	full video @ 4 FPS	0.21–4 FPS	32 frames cap

Settings. Verifier confidence threshold 0.50 (swept in Appendix E.6), temporal merge IoU 0.6, router top-$k = 3$ dimensions per subtask segment, non-overlapping 4 s windows (about five on a clip of mean duration, so observation calls grow linearly with clip length; independent window calls are issued in parallel; the choice of length is discussed below), temperature 0 everywhere except B2, at most three retries on malformed output. A clip whose output cannot be parsed after three retries is scored as an empty report rather than dropped (on a failure-free clip this counts toward clean accuracy), which occurred for 0.4% of clip–method pairs; retries enter the compute accounting. Full prompt text is released with the code, each with a version string and content hash recorded in the run manifest.

Table 17: Configuration of every method compared. All share the backbone, the backbone-specific visual-input interface, the matching rule and the scoring configuration; none is fine-tuned.

	Method	Context supplied	Inference pattern	Verification
–	Vanilla	task instruction	1 call, $T{=}0$	none
–	CoT	task instruction	1 call, $T{=}0$, reasoning prefix	none
B1	Full-context schema + CoT	task, initial scene, six-dimension, thirty-type schema, severity rubric, normal alternatives	1 call, $T{=}0$	none
B2	B1 + self-consistency	as B1	9 samples, $T{=}0.7$; cluster at desc. sim. 0.7 / IoU 0.5; keep events with ≥ 5 supports	consensus only
B3	Generic propose-and-verify	as B1	propose, then verify each candidate; no subtasks or specialists	generic critic
B4a	GLiDE + task/scene	task, initial scene	GLiDE memory, scanner, debate, frame re-inspection, temporal zoom, crop	debate
B4b	B4a + schema/rubric	as B4a plus schema and rubric	as B4a	debate
B5	RoboGaze w/o critic	task, scene, subtasks	routed specialists	none
B6	**RoboGaze**	task, scene, subtasks	routed specialists over re-inspected windows	evidence critic + merge

Window length. The observer reads the clip in non-overlapping windows of 4 s. The length follows from the per-call frame budget: at the open-weight input rate of 4 FPS, a 4 s window fills one call's 16-frame budget, so every window is read with a frame every 0.25 s, and the median annotated event (1.4 s) spans several sampled frames inside a single window. Shorter windows add observation calls without adding frame density; longer windows must either spread the same 16 frames more thinly or use a larger call, up to the open-weight interface cap of 32 frames. Table 18 reports RoboGaze on Gemma4-31B with the window length varied and everything else fixed. Halving the window to 2 s lowers F_1 to 55.8, likely because more events straddle window boundaries; doubling it to 8 s at the same 16 frames halves the frame density and lowers F_1 to 54.8. Doubling both the window and the frames recovers 57.0, but requires 32-frame calls, beyond the 16-frame budget of the closed-API backbones, so 4 s is the longest window that every backbone can read at full density. Clean accuracy stays within .85–.87 throughout: the window length changes what the specialists see, not what the critic rejects.

Table 18: Window-length sweep, RoboGaze on Gemma4-31B, all 914 clips; every other setting as in the main experiments. The 4 s row is the default configuration of Table 3.

Window	Frames/window	$F_1 \uparrow$	$F_1{\times}$IoU$\uparrow$	Clean$\uparrow$
2 s	8	55.8	33.5	.85
4 s (default)	16	**57.3**	**36.1**	.86
8 s	16	54.8	32.5	**.87**
8 s	32	57.0	35.5	.86

D.2 Metrics

Judge. Description similarity S_{ij} is assigned by Gemini 3.1 Pro (the snapshot in Table 15) at temperature 0, from the two event descriptions only; malformed outputs are retried up to three times. GPT-5.5 (Table 15) serves as the alternate judge in the robustness checks. Because the primary judge is also an evaluated backbone, self-preference is tested directly by that swap (Appendix E.6). The judge's similarity ratings correlate with human ratings of the same pairs at $r = .79$ (Table 14).

