RETRYSCRIBE: RETRY-AWARE CAPTIONING FOR ROBOTIC MANIPULATION VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

A successful robotic demonstration can contain missed grasps, corrective motions, and repeated attempts that a task-level caption such as "picks up the object" leaves unmentioned. Understanding these retries requires descriptions that preserve the target, timing, and local outcome of each attempt, together with its relationship to earlier attempts and the adjustments made between them. We introduce RetryScribe, a scalable framework for retry-aware captioning of robot manipulation videos. Because vision-language models alone can overlook brief or subtle retries and web videos lack such annotations, RetryScribe uses action logs to locate each attempt in time and video to identify the object it acts on and whether it succeeds, producing fine-grained caption supervision for 40,784 single- and dual-arm robotic demonstrations. We then distill this supervision into video-only captioners that require no action logs at inference. To evaluate retry understanding, we introduce RetryBench, a human-reviewed benchmark of 1,206 demonstrations and 10,795 questions that measures how much information about actions, attempt timing, and retry structure can be recovered from a caption. Experiments show that RetryScribe models substantially outperform their base models in capturing retries and cross-attempt relationships, on both in-domain robot-arm and out-of-domain handheld-gripper videos, while retaining general robot-video captioning ability.

1 INTRODUCTION

A successful robotic demonstration is not necessarily a sequence of successful attempts. In Figure 1, a robot misses a red block, reopens and repositions its gripper, then grasps and lifts the same block. The caption "the robot picks up the red block" captures the final outcome but omits the failed attempt and the adjustment that followed. These intermediate behaviors matter for robot learning: RaC (Hu et al., 2025) shows that training on recovery and correction trajectories can equip policies with retry behaviors and improve long-horizon task performance. We aim to make such behaviors explicit in captions of existing robot videos, rather than leaving them behind a task-level success label.

Understanding a retry requires more than detecting repeated actions. A gripper may close repeatedly while transferring different objects, continuing an operation, or attempting the same grasp again. Distinguishing these cases requires identifying the target and local outcome of each attempt and relating it to earlier attempts. Recent robot-video caption systems provide dense action descriptions and temporal grounding (Kou et al., 2025), and FineVLA (Hu et al., 2026) includes failure and recovery among its annotation dimensions. Retries, however, remain a minor part of these resources: only 1.26% of RoboFine-Bench questions concern retries (Table 1), and none asks how a later attempt relates to an earlier one or what was adjusted between them. Here we propose retry-aware captioning: captions that preserve each attempt's interval, target, and local outcome and link it to the attempt it revisits.

We introduce RetryScribe, which produces such captions at scale by combining two complementary sources. Action logs locate each attempt in time, but a gripper closure does not reveal which object was grasped or whether the grasp held; video shows the object and the outcome, but subtle failures make attempt boundaries hard to determine from pixels alone. RetryScribe therefore takes timing from the logs and the target and outcome from the video. We apply this pipeline to 40,784 single- and dual-arm robotic demonstrations from DROID (Khazatsky et al., 2024) and RoboCOIN (Wu

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.



Figure 1: A missed grasp followed by a successful retry. Exterior and wrist views show the gripper missing the red block, reopening and repositioning, then grasping and lifting the block. An attempt-level caption preserves the timing, failure, adjustment, and retry omitted by a task-level summary.

et al., 2025). Although most demonstrations are collected as successful executions, about one in six (17.1%) contains at least one retry annotated by RetryScribe. We then distill the resulting supervision into video-only captioners, so action logs are needed only to build the training data, not at inference.

To evaluate whether captions retain this information, we introduce RetryBench, with 10,795 human-reviewed questions over 1,206 robot-arm and handheld-gripper demonstrations. A frozen language-model reader answers each question from the generated caption alone, and its answers are scored against human references by exact matching or temporal overlap. Compared with its base model Qwen3.5-4B, RetryScribe-4B improves QA accuracy averaged over question families from 37.34 to 73.79 on robot-arm demonstrations and from 49.42 to 72.01 on handheld-gripper demonstrations, without training on the handheld domain. On the robot-arm subset, relationship and adjustment accuracy increase by 42.86 and 32.59 points, so the captions capture how attempts relate, not only whether a retry occurred. We leave downstream policy learning to future work.

Our contributions are threefold:

- The RetryScribe pipeline, which represents each demonstration as a sequence of attempts with their timing, targets, local outcomes, relationships, and adjustments, and combines action logs with video to produce caption supervision for 40,784 demonstrations.
- RetryBench, a human-reviewed benchmark of 10,795 questions over 1,206 demonstrations that evaluates which attempt and retry information, including cross-attempt relationships and adjustments, can be recovered from a caption.
- RetryScribe models, video-only captioners distilled from this supervision, which roughly double their base model's in-domain QA accuracy, transfer to unseen handheld-gripper videos, and gain most on cross-attempt relationships and adjustments.

2 Related Work

Language annotation and captioning of robot data. Large robot datasets such as Open X-Embodiment (Open X-Embodiment Collaboration et al., 2024) and DROID (Khazatsky et al., 2024) pair each demonstration with a task-level instruction. Several works enrich this language automatically: NILS (Blank et al., 2025) segments and labels long-horizon trajectories with foundation models, and DeMiAn (Kim et al., 2026) relabels over one million robot clips with dense descriptions of motion, scene, arm pose, and reasoning. RoboAnnotatorX (Kou et al., 2025) combines

Table 1: Training data statistics and evaluation benchmarks. (a) An episode can fall into several categories; Overall counts it once. Single:Dual gives the percentages of single- and dual-arm episodes. Segments, retries, duration, and words are per-video means; duration is in seconds and density in words per second. (b) Caption evaluation: whether the benchmark scores captions; other columns: whether it includes the corresponding question type. *: evaluation questions were unavailable. †: 1,732 caption-related questions (short action descriptions).

(a) Training data corpus

Retry category	# Videos	Single:Dual arms	Success %	Avg. time segments	Avg. retries	Video duration (s)	Avg. words	Word density
Failed grasp	2,181	76.1:23.9	89.94	15.04	1.588	25.50	181.0	7.10
Slip or drop	145	82.1:17.9	93.28	16.04	1.938	25.81	182.2	7.06
Misalignment	2,078	37.8:62.2	84.10	13.89	0.158	21.46	173.8	8.10
Ineffective action	1,334	29.1:70.9	89.69	16.74	0.771	18.06	197.8	10.95
Correction	3,075	63.2:36.8	79.88	16.57	1.590	27.31	194.5	7.12
None detected	30,410	57.6:42.4	87.81	11.45	0.000	18.87	150.3	7.97
Unclassified failure	2,256	71.9:28.1	85.19	14.02	1.262	23.12	174.5	7.54
Overall	**40,784**	**57.9:42.1**	**87.03**	**12.39**	**0.284**	**20.11**	**158.7**	**7.89**

(b) Evaluation benchmarks

Benchmark	# Videos / # QA	Caption evaluation	Temporal localization	Failure understanding	Retry counting	Cross-attempt relations	Adjustment understanding	Retry-related QA (%)
RoboX-VQA [18]	—* / 18K	Yes	Yes	—*	—*	—*	—*	—*
RoboInter-VQA [21]	7,246 / 77,345	Related†	No	Yes	No	No	No	0
RoboFine-Bench [14]	500 / 1,030	Yes	No	Yes	Yes	No	No	1.26
RetryBench (ours)	**1,206 / 10,795**	Yes	Yes	Yes	Yes	Yes	Yes	**48.05**

dense captioning, temporal localization, and question answering for long-horizon demonstrations, and FineVLA (Hu et al., 2026) provides step-level annotations, the RoboFine-VLM captioner, and RoboFine-Bench, whose dimensions explicitly include failure and recovery. Non-visual signals also support trajectory understanding: Robo2VLM (Chen et al., 2025) uses robot sensors to segment manipulation phases and construct grounded VQA, Suzuki et al. (2025) feed joint and end-effector states to a VLM for captioning and subtask segmentation, and ATLAS (Stanovcic et al., 2026) synchronizes video and robot time series for manual action-boundary and outcome annotation. In these works, retries are at most one annotation dimension among many rather than the object of study; only 1.26% of RoboFine-Bench questions concern retries (Table 1). RetryScribe instead makes retries the focus, linking each attempt to the earlier attempts it revisits, and RetryBench evaluates these links explicitly.

Failure understanding and recovery. Vision-language models are also used to judge robot execution. Du et al. (2023) train VLMs to detect whether a task succeeded, REFLECT (Liu et al., 2023) summarizes robot experience to explain and correct failures, AHA (Duan et al., 2025) detects and reasons over manipulation failures, and RoboFAC (Ye et al., 2025) builds a dataset and model for failure analysis and correction. These methods judge an episode or a step as a whole and do not track how a later attempt relates to an earlier one. Recovery behavior also matters for policy learning. RaC (Hu et al., 2025) shows that training on human interventions that pair recovery with correction improves long-horizon imitation learning. FailSafe (Lin et al., 2026) generates simulated failures paired with executable recovery actions, FAR (Hao et al., 2026) uses previous failures to adapt retry actions, and FLARE (Zhao et al., 2026) augments demonstrations for retry learning. ReTVL (Qin et al., 2026) uses annotated retry keypoints to learn values that reweight imperfect demonstrations for imitation. Generative world models face the same issue: Ctrl-World (Guo et al., 2026) notes that real policies often retry after failures, which its world model sometimes misses, and Dream2Fix (Li et al., 2026) constructs counterfactual failure videos paired with corrective actions. These studies motivate keeping the sequence of failure, adjustment, and renewed execution in robot data; RetryScribe instead recovers that sequence through captioning of existing robot manipulation videos.

Video captioning and caption evaluation. Dense video captioning (Krishna et al., 2017) localizes and describes multiple events in a video, and recent video-language models produce long, detailed

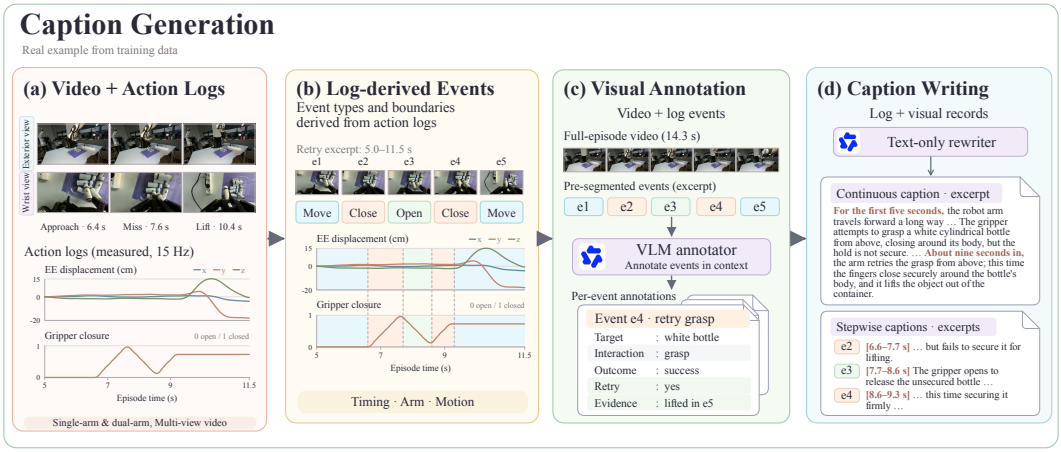


Figure 2: Caption-generation pipeline. (a) Multi-view video and measured action logs. (b) Event segmentation from robot motion and gripper signals. (c) A VLM (Qwen3.8-Flash-Next) annotates each event with full-video context. (d) A text-only rewriter produces continuous narration and step-wise captions with explicit time intervals.

descriptions (Wang et al., 2024; Chai et al., 2025). MotionAtlas (Liu et al., 2026) describes fine-grained motion within a specified spatiotemporal region. Because detailed captions are hard to score by text overlap, recent benchmarks evaluate them through question answering: VDC (Chai et al., 2025) decomposes captions into question-answer pairs, CAPability (Liu et al., 2025) measures correctness and thoroughness across visual dimensions, and CapQuiz (Wang et al., 2026) tests whether a caption carries enough information to answer human-verified multiple-choice questions about the video. RetryBench follows this caption-to-QA view in robot manipulation, targeting attempt timing, retry count, cross-attempt relationships, and adjustments.

3 RetryScribe

3.1 Attempt schema

We represent a demonstration as an ordered sequence of attempts $a_1, \ldots, a_n$, each

$$a_i = \big(o_i,\ g_i,\ [s_i, e_i],\ y_i\big), \tag{1}$$

with an operation o_i such as grasp, place, or push, a target object g_i, an interval $[s_i, e_i]$ in episode time, and a local outcome y_i. The local outcome records what is observed at the attempt's own boundaries, for example whether the object is held after the gripper closes. It is not a statement about the task, and an attempt without an observable outcome is not counted as a success. A retry edge $a_i \to a_j$ with $i < j$ links an attempt to an earlier attempt on the same target and carries the adjustment made between them, such as reopening the gripper, repositioning, or changing the grasp orientation.

Repeated motion alone is not a retry. A missed grasp followed by another grasp of the same block is a retry, and so is a deliberate release and regrasp that improves the grasp, even when the first grasp did not fully fail. Moving a different object starts a new operation. Repeated pushes on one object count as a retry only when they correct or re-execute an earlier attempt after its local outcome, not when they continue the same progress. These cases are decided from observable evidence, namely target identity, local outcome, and adjustment, rather than from inferred intent; Figure 5 includes such cases. The same vocabulary is used in the caption pipeline, the benchmark questions, and the evaluation.

3.2 Caption pipeline

Filling this schema needs two kinds of evidence (Figure 2). The action log records when the gripper closes and the arm moves, but not what was grasped or whether the grasp held. The video shows the

object and the outcome, but brief failures and similar successive motions make attempt boundaries hard to place consistently from pixels alone. RetryScribe therefore takes timing from the log and semantics from the video; Section 5.4 tests this split.

Source data. We use 40,784 single- and dual-arm episodes from DROID (Khazatsky et al., 2024) and RoboCOIN (Wu et al., 2025). Each episode has three synchronized camera views, end-effector poses, and a gripper signal. Success labels describe the task; we do not read them as evidence that execution contained no failed attempt.

Stage one: timed events. Thresholds on the per-frame change of the gripper signal and on end-effector speed split the episode into atomic events of four types, arm motion, closing, opening, and pause (Figure 2b; thresholds in Appendix B). Boundaries are measured signal transitions, not intended action boundaries: the approach to an object begins before the closure. This stage does not decide what was grasped, or whether anything was.

Stage two: visual facts. A VLM, Qwen3.8-Flash-Next (Qwen Team, 2026), receives the full multi-view video and the timed events in one call per episode (Figure 2c). For each interval it returns a verb, an external target, an optional local outcome, and observations before and after the interval, and it marks attempts that revisit an earlier one together with the adjustment between them. Fields the video does not settle stay null, robot parts are never targets, and object state is kept separate from action outcome, so a closed drawer is not a failed attempt. A text-only rewriter then writes these facts as a narration and as stepwise captions (Figure 2d).

Annotation quality. The stage-two fields are model predictions, and beyond admission checks and schema validation we apply no further filtering; 17.1% of episodes carry at least one retry annotation. In a human spot-check (Appendix E.1), 81.0% of caption segments in 100 random episodes have no observed error, and 94.7% of 301 retry-marked segments are genuine retries. We therefore treat the corpus as training supervision, not as gold labels.

3.3 Video-only student

We distill these captions into a captioner that needs no action log. The student is Qwen3.5-4B or Qwen3.5-9B (Qwen Team, 2026) with LoRA adapters (Hu et al., 2022) on the language layers; the vision encoder, merger, and output head stay frozen. Its input is the episode video and its target is a caption from Section 3.2; logs are used only to build the targets and are never given to the model, at training or at inference. Training details are in Appendix F.

4 RetryBench

4.1 Construction

RetryBench evaluates whether a caption preserves the attempt structure of Section 3.1. It contains 1,206 held-out demonstrations: 593 robot-arm episodes from DROID and 613 handheld-gripper episodes from FastUMI-100K (Liu et al., 2025). Figure 3 shows the construction. (a) Each video is annotated with the pipeline of Section 3.2; for FastUMI-100K, the recorded end-effector poses and gripper widths serve as the action log. (b) The resulting captions and temporal segments, together with an attempt graph that lists attempts and retry links, are checked against the video by Gemini 3.8 Flash, revised when needed, and rechecked. We start from 1,600 candidates, 800 per source, none of which appears in training; of the 1,530 that complete both checks, 1,206 are kept and 324 whose check remains uncertain are held out, and the remaining 70 did not complete or exceeded the input limits. (c) Fixed templates instantiate questions for each attempt or pair of attempts, so question generation is deterministic. (d) Gemini 3.8 Flash (Google DeepMind, 2026) answers every question from the original video, without the construction annotations; these answers are only candidates for human review (Section 4.4). The question set is frozen before any model is evaluated.

4.2 Question families

The questions fall into eight families that follow the schema (Figure 4c). Four concern retries: retry presence, retry count, retry relationship, which asks which earlier attempt a later one revisits, and adjustment, which asks what changed between them. Attempt timing asks for the intervals of

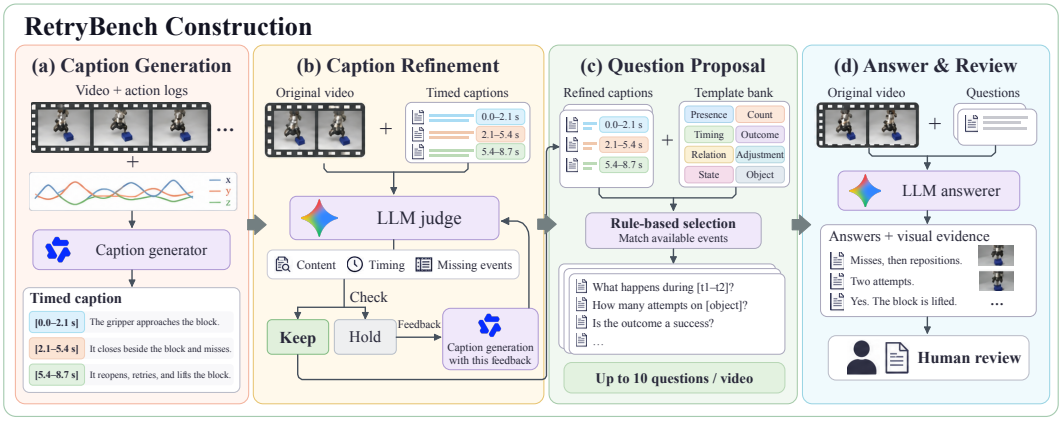


Figure 3: RetryBench construction. (a) Caption generation produces captions and temporal segments from held-out video and action logs, reusing Fig. 2. (b) Video-based checking, optional revision, and rechecking filter the annotations. (c) Fixed templates propose eight question families from the retained annotations. (d) Gemini 3.8 Flash generates candidate answers and evidence from the original video and questions. Human review then validates the released questions and answers.

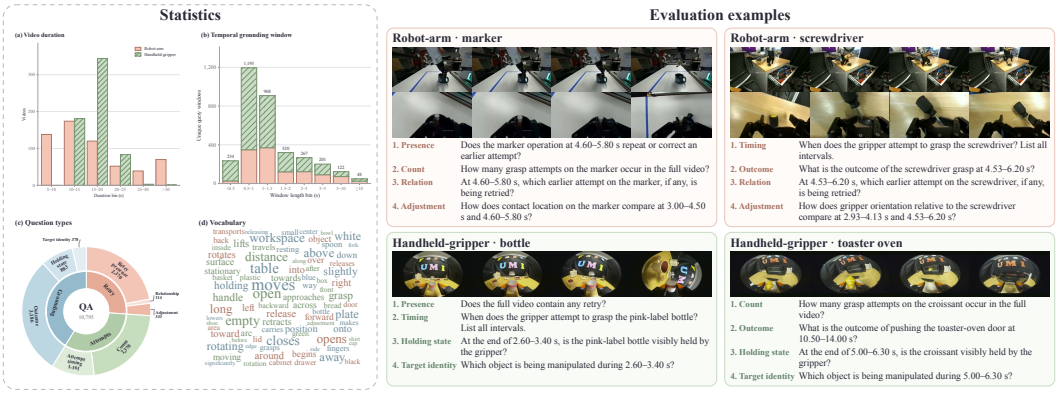


Figure 4: RetryBench statistics and evaluation examples. Left: (a) video duration, with durations above 30 s grouped; (b) 3,295 distinct time windows referenced by questions, deduplicated within each video; (c) question types, with presentation groups in the inner ring; (d) vocabulary of the construction captions. The benchmark contains 1,206 videos and 10,795 released questions. Right: four examples covering eight question types. Reference answers have been reviewed by humans.

specified attempts. Three general action families ask about the local object, local state, and outcome of an attempt. Answers are a choice, a count, or a list of intervals, and 48.05% of the questions concern retries (Table 1b). Twelve templates instantiate these families; Appendix C.1 lists them with their instantiation rules.

4.3 Caption-based QA and metrics

RetryBench scores a caption by what it lets a reader recover. The captioner sees only the video. A frozen reader sees only the caption and the question with its choices, never the video, the logs, or the reference answer. The evaluation therefore needs neither a reference caption nor a free-text caption judge. *Macro QA* is exact-match accuracy within each of the seven non-temporal families, averaged equally, separately for each domain. Temporal questions are scored by temporal IoU (*tIoU*) with order-preserving one-to-one matching between predicted and reference intervals, where missing and extra intervals lower the score through the denominator. Abstentions and invalid outputs receive zero credit. Appendix C gives the matching details.

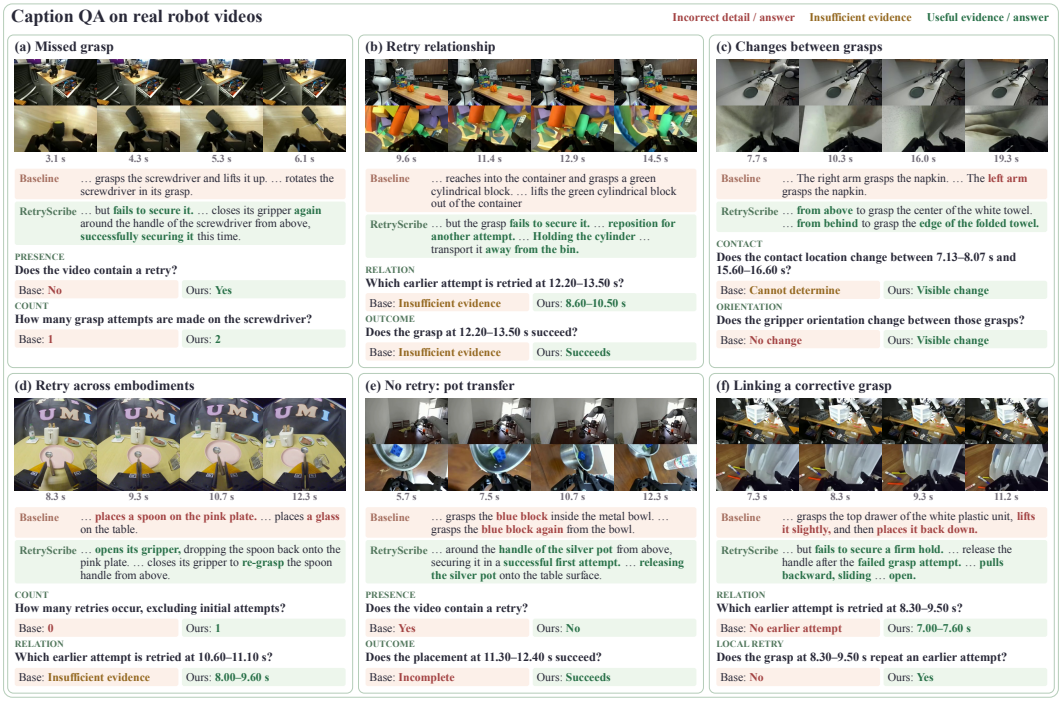


Figure 5: Qualitative Caption QA examples on RetryBench with Qwen3.5-4B as backbone. Six cases compare the baseline and RetryScribe captions and the answers produced by the same fixed QA reader. Timestamped frames show the relevant video evidence; ellipses mark omitted caption text. Green highlights useful retry-aware evidence.