Matching and scoring. Each predicted–reference pair is scored $C_{ij} = S_{ij} \cdot \mathrm{IoU}_t(\tau_i, \hat{\tau}_j) \cdot (1 + 0.25\,\mathbf{1}[\text{same dimension}])$, so the dimension bonus breaks ties between otherwise similar candidates but cannot create a match on its own. Pairwise temporal IoU is computed at the frame level. Pairs with $\mathrm{IoU}_t = 0$ have $C_{ij} = 0$ and are removed before matching; Hungarian assignment runs in maximization mode and zero-cost pairs are dropped, so a pair is matched only if $C_{ij} > 0$. The score is multiplicative rather than additive because a valid match must satisfy both semantic and temporal agreement. Following GLiDE (Zheng et al., 2026), with N predicted and M reference events in a clip and matched pairs $\mathcal{A}$, description precision and recall are $\sum_{(i,j)\in\mathcal{A}} S_{ij}/N$ and $\sum_{(i,j)\in\mathcal{A}} S_{ij}/M$, and description F_1 is their harmonic mean; mIoU is the mean IoU_t over $\mathcal{A}$; and $F_1{\times}$IoU replaces S_{ij} by $S_{ij}\,\mathrm{IoU}_t$ in the same precision–recall computation, so a matched event counts only as far as it both

describes and localizes the failure. All three are computed per clip and averaged over clips; $F_1 \times$IoU is therefore not the product of the reported F_1 and mIoU. An unmatched prediction contributes to the precision denominator and an unmatched reference to the recall denominator. An empty report on a clip with reference events scores zero recall; on a clean clip it scores one for clean accuracy and is excluded from the description metrics, whose denominators are undefined there. Severe-event recall is computed over all severity-4/5 references including unmatched ones, so it cannot be inflated by answering only on the events the evaluator found. Sev@1 is the percentage of matched events whose predicted severity lies within ± 1 of the reference severity on the 1–5 scale.

Confidence intervals are bootstrap intervals over source conditions rather than clips, since clips that share a source condition are not independent; 1,000 resamples are drawn and the macro mean over clips recomputed in each. Method comparisons are paired on the same clips and resampled jointly, so an interval on a difference is an interval on the paired difference. No threshold is selected separately for any backbone, generator or slice.

E Complete Results and Robustness

E.1 Baselines and Measured Cost

Table 19 reports every method compared on two backbones; Section 5.2 interprets the controls, and Figure 5 plots them against cost. Two details are recorded here. First, the shares of the CoT-to-RoboGaze distance closed by B1 and B4b are 49% and 68% on Gemma4-31B and 49% and 66% on Gemini 3.1 Pro, so the ordering of the controls does not depend on the backbone. Second, on the Gemma deployment inference cost scales with clip length for RoboGaze (7.1 calls and USD 0.138 per clip below ten seconds, 14.2 calls and USD 0.309 above thirty) and is flat for the single-pass baselines on clips longer than 8 s, where the 32-frame cap fixes their input.

Table 19: Every method compared, on all 914 clips. Sev-R counts all severity-4/5 references, misses included. Budget columns are the Gemma4-31B local deployment (4×L40, BF16, TP=4, USD 1.20 per GPU-hour), end-to-end wall-clock time per clip (s/clip × 4 GPUs × USD 1.20), including encoding, retries, tools and parsing; independent calls within a clip are issued concurrently, each at batch size 1; Gemini API cost per clip runs from USD 0.06 (vanilla) to 0.52 (B2) and 0.31 for RoboGaze.

		Gemma4-31B				Gemini 3.1 Pro			Budget (Gemma)		
	Method	F_1	$F_1 \times$IoU	Sev-R	Clean	F_1	$F_1 \times$IoU	Clean	Calls	s/clip	USD
–	Vanilla	16.4	5.1	.61	.04	24.1	9.2	.09	1.0	21	.028
–	CoT	21.8	8.2	.68	.11	30.2	12.9	.19	1.0	27	.036
B1	Full-context schema + CoT	39.3	20.4	.79	.58	47.9	26.1	.66	1.0	62	.083
B2	B1 + self-consistency (N=9)	43.4	22.9	.83	.64	51.6	28.8	.71	9.0	168	.224
B3	Generic propose-and-verify	42.1	22.1	.81	.66	50.2	27.9	.72	6.0	149	.199
B4a	GLiDE + task/scene	43.6	23.2	.84	.65	51.8	29.0	.71	9.0	141	.188
B4b	GLiDE$^+$ (B4a + schema)	45.9	24.8	.86	.69	54.1	30.1	.74	10.0	156	.208
B5	RoboGaze w/o critic	44.1	23.4	.94	.21	52.6	28.6	.29	7.6	118	.157
B6	**RoboGaze**	**57.3**	**36.1**	**.91**	**.86**	**66.4**	**42.4**	**.91**	9.4	152	.202

E.2 Results by Slice

Task family is not independent of duration: coordinated and deformable tasks are the longest in the benchmark, and their gain (+12.8) is close to what the duration bands alone predict, so we do not read it as a separate effect. Temporal mIoU by duration band is .52/.45/.36/.30 for B4b and .61/.57/.51/.47 for RoboGaze. From the shortest band to the longest, B4b therefore loses 15.4 F_1 and 42% of its temporal mIoU, against 9.0 and 23% for RoboGaze. With sixteen uniformly sampled frames, 68% of reference events in clips above forty seconds fall entirely between two sampled frames, and more inference over the same frames helps less and less: the advantage of B2 over B1 falls from +4.8 to +2.0 F_1 from short to long clips. The smallest gain (+9.8) falls on the one block that was quality-screened when first collected (Appendix C.5).

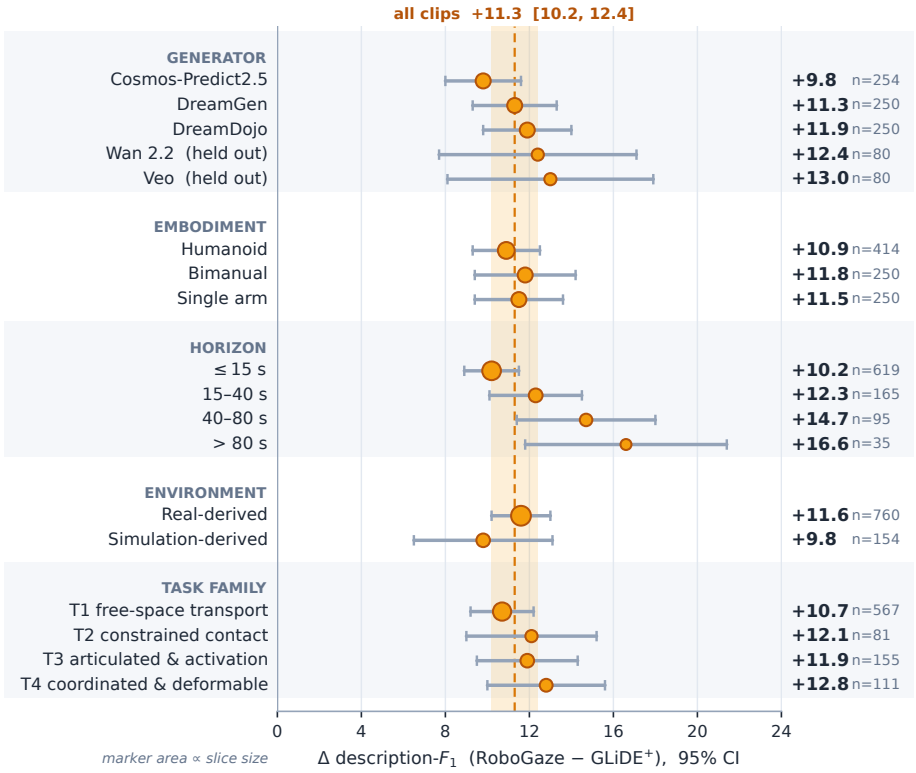


Figure 12: Paired description-F_1 gain over B4b on every benchmark axis, pooled over the eight backbones, with 95% source-grouped bootstrap intervals. All eighteen intervals exclude zero.