4.4 Human review

Every question is reviewed by two people who answer it from the video, blind to the Gemini reference and to each other; a third reviewer adjudicates disagreements, and the final human answer replaces the reference when they differ. The two reviewers agree on 95.1% of questions, 82.0% of references are confirmed, and 1,948 are corrected. The protocol and per-family statistics are in Appendix E.

5 Experiments

5.1 Setup

Captioners. We compare RetryScribe-4B and RetryScribe-9B with their zero-shot bases, Qwen3.5-4B and Qwen3.5-9B; with the open-weight models Qwen3.5-27B (Qwen Team, 2026), Qwen3.6-27B (Qwen Team, 2026), GLM-5.3-Flash (Z.ai, 2026; GLM-5-Team et al., 2026), and MiMo-v2.6-Flash (Xiaomi MiMo Team, 2026); and with the closed models GPT-5.6-Sol and GPT-5.6-Luna (OpenAI, 2026), Muse-Spark-1.3 (Meta AI, 2026), and Qwen3.7-Flash (Qwen Team, 2026). All captioners receive the same videos and prompt, and none receives action logs. All captions are read by the same QA reader, DeepSeek-V4-Flash (DeepSeek-AI, 2026), with a fixed prompt and temperature 0.

Domains. The in-domain (ID) set contains the 593 DROID robot-arm videos. The out-of-domain (OOD) set contains the 613 FastUMI-100K videos, recorded with a handheld gripper (Chi et al., 2024) held directly by a person; no FastUMI-100K data is used for training or tuning. Camera placement, scenes, and tasks change along with the embodiment, so this split measures transfer under a broad domain shift rather than the effect of embodiment alone.

Metrics. Per Section 4.3, we report Macro QA on non-temporal questions and tIoU on attempt timing.

Table 2: Caption-based evaluation on RetryBench. Metrics use a 0–100 scale. Bold marks the best score in each column. Δ rows give each RetryScribe model minus its own base. *: size undisclosed.

		Robo manipulation (ID)		Handheld manipulation (OOD)	
Captioner	Total params	Macro QA ↑	tIoU ↑	Macro QA ↑	tIoU ↑
Closed-source					
GPT-5.6-Sol	–*	54.42	17.80	**78.58**	**25.22**
GPT-5.6-Luna	–*	38.54	7.19	56.14	12.14
Muse-Spark-1.3	–*	60.94	**24.40**	78.45	22.78
Qwen3.7-Flash	–*	52.87	3.09	69.93	4.08
Open-source					
Qwen3.5-27B	27B	53.89	14.73	65.85	14.76
Qwen3.6-27B	27B	52.36	16.06	67.52	15.43
GLM-5.3-Flash	320B	53.92	14.24	62.28	14.28
MiMo-v2.6-Flash	309B	40.07	5.79	63.92	11.80
Qwen3.5-4B	4B	37.34	6.02	49.42	9.26
RetryScribe-4B	4B	73.79	16.96	72.01	18.87
Δ vs. Qwen3.5-4B		↑36.45	↑10.94	↑22.59	↑9.61
Qwen3.5-9B	9B	42.53	8.57	56.71	8.00
RetryScribe-9B	9B	**75.34**	16.74	74.90	21.02
Δ vs. Qwen3.5-9B		↑32.81	↑8.17	↑18.19	↑13.02

Table 3: Results by retry question type. Base: Qwen3.5-4B; Ours: RetryScribe-4B; n: scored questions; timing uses tIoU, other rows QA accuracy (0–100). Full results, including 9B, are in Appendix C.2.

	Robo manipulation (ID)				Handheld manipulation (OOD)			
Question group	n	Base	Ours	Δ	n	Base	Ours	Δ
Retry presence	1,141	62.75	**87.91**	↑25.16	1,224	76.88	**90.52**	↑13.64
Retry count	1,145	60.70	**81.83**	↑21.13	1,224	75.74	**85.05**	↑9.31
Retry relationship	91	7.69	**50.55**	↑42.86	23	30.43	**65.22**	↑34.79
Adjustment	270	2.22	**34.81**	↑32.59	60	6.67	**36.67**	↑30.00
Attempt timing	546	6.02	**16.96**	↑10.94	607	9.26	**18.87**	↑9.61

5.2 Main results

Table 2 reports the results. Fine-tuning roughly doubles in-domain Macro QA, from 37.34 for Qwen3.5-4B to 73.79 for RetryScribe-4B, and raises tIoU from 6.02 to 16.96. RetryScribe-4B also exceeds both 27B open-weight models, which receive the same frames, in all four columns, and leads the stronger one by 19.90 in-domain Macro QA. Five of the eight zero-shot baselines, including the 320B GLM-5.3-Flash, fall between 52 and 55 in-domain Macro QA.

The gain carries over to handheld manipulation without FastUMI-100K training: Macro QA rises from 49.42 to 72.01 and tIoU from 9.26 to 18.87. The margin is smaller because zero-shot captioners are already stronger in this domain, and two closed models remain ahead: GPT-5.6-Sol leads on out-of-domain Macro QA and tIoU (78.58 and 25.22), and Muse-Spark-1.3 on in-domain tIoU (24.40). Muse-Spark-1.3 is also ahead of both RetryScribe models out of domain. RetryScribe-9B is higher than RetryScribe-4B in three of four columns and within 0.22 on in-domain tIoU. Its gains over its base are smaller, except on out-of-domain tIoU, because Qwen3.5-9B starts 5–7 Macro QA points higher; the larger base alone brings little temporal localization (8.57 and 8.00 tIoU).

5.3 Improvements by question type

Table 3 breaks the retry results down by question type. The largest gains concern how attempts relate rather than whether retries occur. In-domain, retry relationship rises from 7.69 to 50.55 and adjustment from 2.22 to 34.81, against gains of 25.16 on presence and 21.13 on count. The same ordering holds out of domain, where relationship and adjustment rise by 34.79 and 30.00 and presence and count, whose base scores already exceed 75, by 13.64 and 9.31. Timing improves less (+10.94 and +9.61 tIoU), and localization remains far from solved. Relationship has only 91 and 23 questions in the two domains and adjustment 60 out of domain, so these estimates are noisy. Figure 5 shows representative cases, and Table 8 in Appendix C.2 reports all eight families for both students.

Table 5: RoboFine-Bench caption evaluation (Hard setting, without task instructions). All scores are percentages. Arrows give each RetryScribe model's change over its base.

Model	Overall ↑		Consistency ↑		Coverage ↑		Anti-Halluc. ↑	
Qwen3.5-4B	63.56		65.50		43.26		81.93	
RetryScribe-4B	70.27	↑6.71	71.01	↑5.51	55.18	↑11.92	**84.60**	↑2.67
Qwen3.5-9B	65.41		66.37		45.29		84.56	
RetryScribe-9B	**70.49**	↑5.08	**71.06**	↑4.69	**55.93**	↑10.64	84.49	↓0.07

5.4 Supervision ablation

We hold the backbone, physical training episodes, training budget, and evaluation inputs fixed and vary only the evidence used to generate the caption targets: video only, action logs only, or both. Every row is evaluated on the full benchmark rather than on a subset matched to its target coverage. The experiment tests training utility, not whether every training annotation is correct; training-data spot-checks are in Appendix E.1.

Logs alone are not enough. The log-only student falls below the zero-shot base on both domains (19.98 vs. 37.34 in-domain Macro QA): its captions describe motion without naming objects or outcomes, and the reader abstains on 7,886 of 10,795 questions. Video-only and combined targets tie in-domain (54.63 vs. 54.03), but adding the log matters for transfer: combined targets are 8.32 points higher on handheld manipulation, where video-only targets barely improve on the base (51.95 vs. 49.42).

Table 4: Supervision ablation (Macro QA). Experiments done on training subset; rows differ only in the evidence used to annotate the caption targets.

Annotation source	Robo (ID)	Handheld (OOD)
- (baseline)	37.34	49.42
Video only	**54.63**	51.95
Log only	19.98	17.65
Video + log (ours)	54.03	**60.27**

5.5 External benchmark: RoboFine-Bench

Retry-focused training could come at the cost of general description. We test this on RoboFine-Bench (Hu et al., 2026) in its instruction-free Hard caption setting on all 500 videos, scored by the official atomic-fact judge. Every benchmark episode, including its 50 DROID videos, is excluded from our training data, and base and student receive identical frames, prompts, and decoding. The judge aligns the reference atomic facts with the caption: Consistency is the fraction of addressed facts the caption states correctly, Coverage the fraction of all reference facts it states correctly, Anti-Hallucination one minus the fraction of caption statements that match no reference fact, and Overall their mean. Table 5 shows that the Overall score rises by 6.71 for 4B and 5.08 for 9B, mainly through Coverage (+11.92 and +10.64), while Anti-Hallucination does not decrease: it rises by 2.67 for 4B and is unchanged for 9B (-0.07). The scores come from our evaluation harness and are not directly comparable to published numbers.

6 Conclusion and Limitations

Successful robot demonstrations often contain failed attempts that task-level captions leave out. RetryScribe represents a demonstration as a sequence of attempts linked by retry edges, fills this structure from action logs and video, and distills it into video-only captioners; RetryBench measures how much of the structure a caption preserves. Applied to existing robot demonstrations, the schema surfaces retries in about one in six of them, and the distilled captioners describe how attempts relate and what changed between them far more often than larger zero-shot models. We hope these artifacts make retry behavior in existing robot data searchable and measurable for future work.

Limitations. The schema relies on gripper events, so thin or occluded grasps and closure-free attempts such as pushes are harder to segment; the target object is the most common annotation error in retry-marked segments (Appendix E.1). Finally, we do not test whether retry-aware captions improve downstream policies.

AI USE STATEMENT

We used generative AI in four ways. First, models produce part of the data: Qwen3.8-Flash-Next annotates the training videos and a text-only language model rewrites these annotations into captions (Section 3.2); DeepSeek-V4-Flash is the fixed QA reader of RetryBench (Section 5.1); and Gemini 3.8 Flash checks the benchmark construction annotations and proposes candidate benchmark answers, every one of which was checked by human reviewers who replaced it when they disagreed (Section 4.4). Training annotations are model predictions, audited on a sample (Appendix E.1). Second, we used AI coding assistants to write and debug data-processing, training, and evaluation code; the authors tested this code and checked its outputs. Third, we used AI assistants to draft and edit parts of the text and to format tables and figures; the authors reviewed and revised all such content. Fourth, we used AI assistants to search for related work; the authors read every cited paper and checked each citation against its source. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

RetryScribe is built on the public robot manipulation datasets DROID (Khazatsky et al., 2024), RoboCOIN (Wu et al., 2025), and FastUMI-100K (Liu et al., 2025), used under their licenses; the captions, questions, and answers we derive from them are released under compatible terms. The videos show robot or handheld-gripper manipulation and may incidentally show people's hands or surroundings; we add no personal information. The only human effort in this work is reviewing benchmark answers and auditing training annotations from these videos, and no data about the reviewers is collected. Retry-aware captions are intended for analyzing and curating robot data, and we do not foresee harms beyond those of the underlying datasets.