Table 20: Description F_1 by slice, pooled over the eight backbones. Δ is the paired RoboGaze−B4b difference with a source-grouped 95% bootstrap interval.

Axis	Slice	n	CoT	B4b	**RG**	Δ [95% CI]
Generator	Cosmos-Predict2.5	254	25.8	49.4	**59.2**	+9.8 [8.0, 11.6]
	DreamGen	250	24.1	46.7	**58.0**	+11.3 [9.3, 13.3]
	DreamDojo	250	22.8	44.8	**56.7**	+11.9 [9.8, 14.0]
	Wan 2.2	80	20.6	41.6	**54.0**	+12.4 [7.7, 17.1]
	Veo	80	19.9	40.4	**53.4**	+13.0 [8.1, 17.9]
Embodiment	Humanoid	414	24.3	47.3	**58.2**	+10.9 [9.3, 12.5]
	Bimanual	250	21.9	43.4	**55.2**	+11.8 [9.4, 14.2]
	Single arm	250	23.8	46.5	**58.0**	+11.5 [9.4, 13.6]
Duration	≤ 15 s	619	25.2	48.4	**58.6**	+10.2 [8.9, 11.5]
	15–40 s	165	22.4	44.6	**56.9**	+12.3 [10.1, 14.5]
	40–80 s	95	17.9	37.6	**52.3**	+14.7 [11.4, 18.0]
	> 80 s	35	15.1	33.0	**49.6**	+16.6 [11.8, 21.4]
Environment	Real-derived	760	23.2	45.4	**57.0**	+11.6 [10.2, 13.0]
	Simulation-derived	154	25.4	49.2	**59.0**	+9.8 [6.5, 13.1]
Task family	T1 free-space transport	567	24.6	47.3	**58.0**	+10.7 [9.2, 12.2]
	T2 constrained contact	81	21.0	42.1	**54.2**	+12.1 [9.0, 15.2]
	T3 articulated & activation	155	23.1	45.2	**57.1**	+11.9 [9.5, 14.3]
	T4 coordinated & deformable	111	20.8	41.8	**54.6**	+12.8 [10.0, 15.6]
All	**RoboGazeBench**	**914**	**23.5**	**46.0**	**57.3**	**+11.3** [10.2, 12.4]

E.3 Complete Ablations, Per-Dimension Scores, and Judge-Free Metrics

Scene memory matters more than task memory (-5.7 against -4.6 F_1), and the two are not additive: removing both costs 6.8 rather than 10.3. The critic is the only component whose removal raises

severe-event recall, and removing it together with the router costs barely more clean accuracy than removing it alone (.19 against .21), so the false reports it suppresses are not produced by routing.

Per dimension, the residual gap to human agreement is inversely related to how well annotators agree on that dimension (Figure 13, Spearman $\rho = -.94$): 0.2 F_1 on Instruction Consistency, where $\kappa = .81$, against 15.3 on Physical Plausibility, where $\kappa = .63$. Without a language judge, the methods separate most on the 210 failure-free clips: clean accuracy on Gemma4-31B is .11 for CoT, .69 for B4b and .86 for RoboGaze, while severe-event recall, also judge-free, is .68, .86 and .91 (Table 19).

Table 21: Complete component ablation, Gemma4-31B. F_1, $F_1 \times$IoU and Sev@1 (severity within ± 1 on matched events; Appendix D.2) are percentages; Sev-R and Clean are fractions.

Configuration	$F_1 \uparrow$	$F_1 \times$IoU$\uparrow$	Sev@1$\uparrow$	Sev-R$\uparrow$	Clean$\uparrow$
RoboGaze	**57.3**	**36.1**	**81.4**	**.91**	**.86**
w/o task memory	52.7	32.0	77.2	.88	.82
w/o scene memory	51.6	31.0	76.1	.87	.80
w/o task *and* scene memory	50.5	29.8	75.0	.86	.79
w/o subtask segmentation	53.8	33.1	78.0	.89	.83
w/o candidate router	52.2	31.9	77.0	.87	.82
w/o dimension specialists	50.9	30.4	75.8	.86	.81
w/o critic verifier	44.1	23.4	71.2	.94	.21
w/o router *and* critic	42.0	21.8	69.5	.93	.19

Table 22: Per-dimension description F_1, 180-clip gold subset, Gemini 3.1 Pro.

Dimension	CoT	B4b	**RG**	*Human*
Task Progress	34.1	63.8	**76.4**	*77.9*
Instruction Cons.	36.9	67.1	**77.0**	*77.2*
Object–Scene	30.2	58.4	**72.0**	*75.3*
Robot-Body	27.4	54.0	**67.3**	*73.6*
Physical Plaus.	20.8	42.1	**54.8**	*70.1*
Visual Quality	22.6	44.6	**56.3**	*68.4*

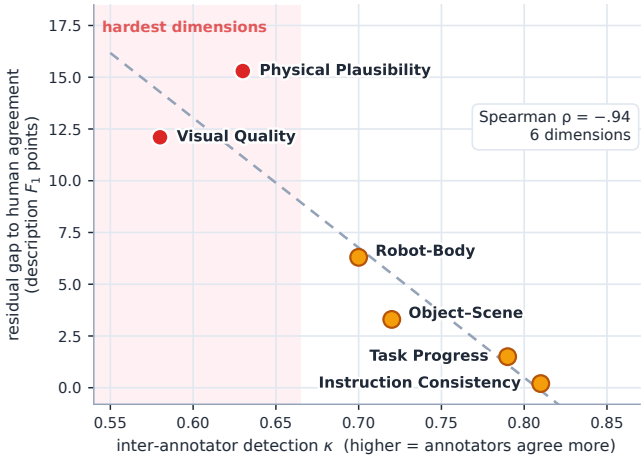


Figure 13: Per dimension, the residual gap between human agreement and RoboGaze (description F_1 points, Gemini 3.1 Pro, gold subset) against inter-annotator detection κ on that dimension. The dashed line is a least-squares fit; the shaded region marks the two dimensions with the lowest annotator agreement.

E.4 Gold Subset

The human reference in Table 3 is measured on the 180-clip gold subset rather than the full benchmark. Model scores on that subset are within 0.9 F_1 of their full-benchmark values (Table 23), so the two can be read against each other. The gap-closure figures in Section 5.1 are computed within this subset, where model and human scores share the same clips: on Gemini 3.1 Pro, RoboGaze recovers

$(67.3 - 31.1)/(74.6 - 31.1) = 83\%$ of the distance from CoT to human agreement in description F_1, GLiDE$^+$ 55%, and RoboGaze 99% of it in clean accuracy.

Table 23: Gold subset (n=180) against the full benchmark (n=914).

Backbone	Method	F_1 (gold)	F_1 (all)	Clean (gold)	Clean (all)
Gemini 3.1 Pro	CoT	31.1	30.2	.20	.19
	B4b	55.0	54.1	.75	.74
	RoboGaze	**67.3**	**66.4**	**.92**	**.91**
Gemma4-31B	CoT	22.4	21.8	.12	.11
	B4b	46.8	45.9	.70	.69
	RoboGaze	**58.2**	**57.3**	**.87**	**.86**
Human agreement		*74.6*	–	*.93*	–

E.5 Offline Candidate Screening

For this study we rendered 40 conditions six times each, under three generators and two seeds, outside the 914-clip benchmark; no screening candidate is also a benchmark clip. Results are in Table 24 (Section 5.3); all evaluators run on Gemma4-31B, and per-clip costs there are measured on these candidates and are not comparable with the benchmark-wide averages in Table 19. Independent annotators, blind to every evaluator output, labelled each candidate usable, unusable or uncertain as a *visual* demonstration; confirmed-usable prevalence is $118/240$ (0.49), so at 50% retention a perfect ranker is bounded above by 0.98 precision. We include two scalar evaluators that cannot emit event-level reports (VideoScore2, a learned judge, and the training-free RBench), scored through their native outputs, because the question here is ranking rather than diagnosis. The result concerns human-confirmed visual usability and not action executability, simulator success or policy performance.