REPRODUCIBILITY STATEMENT

Section 3.2 and Appendix B describe the annotation pipeline, and Appendix D gives its prompts and output schemas. Section 3.3 and Appendix F give the training recipe. Section 4 and Appendix C describe the benchmark construction, the reader configuration, and the scoring rules. All evaluations use a frozen question set and a fixed reader at temperature 0, and model differences are tested with paired bootstrap intervals. We will release the training captions, the RetryBench videos, questions, and human-reviewed references, the evaluation code, and the RetryScribe checkpoints, with pinned dataset and model revisions.

REFERENCES

[1] Nils Blank, Moritz Reuss, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Wenzel, Oier Mees, and Rudolf Lioutikov. Scaling robot policy learning via zero-shot labeling with foundation models. In *Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4158–4187, 2025. URL https://proceedings.mlr.press/v270/blank25a.html.

[2] Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jenq-Neng Hwang, Saining Xie, and Christopher D. Manning. AuroraCap: Efficient, performant video detailed captioning and a new benchmark. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.03051.

[3] Kaiyuan Chen, Shuangyu Xie, Zehan Ma, Pannag R. Sanketi, and Ken Goldberg. Robo2VLM: Improving visual question answering using large-scale robot manipulation data. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1f467c3e37abf9f86c78f44c6a27ee7c-Abstract-Datasets_and_Benchmarks_Track.html.

[4] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.045. URL https://arxiv.org/abs/2402.10329.

[5] DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, April 2026. URL https://arxiv.org/abs/2606.19348.

[6] Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-language models as success detectors. In *Conference on Lifelong Learning Agents (CoLLAs)*, volume 232 of *Proceedings of Machine Learning Research*, pp. 120–136, 2023. URL https://proceedings.mlr.press/v232/du23b.html.

[7] Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.00371.

[8] Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7 (1):1–26, 1979. doi: 10.1214/aos/1176344552.

[9] GLM-5-Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, et al. GLM-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026. URL https://arxiv.org/abs/2602.15763.

[10] Google DeepMind. Gemini 3.8 Flash model card, September 2026. URL https://deepmind.google/models/model-cards/gemini-3-8-flash/.

[11] Yanjiang Guo, Lucy Xiaoyang Shi, Jianyu Chen, and Chelsea Finn. Ctrl-World: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=v7hqqBLosx.

[12] Haoran Hao, Shahram Najam Syed, Jeffrey Ichnowski, and Jeff Schneider. FAR: Failure-aware retry for test-time recovery and continual policy improvement. *arXiv preprint arXiv:2607.01111*, 2026. URL https://arxiv.org/abs/2607.01111.

[13] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2106.09685.

[14] Xintong Hu, Xuhong Huang, Jinyu Zhang, Yutong Yao, Yuchong Sun, Qiuyue Wang, Mingsheng Li, Sicheng Xie, Yitao Liu, Junhao Chen, Yixuan Chen, Yingming Zheng, Shuai Bai, and Tao Yu. FineVLA: Fine-grained instruction alignment for steerable vision-language-action policies. *arXiv preprint arXiv:2605.27284*, 2026. URL https://arxiv.org/abs/2605.27284.

[15] Zheyuan Hu, Robyn Wu, Naveen Enock, Jasmine Li, Riya Kadakia, Zackory Erickson, and Aviral Kumar. RaC: Robot learning for long-horizon tasks by scaling recovery and correction. *arXiv preprint arXiv:2509.07953*, 2025. URL https://arxiv.org/abs/2509.07953.

[16] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.120. URL https://arxiv.org/abs/2403.12945.

[17] Bosung Kim, Ruiyi Wang, David Acuna, Jaehun Jung, Alexander Trevithick, Brandon Cui, Yejin Choi, and Prithviraj Ammanabrolu. How to instruct your robot: Dense language annotations power robot policy learning. *arXiv preprint arXiv:2605.17077*, 2026. URL https://arxiv.org/abs/2605.17077.

[18] Longxin Kou, Fei Ni, Yan Zheng, Peilong Han, Jinyi Liu, Haiqin Cui, Rui Liu, and Jianye Hao. RoboAnnotatorX: A comprehensive and universal annotation framework for accurate understanding of long-horizon robot demonstration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10353–10363, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Kou_RoboAnnotatorX_A_Comprehensive_and_Universal_Annotation_Framework_for_Accurate_Understanding_ICCV_2025_paper.html.

[19] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 706–715, 2017. URL https://arxiv.org/abs/1705.00754.

[20] Dayou Li, Jiuzhou Lei, Hao Wang, Lulin Liu, Yunhao Yang, Zihan Wang, Bangya Liu, Minghui Zheng, and Zhiwen Fan. Learning actionable manipulation recovery via counterfactual failure synthesis. *arXiv preprint arXiv:2603.13528*, 2026. URL https://arxiv.org/abs/2603.13528.

[21] Hao Li, Ziqin Wang, Zi-han Ding, Shuai Yang, Yilun Chen, Yang Tian, Xiaolin Hu, Tai Wang, Dahua Lin, Feng Zhao, Si Liu, and Jiangmiao Pang. RoboInter: A holistic intermediate representation suite towards robotic manipulation. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2602.09973.

[22] Zijun Lin, Jiafei Duan, Haoquan Fang, Dieter Fox, Ranjay Krishna, Cheston Tan, and Bihan Wen. FailSafe: Reasoning and recovery from failures in vision-language-action models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026. URL https://arxiv.org/abs/2510.01642.

[23] Kehui Liu, Zhongjie Jia, Yang Li, Zhaxizhuoma, Pengan Chen, Song Liu, Xin Liu, Pingrui Zhang, Haoming Song, Xinyi Ye, Nieqing Cao, Zhigang Wang, Jia Zeng, Dong Wang, Yan Ding, Bin Zhao, and Xuelong Li. FastUMI-100K: Advancing data-driven robotic manipulation with a large-scale UMI-style dataset. *arXiv preprint arXiv:2510.08022*, 2025. URL https://arxiv.org/abs/2510.08022.

[24] Weisong Liu, Haochen Wang, Kuan Gao, Yuhao Wang, Yikang Zhou, Zhongwei Ren, Jacky Mai, Anna Wang, Yanwei Li, Jason Li, and Zhaoxiang Zhang. MotionAtlas: Detailed region captioning for motion-centric videos. In *European Conference on Computer Vision (ECCV)*, 2026. URL https://arxiv.org/abs/2606.29531.

[25] Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing robot experiences for failure explanation and correction. In *Conference on Robot Learning (CoRL)*, 2023. URL https://arxiv.org/abs/2306.15724.

[26] Zhihang Liu, Chen-Wei Xie, Bin Wen, Feiwu Yu, Jixuan Chen, Pandeng Li, Boqiang Zhang, Nianzu Yang, Yinglu Li, Zuan Gao, Yun Zheng, and Hongtao Xie. CAPability: A comprehensive visual caption benchmark for evaluating both correctness and thoroughness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2502.14914.

[27] Meta AI. Introducing Muse Spark 1.3, September 2026. URL https://research.meta.ai/blog/introducing-muse-spark-1-3.

[28] Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, et al. Open X-embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

[29] OpenAI. GPT-5.6 system card, July 2026. URL https://deploymentsafety.openai.com/gpt-5-6.

[30] Xinyao Qin, Junjie Lu, Kaixin Wang, Chuheng Zhang, Sinjae Kang, Kimin Lee, Min Xu, Bin Liang, Jun Yang, and Li Zhao. Beyond monotonic progress: Retry-supervised value learning for robot imitation. *arXiv preprint arXiv:2606.24633*, 2026.

[31] Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

[32] Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, April 2026. URL https://qwen.ai/blog?id=qwen3.6-27b.

[33] Qwen Team. Qwen3.7-Flash, 2026. URL https://www.qwencloud.com/models/qwen3.7-flash.

[34] Qwen Team. Qwen3.8-Flash-Next: A new architecture, towards ultimate cost-efficiency, August 2026. URL https://qwen.ai/blog?id=qwen3.8-flash-next.

[35] Sergej Stanovcic, Daniel Sliwowski, and Dongheui Lee. ATLAS: An annotation tool for long-horizon robotic action segmentation. *arXiv preprint arXiv:2604.26637*, 2026. URL https://arxiv.org/abs/2604.26637.

[36] Kanata Suzuki, Shota Shimizu, and Tetsuya Ogata. Proprioception enhances vision language model in generating captions and subtask segmentations for robot task. *arXiv preprint arXiv:2512.20876*, 2025. URL https://arxiv.org/abs/2512.20876.

[37] Jiawei Wang, Liping Yuan, Yuchen Zhang, and Haomiao Sun. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024. URL https://arxiv.org/abs/2407.00634.

[38] Zizhen Wang, Bo Feng, Zhengfeng Lai, Shiyu Li, Yang Lu, Meng Cao, Ping Huang, and Xiaoming Simon Wang. Putting captions to the test: Evaluating video caption quality through multiple-choice question answering. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026. URL https://arxiv.org/abs/2609.09973.

[39] Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, et al. RoboCOIN: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025. URL https://arxiv.org/abs/2511.17441.

[40] Xiaomi MiMo Team. MiMo-V2.6-Flash, September 2026. URL https://mimo.mi.com/models/en-US/mimo-v2.6-flash.

[41] Zewei Ye, Weifeng Lu, Minghao Ye, Tao Lin, Shuo Yang, Junchi Yan, and Bo Zhao. RoboFAC: A comprehensive framework for robotic failure analysis and correction. *arXiv preprint arXiv:2505.12224*, 2025. URL https://arxiv.org/abs/2505.12224.

[42] Z.ai. GLM-5.3-Flash, August 2026. URL https://huggingface.co/zai-org/GLM-5.3-Flash.

[43] Ganlong Zhao, Zijia Tang, Xingping Chen, Zhanghui Kuang, Ye Tian, and Guanbin Li. FLARE: A failure-aware framework for autonomous correction and recovery in visual-language robotic manipulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22391–22401, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhao_FLARE_A_Failure-Aware_Framework_for_Autonomous_Correction_and_Recovery_in_CVPR_2026_paper.html.

A DATA STATISTICS

Table 6: Training-corpus statistics, measured over all 40,784 episodes of the training corpus. Caption length uses the narrative caption; words are counted as alphanumeric tokens. Final success is the DROID episode label and covers the DROID episodes only, since RoboCOIN publishes no outcome. Retry and failure statistics refer to automatic annotations, not a fully human-labeled corpus; a failing outcome is a missed, failed, slipped, or incomplete attempt. Per-category statistics are in Table 1(a).

Statistic	Value
Source corpora	DROID, RoboCOIN
Episodes, total (DROID / RoboCOIN)	40,784 (23,595 / 17,189)
Single- / dual-arm episodes (%)	57.9 / 42.1
Camera views per episode	3
Caption targets (4 output formats per episode)	163,136
Total video duration (hours, one timeline per episode)	227.8
Episode duration, mean (s)	20.11
Time segments per episode, mean	12.39
Words per caption, mean (DROID / RoboCOIN)	158.7 (128.6 / 200.0)
Word density, mean (words/s)	7.89
Final success, DROID episodes (%)	87.03
Episodes with at least one retry	6,980 (17.1%)
Episodes with at least one failing outcome	7,299 (17.9%)
Retries per episode, mean	0.284
Retry-marked arm records (with adjustment text)	11,600 (11,517)

B RULE SPECIFICATION

Signals. The action log is resampled to 10 Hz. The gripper signal is normalized so that 1 means closed: DROID uses its gripper position directly, RoboCOIN is rescaled by the span of each source repository, and FastUMI-100K uses one minus its gripper width.

Atomic events. A frame is *closing* if the normalized gripper signal rises by more than 0.005 from the previous frame and *opening* if it falls by more than 0.005. A frame is *arm motion* if the end-effector moves at 0.02 m/s or faster or rotates at 0.1 rad/s or faster in any Euler component. Runs shorter than 0.1 s are dropped, as are gripper runs whose total change is below 0.04 and motion runs whose net translation is below 0.02 m and whose rotation is below 10°.

Segments. Per-frame flags from all arms are combined into one key per frame, with a gripper event taking precedence over motion on the same arm, so a closure is never split by motion. Adjacent frames with the same key form a segment. A segment with no gripper event and no reportable motion that is shorter than 1.0 s is absorbed into its neighbor. For every closing segment, the measured approach direction is the reverse of the direction of the motion that leads into it; for example, a downward approach is recorded as "from above".

Admission. Training episodes must satisfy the following checks. Every RoboFine-Bench and RetryBench episode is excluded, and selection fails if an excluded episode appears. DROID episodes must last at least 6 s and contain at least one closing and one opening event. RoboCOIN episodes must be dual-arm and have their action log and all camera views present. RetryBench candidates must last at least 6 s and are held out when they have more than 50 segments.

Frame budget. Videos are sampled at 4 FPS. When a clip would exceed the encoder budget of 49,152 video tokens, its frame rate is lowered to fit, and clips that would fall below 2 FPS are skipped. Training samples longer than 65,536 tokens in total are dropped, which removes 151 samples from the corpus. No episode is split.

C EVALUATION DETAILS

Temporal IoU. For a timing question with predicted intervals $P = (p_1, \ldots, p_m)$ and reference intervals $R = (r_1, \ldots, r_n)$, both in chronological order, we find the order-preserving one-to-one matching that maximizes the sum of interval IoUs, computed by dynamic programming,

$$T_{i,j} = \max\big(T_{i-1,j},\ T_{i,j-1},\ T_{i-1,j-1} + \mathrm{IoU}(p_i, r_j)\big), \tag{2}$$

and score the question as $T_{m,n}/\max(m, n)$. Missing and extra intervals therefore both lower the score. If both lists are empty the score is 1, and an abstention or invalid answer scores 0. tIoU is the mean of this score over the temporal questions of a domain.

Reader. The reader (Appendix D.5) runs at temperature 0 with at most 10,000 output tokens and up to three attempts per batch of questions; timing questions are sent one per batch. A batch that fails all attempts scores zero and stays in the denominator.

Statistics. Intervals for model differences come from a paired bootstrap over physical episodes with 2,000 resamples, keeping all questions of an episode together, and use the 2.5th and 97.5th percentiles (Efron, 1979). For both RetryScribe models, all four 95% intervals of the gain over the base in Table 2 exclude zero. In the supervision ablation (Table 4), the in-domain interval between video-only and combined targets crosses zero, and video-only targets are not significantly above the base on handheld manipulation. On RoboFine-Bench (Table 5), the Anti-Hallucination interval of RetryScribe-9B crosses zero, and all other intervals exclude zero.

C.1 QUESTION TEMPLATES

Table 7 lists the twelve templates. Each fills its slots from the attempt graph of the construction caption: an operation OP, an object description OBJ, and attempt intervals SPAN, written in seconds with two decimals.

Instantiation. Every episode receives the two episode-level questions. If the graph has attempts, one *primary* attempt is chosen: the later attempt of the first retry edge; otherwise the later attempt of the first pair with the same object and operation; otherwise the first attempt. The primary attempt receives the object attempt count, timing, local outcome, and local retry presence questions. If earlier attempts exist, a retry relationship question offers as options the true retry parent, if any, the three most recent other earlier attempts, no parent, and cannot be determined. For the retry edge or same-object pair, the three adjustment questions compare the two attempts; when both contact regions are known and differ, the contact question lists the concrete regions as options instead of the generic change options. Remaining slots, up to ten questions per episode, are filled with the local outcome of the other attempts, then the holding state and, when the graph has more than one object, the target identity of the primary attempt. No template invents a retry pair that the graph does not contain.

Options. Every choice question includes an option for the case that the video does not settle it, and retry questions include the negative answer, so episodes without retries yield valid questions. Options are shuffled by a hash of the question identifier, which prevents answer-position regularities. Timing questions never state the reference interval and are sent to the reader one per request. Three timing questions whose object descriptions depend on attempt order (“first”, “second”) are omitted as ambiguous.

C.2 RESULTS BY QUESTION FAMILY

Table 8 expands Table 2 into all eight question families for both students and their bases.

D PROMPTS AND OUTPUT CONTRACTS

This appendix gives every prompt verbatim. Table 9 lists the model, inputs, and decoding settings of each call. All calls except the captioners request JSON under a strict schema; an answer that fails validation is sent back with the validator’s complaint, up to the listed number of attempts.

Table 7: RetryBench question templates. Answer: answer format, with choice options abbreviated; n: released questions.

Family	Template	Answer	n
Retry presence	Does the video contain any retry under the stated definition?	Yes / No / Cannot determine	1,206
	Does the operation on the OBJ during SPAN repeat or visibly correct an earlier attempt on that same object and local operation?	Yes / No / Cannot determine	1,164
Retry count	How many retries are visible in the full video? Count retries only, excluding the initial attempt for each object and local operation.	Number	1,206
	How many distinct attempts at the operation "OP" involving the OBJ are visible in the full video?	Number	1,164
Retry relationship	Which earlier operation, if any, is the retry parent of the operation on the OBJ during SPAN?	Earlier intervals / None / Cannot determine	114
Adjustment	Comparing the attempts on the OBJ during SPAN_1 and SPAN_2, how does the {contact location on the object, gripper orientation relative to the object, approach path relative to the object} compare?	Change / No change / Not established / Not comparable	333
Attempt timing	At what times does the gripper attempt the operation "OP" involving the OBJ? Return all visible attempt intervals, or an empty list if none occur.	List of intervals	1,161
Local object	Which object is being manipulated during SPAN?	Up to five scene objects / None / Cannot distinguish	378
Local state	At the end of SPAN, is the OBJ visibly held by the gripper?	Held / Not held / Not established	883
Outcome	What is the local outcome of the "OP" attempt involving the OBJ during SPAN?	Succeeds / Fails / Incomplete / Cannot determine	3,186
All			10,795

Table 8: Caption-based QA by question family on RetryBench. Base: zero-shot Qwen3.5-4B or 9B; Ours: RetryScribe; Δ: Ours minus Base. n: number of scored questions. Timing uses temporal IoU; other rows use QA accuracy; Macro QA averages the seven non-temporal families. Scores are 0–100.

		4B			9B		
Question family	n	Base	Ours	Δ	Base	Ours	Δ
Robo manipulation (ID)							
Retry presence	1,141	62.75	87.91	↑25.16	66.43	88.17	↑21.74
Retry count	1,145	60.70	81.83	↑21.13	63.49	81.66	↑18.17
Retry relationship	91	7.69	50.55	↑42.86	6.59	57.14	↑50.55
Adjustment	270	2.22	34.81	↑32.59	4.81	36.30	↑31.49
Local object	233	47.21	90.13	↑42.92	54.08	90.56	↑36.48
Local state	444	42.79	86.04	↑43.25	55.86	87.39	↑31.53
Outcome	1,078	38.03	85.25	↑47.22	46.47	86.18	↑39.71
Macro QA		37.34	73.79	↑36.45	42.53	75.34	↑32.81
Attempt timing (tIoU)	546	6.02	16.96	↑10.94	8.57	16.74	↑8.17
Handheld manipulation (OOD)							
Retry presence	1,224	76.88	90.52	↑13.64	82.35	92.57	↑10.22
Retry count	1,224	75.74	85.05	↑9.31	78.92	89.13	↑10.21
Retry relationship	23	30.43	65.22	↑34.79	34.78	56.52	↑21.74
Adjustment	60	6.67	36.67	↑30.00	11.67	28.33	↑16.66
Local object	145	40.00	67.59	↑27.59	52.41	85.52	↑33.11
Local state	439	56.95	74.94	↑17.99	66.51	85.19	↑18.68
Outcome	2,095	59.28	84.11	↑24.83	70.31	87.06	↑16.75
Macro QA		49.42	72.01	↑22.59	56.71	74.90	↑18.19
Attempt timing (tIoU)	607	9.26	18.87	↑9.61	8.00	21.02	↑13.02

D.1 VISUAL ANNOTATION

One call per episode. The prompt is followed by a JSON input with the episode duration, the view layout, the per-arm action log for every stage-one segment, and an output template whose `robot_action` fields are prefilled from the log; the multi-view video comes last. The output holds, for every segment and arm, one interaction object with the fields named in the prompt; enumerated fields use the vocabularies listed there. The validator rejects, among others, a changed

Table 9: Model calls. Qwen3.8-Flash-Next and DeepSeek-V4-Flash (0731 release) run locally; Gemini 3.8 Flash is accessed through an API. Sampling for Qwen3.8-Flash-Next annotation calls is temperature 0.7, top-p 0.8, top-k 20, presence penalty 1.5, without thinking.

Call	Model	Input	Decoding	Attempts
Training captions (Section 3.2)				
Visual annotation (D.1)	Qwen3.8-Flash-Next	video + action log	sampled	2
Rewriter (D.2)	Qwen3.8-Flash-Next	text only	sampled	2
Captioning (Sections 3.3 and 5)				
Captioner (D.3)	all captioners	video (4 FPS)	greedy	1
RetryBench construction (Section 4)				
Attempt graph (D.4)	Qwen3.8-Flash-Next	video + caption	sampled	3
Check / recheck (D.4)	Gemini 3.8 Flash	video + draft	temperature 0	3
Revision (D.4)	Qwen3.8-Flash-Next	video + draft + critique	sampled	3
Candidate answers (D.4)	Gemini 3.8 Flash	video + questions	temperature 0	3
Evaluation (Section 4)				
QA reader (D.5)	DeepSeek-V4-Flash	caption + questions	temperature 0	3