Table 24: Offline screening of generated demonstrations. 40 conditions $\times$ 6 candidates rendered outside RoboGazeBench, ranked by each evaluator; human annotators, blind to every evaluator, label each candidate usable or not. P@50: usable precision in the retained half (random .49, perfect .98); Bad@50: rate of severity-4/5 failures in the retained half; Pick1: rate at which the top-ranked candidate is usable.

Evaluator	P@50↑	Bad@50↓	Pick1↑	USD/clip↓
Random selection	.49	.20	.49	.00
B1 Full-context schema + CoT	.66	.13	.63	.06
B4b GLiDE$^+$	.71	.11	.68	.11
VideoScore2 (He et al., 2025)	.65	.14	.61	.04
RBench (obj. + act. + phys.) (Deng et al., 2026)	.62	.15	.58	.05
RoboGaze	**.80**	**.05**	**.78**	.10

E.6 Robustness Checks

No check below changes the ordering of Table 3. The largest effect on the RoboGaze−B4b margin is 0.8 F_1, from halving the frame budget; every other check moves it by at most 0.5. Figure 14 gives the verifier operating curve; 0.50 was fixed on the development pool and the curve is reported so that a deployer can choose a different point deliberately.

Table 25: Robustness checks.

If this were true	Check	Result
The judge favours its own model family	re-score all cells with GPT-5.5	$\rho = .95$ over all 96 backbone×method×metric cells, 24 of which (clean accuracy) no judge affects, $\max \lvert\Delta\rvert = 1.9$, mean $\lvert\Delta\rvert = 0.8$
The result needs *some* LLM judge	BERTScore matching; judge-free metrics	backbone ranking $\rho = .93$, 61% of the gap retained; clean accuracy and Sev-R use no judge (Table 19)
The gain is explained by inference budget	B1 → B2, nine samples over identical evidence	$+4.1$ F_1 at $2.7\times$ cost, 23% of the residual gap; $+4.8 \rightarrow +2.0$ from short to long clips
The gain is the schema	B1 and B4b hold the full schema	they close 49% and 68% of the distance but reach only .58 and .69 clean accuracy
The operating point was chosen post hoc	threshold sweep, fixed on development data	ordering stable over .30–.70 (Figure 14)
The matching rule favours RoboGaze	$\lambda_{\text{dim}} \in \{0, .25, .5\}$; strict vs. loose	margin 11.0/11.4/11.4; loose 10.9; no ordering changes
Frame sampling drives the result	half, default and double the frame budget	RG 54.8/57.3/58.1, B4b 44.2/45.9/46.4; margin 10.6/11.4/11.7

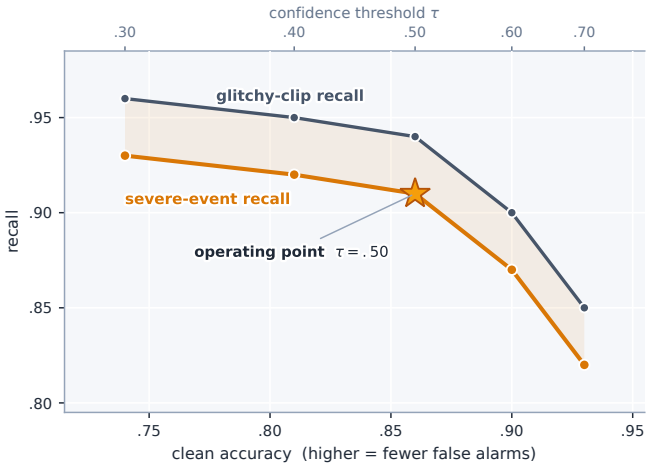


Figure 14: Verifier operating curve, Gemma4-31B: recall against clean accuracy as the confidence threshold varies over .30–.70. The starred point is the operating threshold, $\tau = .50$, fixed on development data. Description F_1 over the same range is 54.7, 56.5, 57.3, 56.6, 54.4.