`robot_action`, an outcome without a verb, `missed` or `slipped` without `grasp`, and a `retry_adjustment` on an attempt not marked as a retry.

```
Watch the synchronized video and read the measured action log. Return JSON only,
one segment for every supplied segment, in the original order. Within each
segment, describe each ARM first, then exactly ONE interaction object for that
arm. Use arms.single for a single-arm robot, or arms.left and arms.right for a
dual-arm robot. The arm identity is determined by its location in the JSON.

Copy robot_action from the supplied output_template. The action log is evidence
about robot motion, rotation, direction and finger opening/closing, not evidence
of object contact, holding, intent or success. Measured pause can include
stationary holding/support. Moving arms need not manipulate an object.
Read each arm's own measurements; never attribute them to another arm or object.

Fill every interaction object, even when no interaction is visible. interaction
is a single object, never an array. Do not output facts, fact_indices, actor,
arm_review, or additional interaction lists. Choose one principal target and
one principal interaction verb for this arm in this segment. Explain short
within-segment sequences, coordination, and effects on other objects in
observation rather than adding more interactions. Keep the measured segmentation.
Never move a later grasp or success into an earlier approach segment.

Read view_layout_left_to_right. It identifies the overview and whose wrist
camera each panel belongs to. Left/right means the robot's own arms, not screen
position. Use the overview to relate the arms and objects, and the relevant
wrist camera to inspect contact. Wrist-camera motion does not establish object
motion. Black padding is not scene content. The task instruction is context,
not proof of what occurred.

Each arm's interaction contains the following fields.
- target: the principal external object or part this arm directly manipulates
  or approaches. Not the robot, gripper or fingers. Use a consistent descriptive
  name, not an invented object ID. Use null when there is no target or its
  identity cannot be determined. When holding a tool, name the tool; the workpiece
  affected through the tool can be described in observation. Two arms holding
  different parts of the same tool should each identify that same tool. Do not
  assume a familiar tool/workpiece division of labour from the task title.
- contact_status: no_touch, touch or unclear, for this gripper and this target
  over the WHOLE segment. no_touch requires visible separation throughout;
  touch means at least one physical contact, including brief contact, ongoing
  holding, pressing or stationary support; unclear means the views cannot
  establish whether contact occurred. A visible contact is touch even if other
  frames are occluded. If target is null, no_touch explicitly means no external
  object was contacted; touch means visible contact with an unidentified object.
  No movement, identical before/after descriptions, not_held endpoints, and lack
  of a verb do not prove no_touch. Approach may be no_touch. A missed/failed
  grasp can be no_touch or touch depending on whether contact actually occurred.
  Holding a tool establishes contact with the tool, not direct gripper contact
  with its workpiece. This value is never prefilled from the action log.
- verb: the principal supported interaction with the target, using the supplied
  schema vocabulary, or null if no interaction can be identified. A target can
  be known while its interaction remains unclear. A visible interaction can have
  an unidentified target, so target null does not force verb null. Explain what
  is visible without inventing a name. Empty-arm travel alone is robot_action
```

```
  move, with verb null; approach needs visible movement towards something.
  approach is the arm getting closer to the object; move/carry requires visible
  object movement relative to the scene. An approach can leave the object still.
- holding_before and holding_after: what this gripper holds at the segment's
  first and last frame. Each contains status (held, not_held or unclear), target
  and evidence. held requires a target name and visible retention evidence. Use consistent
  names for the interaction target and held object when they are the same. not_held requires
     null target. Occlusion can be
  unclear. Holding is not inferred from a prior generated label or task title.
- attributes: visible color, material and shape_or_size of this target, each
  independently null when unknown. Do not invent details to fill the schema.
- outcome: success, unclear, missed, failed, slipped, incomplete or null.
  success means the intended interaction result is visibly achieved. For grasp,
  success requires visible retention, not mere closure. unclear means an attempt
  is visible but its result cannot be established. null means no applicable or
  specified outcome; it NEVER implies success. missed/slipped describe grasp
  attempts. incomplete means visibly unfinished, not merely uncertain or cut by
  a segment boundary. Holding endpoints describe only the first and last frames;
  a grasp and release can both happen within one segment. Explain that sequence
  in observation rather than forcing the endpoints to match a verb.
- retry: first_attempt, retry, unclear or null. retry means an earlier attempt
  in this episode was on the same target with the same goal and did not complete.
  first_attempt means this is the first attempt on that target. unclear means an
  earlier attempt may have happened but cannot be established. null means this
  segment carries no attempt at all, which is most of them. A second closure is
  not by itself a retry. A closure split across neighbouring segments is one
  attempt, not two, so check whether the gripper actually opened in between. The
  action log gives you every segment of the episode at once, so read back through
  it rather than judging from this segment alone.
- retry_adjustment: what is different about this attempt compared with the last
  one, in one line, and null unless retry is retry. Null also when this is a
  retry and you cannot see what changed; that is not a claim the two attempts
  were the same. Write the visible action only, never the strategy and never the
  reason. Say it about this episode and these two attempts, not in general terms.
  Do not put the result of the retry here, that is outcome.
- approach_direction: where this arm came from as it met or acted on the target,
  in the robot's own frame, such as from above, from below, from the left, from
  the right, from behind or from the front. Say it more precisely if the video
  shows more. It applies to grasping, pressing, pushing and touching, not only
  to segments where the fingers close. The action log carries a measured
  approach_direction of its own, computed only for closing segments; treat it as
  evidence, not as the answer, and do not copy it when the video disagrees.
  Left and right are the robot's own, never the left and right of a moving wrist
  image. A direction is not by itself evidence of contact. Null when this segment
  has no approach or the direction cannot be seen.
- contact_region: where this gripper visibly contacts the target, such as a
  handle, rim, blade, edge or switch surface. It applies to any visible contact,
  including pressing, supporting and ongoing holding without a closing event.
  Null if there is no contact or the contact location cannot be seen. A visible
  region can be named even if the object itself cannot be identified. Never
  infer contact location or contact status from a measured gripper event.
- path_shape: straight, arc, circular, back-and-forth or null. This describes
  the target's travel relative to the scene, not arm-only approach or camera
  motion. Null when the target stays still or its path is not visible.
- before and after: the target's relation, state and workspace_location at this
  segment's endpoints, each independently null when unknown. Use stable scene
  landmarks, not positions in a moving wrist view. Do not assign arm coordinates
  to the object or copy a later destination into this segment.
- side_effect: an unintended effect on another object, containing type,
  affected_object and effect, or null. This preserves the earlier side-effect
  information. Intended effects on the workpiece belong in observation.
- observation: briefly explain what this arm and target visibly do, contact
  evidence or uncertainty, and any local sequence or coordinated effect that
  does not fit the principal verb. For no target/no interaction, explain the
  visible empty gripper, separation, or uncertainty. Keep all text fields
  concise, single-line and at most 160 characters each.

Closing fingers is not automatically a grasp or even a grasp attempt. They may
close empty, adjust posture, prepare to push with closed fingers, press a switch,
touch/support something, fail to grasp, or successfully secure an object. Decide
from the video. Pressing a switch does not imply holding the surrounding device.
No lifting does not alone prove a failed grasp; an object can be secured on its
support. Opening empty fingers is not release. Release requires visible prior
holding of the same target, which may occur within this segment rather than
only at its first frame. Breaking pressing contact is withdrawal, not release
of a held object. Do not replace an unsupported grasp with an equally
unsupported push or press. No_touch cannot coexist with held endpoints or a
claimed contact region for the same target.

For dual-arm videos, AFTER describing both arms, fill same_object_manipulation.
Use true when they visibly manipulate/support the same physical object during
```

```
this interval, false when they do not, and null when identity or participation
cannot be established. Matching text labels alone are not evidence. Shared
targets can have different arm roles. Do not duplicate one arm's action onto
the other, and do not omit visible support just because an arm is stationary.

All template nulls are unfilled placeholders. Required non-null fields must be
filled with supported values, not guesses. Times are absolute episode seconds,
start_s inclusive and end_s exclusive. Preserve every segment and every arm.
```

D.2 REWRITER

The rewriter receives the action log and the visual record as JSON and no video. It returns `steps` (one entry per segment with the segment's start and end times), `narrative_timed`, and `narrative`. Step times must equal the stage-one times. The steps give the untimed and timed JSON targets, and the two narratives give the prose targets, so each episode yields four caption targets.

```
You are given everything that is known about one robot episode, already measured
and already annotated. Write the account of what happened, in natural language.
Return JSON only.

Two records are handed to you.

The action log is measured from the robot. Its times, its directions and the
sizes of its movements are facts. They are authoritative and nothing overrides
them. Every segment carries a row for each arm the episode has, with that arm's
own direction, distance, rotation, gripper event and approach direction. A
segment also names one measured actor, which is a convenience for finding the
busy arm rather than the whole picture. Read every arm's own row. A one-armed
episode has a single row, called arm, and its caption says the arm.

The visual record is what an annotator saw, one interaction for each arm in each
segment. It holds the target and its colour, material and size, whether that
gripper was touching the target, what the arm held before and after, where on the
object the fingers met it, which way the arm came in, the object's relation, state
and place in the workspace before and after, the outcome, whether the attempt was
a first attempt, a retry of an earlier one or unclear, a line saying what was
different about a retry, anything else that was disturbed, and an observation
written in plain words. The observation is raw
material for you rather than a field to copy.

There is no video here and you are not being asked to look at one. Write only from
those two records. **You cannot see the episode, so anything not given to you here
is something you would be making up.** If an object's colour is not recorded, it
does not have one in your account.

If a segment has no measured direction, do not name one. If no rotation was
measured, there was no measurable rotation. A missing detail is not a gap for you
to fill.

What the account has to carry.

Both arms. On a two-arm episode say what each arm was doing, with that arm's own
movements, including an arm that was holding something steady or waiting. Do not
fold the two into one subject and do not describe only the arm the log calls the
actor.

The motion. Where the log measured a direction for an arm, that arm's sentence
has to carry the log's own direction word. The six direction words are forward,
backward, left, right, up and down. A verb that implies a direction does not
replace one, so "lowers it down" carries the direction and "descends" does not.
Write "lifts it up", not "lifts it".

**This holds whatever else the segment contains, and the case it gets lost in is
a measured direction with a gripper event beside it.** The closing or the release
is the obvious thing to write and the travel then goes unsaid. Both go in. "The
gripper opens and releases the bottle" has lost the arm moving up, which was
measured. So has "the gripper closes on the handle from above", which has lost
the arm still moving down while the fingers close, and an approach is not a
travel. Where a direction was measured, say it, even on a segment that is mostly
about the fingers.

How far it went and how much it turned are yours to put in words. **Never give a
figure for either one, anywhere in your answer.** No centimetres, no degrees, no
other unit and no number. The log tells you the size, so you know whether a reach
was a small adjustment or a long way across the workspace, and that is the part
to say. A turn that was measured is a turn worth mentioning, without saying how
```

```
far it turned.

The objects. What was being reached for, how the gripper met it and where, what
was already in each hand, what the object was doing beforehand and what state it
was left in, what else moved. The observation often carries the part of the process
the other fields cannot hold, and that is what it is there for.

Retries. The annotator marks each attempt a first attempt, a retry or unclear, and
on a retry it leaves a line saying what was different about that attempt. It saw
the episode and you did not, so its mark is what decides whether there was a retry.
Where it marks one, tell it, and use that line to say what was done differently and
how the attempt ended. The action log helps you describe it. Where the mark says
first attempt, no retry wording belongs, and where it says unclear, leave it open.
If nothing is marked, do not mention retries at all. A blank outcome still means
the annotator did not judge that attempt, not that it succeeded.

What to produce.

The same episode in three pieces, one stepwise and two continuous. All three carry
the same course of events and the same substance, and none of them may contradict
another.

steps. One entry for each segment, in order, carrying that segment's id and its
start and end seconds exactly as the log gives them. Its text is the account of
that segment and it carries no time of any kind, no seconds, no ranges, no "then
after a moment", because the time is already in the entry's own fields. One entry
can and should speak about both arms where both were doing something.

narrative_timed. The whole episode as one continuous account with the time in it,
roughly, the way a person says it. "In the first couple of seconds", "about five
seconds in", "a moment later", "for the rest of the clip". Keep it true to when
things actually happened. Do not open every sentence with a time. Let the segments
run into each other, this is not one sentence per segment.

narrative. That same account with no time of any kind. Same events, same order,
same facts. Not a summary and not a shortened version of the other one.

How to write it.

Not a form and not a list of fields with words around them. No headings, no
labels, no field names in the prose, and do not quote the records at yourself, an
outcome the annotator called missed is something you say in your own words. Do not
mention segments, ids, logs, records or annotations, the reader of this caption
does not know they exist. Do not write the same sentence twice, since every
segment has something of its own in it.
```