F Qualitative Atlas, Release, and Ethics

Every frame shown in this paper is an unmodified benchmark frame. Crops are lossless and overlays are vector graphics; no frame has been upscaled, enhanced or generated by an image model, and captions state whether a panel is raw, cropped, annotated, plotted or schematic. The clip identifier, generator group, timestamp range and crop coordinates of every displayed case are recorded in the release.

The release, made public upon acceptance, will contain all 914 clips with their conditioning instructions and initial observations, the manifest, per-annotator labels and adjudication records, the annotation guide and interface, every prompt with its version and hash, the evaluator and baseline code including all configurations, every model output, the scoring code, and the analysis scripts that produce each table and figure. Source datasets are redistributed only under their original licences.

RoboGaze reports visible failures in generated video. An empty report means no sufficiently supported visible evidence of failure was found; it is not a certificate of physical validity and must not be used as a safety certification, as evidence of executability, or as a substitute for human oversight

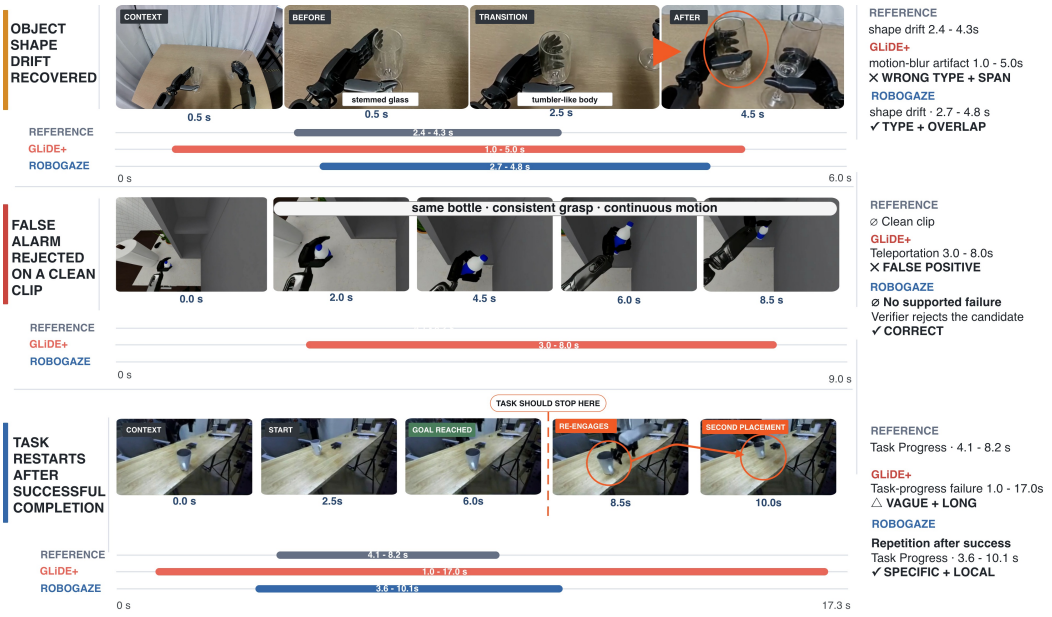


Figure 15: Where structured diagnosis pays off. Three representative single-view clips compare the human reference (gray), GLiDE$^+$ (red), and RoboGaze (blue). (a) GLiDE$^+$ misclassifies an object distortion (shape drift) as motion blur, whereas RoboGaze recovers the failure type and overlapping span. (b) The verifier rejects an unsupported teleportation report on a clean clip. (c) Both methods detect a post-completion task restart, but RoboGaze provides a more specific diagnosis and tighter localization. Frames are unmodified; complementary failure cases appear in Figure 16.

where an incorrect diagnosis could propagate into a physical system. The operating curve in Figure 14 is reported so that a deployer chooses a point rather than inheriting ours.



Figure 16: Where RoboGaze fails. One case per error mechanism, i.e. the pipeline stage that caused the error (a coding complementary to the outcome kinds of Figure 11), ordered by frequency, on five different generator groups. Each row shows unmodified frames spanning the event, the human reference and the predicted span, and the mechanism behind the error. Clip identifiers, timestamps and crop coordinates are in the release. Figure 15 shows the complementary successful cases.