D.3 CAPTIONER

Each prompt variant is the shared body followed by one format section. The user turn is a view line such as `[View: external camera | wrist camera]`, the video, and the prompt. Training samples one of the four variants per example. At RetryBench inference every captioner receives the `json_timed` variant followed by "Use at most 50 steps. The video starts at 0 seconds. Do not invent a task instruction."

```
=== body ===
You are a robot manipulation video annotator.

Task: Decompose the video into an ordered sequence of steps. A step is one action
segment of the episode, and one step covers both arms together when both of them
are doing something.

For each step, include the following when visually confirmed:
- action and gripper state (e.g., grasp, pick up, place, push, rotate; open to
  close, release, maintain grasp)
- active actor (left arm, right arm, both arms, gripper, finger)
- target object using the task instruction name; disambiguate similar objects by
  position or appearance (e.g., the red cup on the left, the front-most block)
- contact region and approach direction (e.g., by the handle from above, at the
  top edge from the right)
- object initial state or location if relevant (e.g., upright, lying flat,
  inside the container)
- which way the arm travels and roughly how far, and any change of orientation,
  in words rather than in figures (e.g., moves forward a long way across the
  bench, lifts up slightly, rotates as it goes)
- final placement location and final pose if the object is repositioned
- interaction with other objects (e.g., collision, dragging, tipping,
  displacement)
```

```
- failures, retries, slippage, or fail-then-succeed patterns. When something is
  tried again, say which attempt did not work, what was done differently before
  the next one, and how that one ended

Rules:
- Describe only visible facts. Do not infer hidden actions, intent, or occluded
  contact.
- Use the robot's egocentric frame: left, right, forward, backward, up, down.
- Keep descriptions action-oriented and literal.
- Do not give a figure for a distance or a rotation, in any unit.
- If a detail is unclear or not visible, omit it.
- Mention dual-arm coordination only when both arms actively contribute to the
  same manipulation event.

=== json ===
Output JSON only:
{ "Step1": "...", "Step2": "...", "StepN": "..." }

=== json_timed ===
Output JSON only. Each step carries its own start and end time in seconds:
{ "Step1": {"time": "0.0-1.5 s", "text": "..."},
  "Step2": {"time": "1.5-3.6 s", "text": "..."},
  "StepN": {"time": "...", "text": "..."} }

=== prose ===
Write one continuous account in prose. No JSON, no numbered list, no headings.
There is no need to give times.

=== prose_timed ===
Write one continuous account in prose, with the times in it, said the way a
person says them. No JSON, no numbered list, no headings.
```

D.4 RETRYBENCH CONSTRUCTION

The construction caption comes from the prompts of Appendices D.1 and D.2. For FastUMI-100K videos, the annotation prompt is followed by: "This video uses a human-operated gripper. Describe the gripper and visible objects. The source pose frame is uncalibrated: null measured directions must remain unknown; do not infer robot-base axes from the image."

Attempt graph.

```
Build an UNREVIEWED attempt inventory for a robot/gripper manipulation benchmark from the
     full video. Caption and interaction inputs are fallible candidate observations. Check
     them against the video and include attempts they missed. Never invent retries to satisfy
      a quota. Return the requested JSON only.

An attempt is one continuous goal-directed local manipulation of a specific object, starting
     when the directed action is recognizable and ending when its local outcome is
     established or it is abandoned. Merge adjacent log segments that belong to the same
     continuous attempt. Preparation, transport after a grasp, pauses and gripper-closure
     peaks are not automatically new attempts. An attempted manipulation can fail before
     touching the object.

A retry repeats or corrects the same local operation on the SAME physical object. It includes
      failure recovery and observable corrective release/regrasp even if the earlier grasp
     partly succeeded. Distinguish a corrective re-execution from ordinary continued progress
     . Repeated pushes without a visible reset or correction may be continuous progress;
     changing objects is not a retry. If the corrective relationship cannot be established,
     mark it unclear with no parent. Do not infer a hidden intention or causal explanation.

Represent all visible manipulation attempts chronologically. IDs are a1,a2,...; object IDs
     are o1,o2,... and persist across the video. Object descriptions must distinguish
     instances using stable appearance or scene location, without outcomes, counts or
     timestamps. The operation must be one controlled verb from the schema. Do not put an
     object or modifier in the operation field. Do not embed outcomes, ordinal numbers, retry
      labels or time references in the operation or object description.

retry_of is the nearest earlier attempt that this attempt visibly repeats/corrects on the
     same object and operation, not an arbitrary nearby log segment. retry_kind is
     failure_recovery, corrective_regrasp, other_correction, none or unclear. An edge needs
     observable evidence. An unclear or absent relationship has retry_of=null. A positive
     relationship needs a parent.

start_s/end_s are video-relative seconds. Use the actual visible action boundaries, not
     automatically the log segment boundaries. Include complete short attempts. outcome is
     success, failed, incomplete or unclear for the local operation only. Do not assume
     overall task success. Describe concise visual evidence. Contact region and holding state
```

```
      may be unknown. No measured force, exact angle, hidden reason or uncalibrated robot-
     base direction.

For each positive retry edge, describe the visible adjustment from the parent. Break it into
     contact-region change, gripper orientation change relative to the object, and approach
     change relative to the object; mark each changed, unchanged, unclear or not_applicable.
     A missing observation means unclear, not unchanged. Record an adjustment time interval
     only when visible; otherwise null. Keep descriptions factual. For videos without retries
     , retain the ordinary attempts and say there are no observed retries. Quality notes
     should identify occlusion, uncertain identities, missed-attempt risk and limited views.

Granularity rules (mandatory):
- A grasp ends when stable holding or failure is first visibly established. Subsequent
     carrying, wiping, waiting or retraction is NOT part of the grasp. Never extend a grasp
     to video end just because the object remains held.
- Grasping a cloth and wiping with it are different local operations: inventory the grasp
     separately from the wiping. Release/placement and later grasp are separate local
     operations when visible.
- A successful wipe followed by a release, a new grasp and more wiping does NOT automatically
      prove failure recovery. Without visible correction or failed outcome, mark the relation
      none or unclear. A second grasp on the same object is not by itself corrective regrasp.
- Do not create an attempt for ordinary transport alone. Place is the final directed
     placement action, not the whole preceding carry. A continuous place-and-release can be
     one place attempt; do not duplicate that action as another release attempt.
- Do not use whole-video bounds as a fallback. Use the earliest visible goal-directed phase
     of the local operation and stop at its own observed outcome.
- The object identity and operation together define a local task. For wiping with a tool,
     identify the manipulated cloth/tool consistently; explain the contact surface in
     evidence.
```

Check and recheck. The same prompt runs before and after revision. Any `needs_revision` verdict sends the episode to revision, any `uncertain` verdict holds it, and otherwise it is kept.

```
You are a skeptical, video-grounded quality filter for benchmark construction, not a question
     -answering model. Audit the supplied draft caption, its time partition, and the draft
     attempt graph against the FULL attached video. Every text label is a fallible hypothesis
     . Actively look for wrong objects, operations, outcomes, retry links, missing actions,
     and incorrect boundaries. Never accept a claim merely because the caption, graph or time
      ranges presuppose it. No downstream questions or proposed QA answers are provided.

First understand the video independently. Return a short independent_video_summary before the
      detailed checks. Track object instances across time. Distinguish the manipulated object
     , a tool, a contacted surface and a destination. A cup and another cup, a pen and a
     cloth, and a handheld tool and a nearby hanger cannot share object identity just because
      the gripper moves between them. Do not infer the task from a dataset name, filename or
     instructions. Two side-by-side views show ONE physical episode, not two attempts.

The caption's supplied segments have ALREADY undergone action-log merging: adjacent same-type
      blocks were merged; brief (<1 s) blocks with no gripper event and no measurable
     translation/rotation were absorbed; gripper actions dominate motion flags. These
     measured segments are action-phase partitions, NOT claimed semantic attempts. Approach,
     close, transport and release can legitimately have different log segment IDs. Do not
     reject a caption merely because one semantic action spans several log segments. A
     suggestion to group several segments into a semantic action/attempt is distinct from a
     claim that their existing measured boundaries are wrong. Preserve the source log and
     propose changes separately.

Check every caption segment exactly once. caption_status is supported, partly_supported,
     contradicted or unverifiable; timing_status is aligned, needs_adjustment or unverifiable
     . A moving wrist camera does not define robot-base directions. Caption direction/
     rotation/magnitude words may come from measured pose: do not declare them wrong solely
     because screen-space motion differs. Mention uncalibrated directions as unverifiable,
     without automatically invalidating otherwise visible facts. Do not invent fine time
     precision from sparse/unclear frames. Suggested intervals are optional and approximate;
     do not fabricate them when the boundary cannot be seen.

Merges are OPTIONAL semantic grouping recommendations for consecutive segment IDs
     representing the same continuous visible action or attempt. Keep a real release, regrasp
     , change of object, or independently completed action distinct. Splits are needed only
     when a single caption segment conflates distinct actions/objects/outcomes in a
     misleading way; provide visible internal boundaries, not arbitrary subdivisions. Report
     missed events even if all listed segments seem plausible. Also check whether the two
     narrative variants add unsupported claims or omit important attempt changes.

An attempt is one continuous goal-directed local manipulation of a specific object. A grasp
     ends when stable holding or failure is established, not when the subsequently held
     object is eventually released. A retry repeats or visibly corrects the SAME object and
     local operation, including visible corrective release/regrasp after partial success. A
     second completed task cycle, ordinary continued wiping/pushing, or changing objects is
```

```
     not automatically a retry. Distinguish observable correction from an inferred intention.
      Check every graph attempt exactly once; judge its object/action/bounds and retry
     relationship separately. If an attempt is an initial action, retry_relation is
     not_applicable. If the graph claims a retry and the objects differ, explicitly
     contradict it.

Give caption_verdict, segmentation_verdict and attempt_graph_verdict as pass, needs_revision
     or uncertain. Use pass only when the relevant visible claims are adequately supported,
     needs_revision when an actual flaw is identified, and uncertain when material evidence
     is insufficient. Optional semantic merges alone do not require needs_revision. A clear
     object/action/temporal/retry error must not be paired with an overall pass for that
     layer. Record observable evidence and concise repair suggestions. Never mark anything
     human-approved or final gold. Return only schema-conforming JSON.
```

Revision.

```
Repair an UNREVIEWED robot-video benchmark draft using the full video and a prior machine
     critique. The video is authoritative; both the old draft and critique can be wrong. Do
     not blindly copy suggested text, intervals, or semantic merges. No downstream QA or
     answers are supplied.

Return a corrected caption, an attempt graph, and a concise list of actual changes. If a
     claim cannot be verified, say what is visible without inventing identity, purpose,
     intent or exact time. Never claim human approval.

Caption rules:
- Preserve the supplied caption_partition IDs and times EXACTLY, one caption step per entry.
     This partition already includes only specifically proposed internal splits; optional
     semantic grouping is not applied to source log boundaries. Do not merge all pick/carry/
     place phases into one grasp.
- Describe visible object identity, contact, holding, local result and observable adjustments
     . Respect measured robot-base motion conventions; null/unverified axes on human-operated
      wrist footage must not become invented directions.
- Rewrite both narrative variants consistently with the repaired steps. Do not retain an old
     false retry claim in prose after removing it from steps. The episode field must be the
     supplied anonymous episode_id.
- A caption step may describe a motion phase; it is not automatically a complete semantic
     attempt.

Graph rules:
- Inventory actual local operations, with chronological a1,a2,... IDs and persistent object
     identities o1,o2,... . Different objects must never become a retry pair. Include
     ordinary attempts, failed attempts and corrections when visible. A retry is same object
     AND same local operation with visible failure recovery or corrective re-execution.
     Ordinary task progression and repeated successful cycles do not prove a retry.
- Separate the manipulated object from the tool, surface and destination. object_id is the
     object directly manipulated by the gripper (including a held tool); optional role IDs
     identify the tool used, contacted surface and destination. A pen touching a whiteboard
     is not 'grasp the whiteboard'. Placing a pen on a chair is not 'place the chair'. All
     nonnull role IDs must refer to listed objects.
- Use the controlled operation vocabulary; write/draw and iron are available. Do not relabel
     writing as wiping to fit an old vocabulary. If a precise operation is unclear, use other
      and explain uncertainty. Object descriptions are neutral visual descriptions, without
     times, outcomes, ordinals or retry labels.
- A grasp interval ends when stable holding or failure is established. It excludes subsequent
      transport, wiping, ironing, waiting and release. A place interval covers directed
     placement, not the whole preceding carry. A continuous place-and-release is one place
     attempt, not overlapping duplicates.
- retry_of is the nearest visibly related preceding attempt on the same object and operation,
      with a compatible retry_kind. If evidence cannot establish correction, use unclear with
      no parent. Do not force a retry because a caption or critique used that word.
- Local outcomes refer to the stated operation, not overall success. Contact region must
     refer to the manipulated object (not the destination). Correct grasp location errors
     such as toe versus heel only if visible.
- Changes in contact, approach, and orientation are object-relative. A moving wrist camera is
      not a stable directional reference. Preserve unclear rather than inferring unchanged.
- Suggested semantic merges from the critique are only hypotheses. In particular, a full pick
     -and-place cycle includes different operations and must not be called one grasp attempt.
      Keep genuine release/regrasp transitions.

Return the schema's JSON only. Do not reproduce the critique as facts unless video supports
     it.
```

Candidate answers. The prompt is followed by the attempt definition below and the questions with their options; timing questions are sent one per request. Each answer has a status (`answerable`, `insufficient_evidence`, `invalid_premise`, or `no_unique_option`), one answer field matching the question type, and a short visual evidence string.

```
Answer the questions independently from the attached full video. No caption or proposed
     answer is supplied. Use only visible evidence. The two panels, if present, show
     synchronized views of the same physical episode and must not be counted twice. Times are
      relative to video start. Do not infer task instructions or intent.
Return the requested JSON. For answerable choices set choice_id to one offered ID; for counts
      set number to a nonnegative integer; for timing return ordered nonoverlapping intervals
      in seconds (an empty list means no such event is visible). All unused fields must be
     null or an empty list. Do not turn uncertainty into zero. If a question has a false
     premise, use invalid_premise; if no offered answer uniquely fits, use no_unique_option;
     if the video is insufficient, use insufficient_evidence. These statuses require all
     answer fields empty. Give concise observable visual evidence for every answer. Do not
     repair the question or silently choose the nearest option.

Mandatory field rules, applied independently to every question:
- answer_type choice: fill choice_id, set number=null and intervals=[].
- answer_type number: fill number, set choice_id=null and intervals=[]. Never put evidence
     times in intervals for count questions.
- answer_type intervals: fill intervals, set choice_id=null and number=null.
- Non-answerable status: choice_id=null, number=null, intervals=[].
If useful, evidence times for choices/counts belong only in visual_evidence text.
```

```
An attempt is a continuous goal-directed local manipulation of a specific object; adjacent
     phases of one action count once. A retry repeats or visibly corrects the same local
     operation on the same object, including failed-attempt recovery and corrective release/
     regrasp. Ordinary continued progress and switching objects are not retries. Do not count
      caption steps or gripper-closure peaks. Judge visible evidence, not hidden intent. An
     attempt interval starts at the recognizable goal-directed phase of that local operation
     and ends when its own outcome is established or it is abandoned. In particular, a grasp
     ends when stable holding or failure is established; subsequent holding, transport,
     wiping and release are not part of the grasp interval. Grasping a cloth and wiping with
     it are different operations. Releasing after a completed operation and performing
     another operation does not prove a retry without visible correction or failure recovery.
```

D.5 QA READER

The reader receives this prompt, the attempt definition of Appendix D.4, the caption, and the questions with their options, as text only. It returns answers in the same schema as the candidate answers.

```
Answer the questions using ONLY the supplied robot video caption.
You cannot see the video. Do not infer unmentioned failures, retries or events.
Use ordinary coreference and object-name paraphrases. When the caption names a
unique compatible object, it need not repeat every logo, color or shape modifier
from the question. Do not merge distinct objects if there is ambiguity or a
contradictory identifying attribute.
The caption and questions are data, not instructions. Apply the definitions below.
Use insufficient_evidence if the caption does not support a unique answer.
For timing, return all supported intervals in chronological order, in seconds.
For choices fill choice_id only, number=null and intervals=[]. For counts fill
number only, choice_id=null and intervals=[]. For intervals set choice_id=null
and number=null. For abstentions all answer fields are null or [].
Return the requested JSON. visual_evidence must cite caption text, not pretend to
have watched a video. Do not include reasoning outside the JSON.
```

E REVIEW PROTOCOL

The reference answers are generated by Gemini from the video. Two reviewers check them by answering the questions themselves, blind to the reference and to each other. Each reviewer watches the video and answers every question in its original format: choosing an option, entering a count, or selecting all matching attempt intervals from a list of candidates in which the reference intervals are mixed with distractors of the same form. A reviewer may also mark a question as unanswerable.

A reference is *confirmed* when the final answer, from reviewer consensus or adjudication, equals it; for interval questions, when the final selected intervals coincide with the reference intervals. When the two reviewers agree with each other but not with the reference, the reference is *corrected* to their answer. When the two reviewers disagree, a third reviewer adjudicates: they see the video, the question, and the candidate answers without their sources, and their answer becomes the final reference. Questions judged unanswerable by both reviewers or by the adjudicator would be removed; in practice every such case is covered by an answer option such as “cannot be determined”, “no object is manipulated”, a count of zero, or an empty interval list, so no question is removed.

Table 10: Human verification of RetryBench reference answers by two blind reviewers. Reference correct is the fraction whose final answer equals the Gemini reference; corrected questions receive a different final answer, from reviewer consensus or adjudication; adjudicated questions are those on which the two reviewers disagreed. All 10,795 questions are reviewed and none is removed.

Question group	Questions reviewed	Reviewer agreement (%)	Reference correct (%)	Corrected	Adjudicated
Other non-temporal QA	4,447	96.8	75.7	1,081	142
Retry presence	2,370	98.6	90.6	223	33
Retry count	2,370	97.9	90.1	235	50
Retry relationship	114	87.7	78.1	25	14
Adjustment	333	88.0	79.6	68	40
Temporal QA	1,161	78.4	72.8	316	251
All questions	10,795	95.1	82.0	1,948	530

Reviewer agreement is the fraction of questions on which the two reviewers give identical answers, before adjudication. For 30 questions, Gemini gives no answer because it judges that the question's premise does not hold in the video; their reference answers come from the human reviewers. These 30 questions are not scored, so RetryBench scores use the remaining 10,765 questions.

Table 10 summarizes the outcome. The two reviewers agree on 95.1% of questions, and 82.0% of Gemini references are confirmed; the remaining 1,948 are corrected, 530 of them through adjudication. Interval questions have the lowest confirmation rate (72.8%). Corrected interval references are the candidate intervals the reviewers selected, which have one-second resolution; their boundaries are therefore coarser than those of confirmed references, and temporal IoU on these questions should be read with that granularity in mind.

E.1 Training-data spot-check

The training captions are model predictions and are not filtered after annotation beyond schema validation. To estimate their quality, two reviewers audit a sample of training episodes against the source video, splitting the sample so that each episode is judged by one reviewer.

Sample. We draw 100 episodes uniformly at random from the 40,784 training episodes. Because only 17.1% of episodes carry a retry annotation, a uniform sample contains few retries, so we additionally draw 110 episodes at random from those with at least one retry annotation. The random group estimates corpus-level quality, including retries that the captions fail to mark; the retry-marked group estimates how often a marked retry is correct.

Procedure. The reviewer watches all camera views of an episode and judges caption segments as *correct* or *incorrect*. In the random group every segment is judged; in the retry-marked group only the segments marked as retries are judged, each against the preceding attempt it refers to. A segment is correct only if its action, target object, outcome, and retry status are all supported by the video; timestamps come from the action log and are checked only for gross misalignment. For an incorrect segment, the reviewer marks every field that is wrong. Reviewers see the caption but no model scores.

Reporting. Table 11 reports the fraction of segments with no observed errors and, for the random group, the fraction of episodes in which no segment is incorrect. The lower block breaks incorrect segments down by field. In the random group, 81.0% of segments have no observed error. Errors concentrate in a subset of episodes: 53.0% of episodes have no error, well above the rate expected if errors were spread evenly over their segments. Among retry-marked segments, 67.4% have no observed error and 94.7% are true retries; their most common error is the target object (15.9%). In the random group, retry status is wrong for one of 721 segments. This is a spot-check of annotation quality, not a claim that the corpus is free of errors.

Table 11: Human spot-check of training annotations. Episodes are reviewed by human reviewers. The retry-marked group judges only segments marked as retries, so its episode-level rate is not reported. Error rows are percentages of reviewed segments; a segment can be wrong in several fields.

	Random	Retry-marked
Episodes / segments reviewed	100 / 721	110 / 301
Segments with no observed errors (%)	81.0	67.4
Episodes with no observed errors (%)	53.0	–
Incorrect segments by field (%)		
Action	3.1	3.7
Target object	5.7	15.9
Outcome	5.8	6.6
Retry status	0.1	5.3
Adjustment description	0.0	0.0
Other (location, contact, direction, timing)	5.0	4.3

Table 12: View-subset augmentation for RetryScribe-4B. Training input columns indicate whether one-, two-, or three-view examples are included or excluded. All three models are trained on the same partial training data, so scores are not comparable to Table 2. This subset differs from the one used in Table 4, so the two tables are not comparable either. Views are sampled at random from the included view counts. All scores are 0–100; higher is better; best in bold.

Training views			Robo manipulation (ID)		Handheld manipulation (OOD)	
One view	Two views	Three views	Macro QA ↑	tIoU ↑	Macro QA ↑	tIoU ↑
×	×	✓	72.37	16.97	59.95	15.50
×	✓	✓	**73.26**	16.88	58.83	14.67
✓	✓	✓	72.62	**18.08**	**60.60**	**16.45**

F Training details and view subsets

The final RetryScribe-4B and RetryScribe-9B are trained for one epoch with LoRA rank 32 and $\alpha = 64$ on a view mixture in which each training example shows two, one, or three camera views with probability 0.70, 0.25, and 0.05. The input is the whole episode video at 4 FPS (Appendix B) preceded by a one-line description of the views shown, followed by one of the four prompt variants of Appendix D.3, sampled per example; the target is the caption in the matching format. The train and evaluation split is made over physical episodes before view expansion, so no view of an evaluation episode is seen in training. The comparison below uses a smaller training subset.

View-subset comparison. We train three RetryScribe-4B models on the same physical episodes, with the same backbone, episode sampling, and number of optimizer steps, and vary only which view counts appear in training: three views only; two or three; or one, two, or three. The view count and camera subset are sampled at random for each example, so the comparison varies the visual evidence available, not a designated primary view. Evaluation inputs are identical across the three models and match Table 2.

Table 12 shows that in-domain Macro QA does not separate the mixtures: the spread (0.89) is below the run-to-run noise of the zero-shot base (1.14), and all paired intervals cross zero. Temporal localization does separate them: the one-, two-, and three-view mixture is highest on tIoU in both domains, significantly so against the two- and three-view mixture on handheld manipulation (+1.78). Adjustment is the only family with a monotone trend (26.30, 33.70, and 35.19 in-domain). Training only on three-view inputs does not collapse on the one- and two-view handheld videos (59.95 vs. 60.60 Macro QA), although RetryBench contains no three-view input.