EgoErrorVQA: Assess Egocentric Comprehension Capabilities Through Procedural Errors For Ego-Agentic AI

Anonymous ACL submission

Abstract

An increasing number of intelligent systems interact with daily human activities, making robust egocentric visual information processing essential. However, existing benchmarks for Visual Agents and Visual Language Models (VLMs) primarily focus on third-person perspectives or capture only short-term visual understanding, limiting their ability to model long-horizon, action-centric procedures. To bridge this gap, we propose EgoErrorVQA, the first visual question answering (VQA) task designed for egocentric procedure comprehension with explicit modeling of procedural errors that reflect common execution failures in real-world tasks. EgoErrorVQA evaluates a range of models using both open-ended and multiple-choice questions, revealing persistent weaknesses in handling procedures with stepwise logical dependencies. In addition, we develop a user-friendly evaluator agent based on the Agent2Agent (A2A) protocol, enabling rigorous and standardized evaluation of visual agents through VQA-based interaction. Finally, we introduce EgoError-CoT, a training-free framework that improves reasoning through in-context learning and task-specific chain-of-thought prompting, delivering consistent gains without additional training.

1 Introduction

With the rapid progress of visual agents, VLMs, and multimodal large language models (MLLMs), benchmarks must keep pace with emerging agent capabilities. Egocentric video understanding is particularly important as it captures the world from a first-person perspective (Plizzari et al., 2024) and underpins agentic and embodied applications (Fung et al., 2025). A key yet under-evaluated requirement in this setting is procedural understanding (Li et al., 2025b): many everyday tasks involve executing interdependent steps under ordering constraints (e.g., assembling a toy car), which demands step-level comprehension, long-horizon memory, and temporal reasoning beyond recognition or captioning (Ging et al., 2024). Moreover, robust assistance requires not only following procedures but also detecting failures, such as out-of-order steps, omissions, wrong-object usage, and redundant actions, motivating evaluation of procedural error detection (Flaborea et al., 2024) with practical relevance to smart homes and factories.

During a task like making a sandwich (take bread $\rightarrow$ spread mayo $\rightarrow$ add ham $\rightarrow$ close sandwich), procedural error detection requires an agent distinguish step omissions (forgetting the ham), out-of-order execution (closing before adding ham), redundant actions (spreading mayo twice), or wrong-object usage (grabbing the wrong condiment). However, existing benchmarks fall short in two key respects. First, despite the growing interest in egocentric procedure understanding (Bansal et al., 2022), procedural errors are rarely characterized in a way that enables systematic evaluation. Besides, existing datasets were not specifically designed for procedural error detection and thus lack detailed error annotations and comprehensive error taxonomy, therefore, difficult to use directly. Second, evaluation pipelines are often cumbersome and difficult to reproduce: users may need to assemble scattered scripts, convert outputs into specific formats, and run separate components for scoring and reporting. This fragmented process makes it costly to evaluate even a single agent, limiting usability and hindering broader adoption.

To address these limitations, we propose EgoErrorVQA, a benchmark that evaluates egocentric video comprehension, memory, and reasoning by formulating procedural error detection as a VQA task, along with an eight-category error taxonomy covering nearly all common procedural error types. As illustrated in Fig.1, we package EgoErrorVQA as an evaluator agent (green agent) that interacts with the agent under test (white agent) via a unified

Benchmark	Videos	QA-pairs	Procedural	Aciton Label	Error Label	LLM Scoring	Multiple Scenarios	Open-end Question	Multiple-choice Question
EgoVQA (Fan, 2019)	16	600	×	×	×	×	✓	✓	✓
AssistQ (Wong et al., 2022)	100	531	✓	✓	×	×	✓	×	✓
EgoTaskQA (Jia et al., 2022)	2K	40K	✓	✓	×	×	✓	×	✓
EgoPlan-Bench (Chen et al., 2023b)	-	4939	✓	✓	×	×	✓	×	✓
EgoSchema (Mangalam et al., 2023)	-	5063	×	×	×	×	✓	×	✓
VidEgoThink (Cheng et al., 2024a)	195	600	×	✓	×	✓	✓	✓	×
OpenEQA (Majumdar et al., 2024)	-	1636	×	×	×	✓	✓	✓	×
ProMQA (Hasegawa et al., 2025)	384	401	✓	✓	✓	✓	×	✓	×
EgoTextVQA (Zhou et al., 2025)	1507	7064	×	×	×	✓	✓	✓	×
EgoErrorVQA (Our)	800	5417	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison between EgoErrorVQA and common video benchmarks. Videos represents the number of original videos.

Figure 1: Evaluation process of EgoErrorVQA, in which establishing communication with the system by the evaluatee is sufficient to automatically trigger the assessment.

Benchmark	Source	Clips	Correct samples	Error samples	QA-pairs
Open-end	CaptainCook4D	960	632	328	3560
	EgoOops	215	175	40	
	Epic-tent	184	124	60	
	Assembly101	446	341	105	
Multiple-choice	CaptainCook4D	1000	660	340	1857
	EgoOops	215	175	40	
	Epic-tent	182	123	59	
	Assembly101	460	352	108	

Table 2: Statistics of EgoErrorVQA.

Agent2Agent (A2A) protocol[1] and fixed interfaces: the evaluator serves information such as the video path together with both open-ended questions (e.g., action correctness) and multiple-choice questions (e.g., error-type identification), collects the model's responses, and automatically produces evaluation results. In practice, users only need to establish an A2A connection to the evaluator agent, after which the full assessment is triggered and executed end-to-end, substantially improving usability and reproducibility.

Finally, we propose a training-free framework that leverages in-context learning and chain-of-thought (CoT) reasoning, and achieve substantial performance improvements on this task. In summary, our main contributions are as follows:

• We introduce EgoErrorVQA, a novel task for egocentric procedure comprehension across diverse scenarios. This task contains a new benchmark and dataset, which, to the best of our knowledge, is the first benchmark explicitly dedicated to procedural error detection and VQA in egocentric setting as shown in Table. 1.

• We encapsulate EgoErrorVQA as an evaluator agent with fixed interfaces to streamline reproducible evaluation of external agents and VLMs, and integrate a scoring suite that combines LLM-based judgment (open-end) with general evaluation metrics (multiple-choice) to assess not only semantic quality and creativity, but metric-based performance, making the evaluation more comprehensive and broadly applicable.

• We conduct extensive experiments on EgoErrorVQA across a variety of models, validating the benchmark and revealing that current agents and VLMs still lag behind humans in procedural error understanding. As a simple yet effective baseline, we propose EgoError-CoT, a training-free prompting strategy in combination with in-context learning that improves procedural comprehension.

2 Related Work

Egocentric video benchmark. Existing egocentric video benchmarks for VQA models typically involve simple questions or tasks such as action recognition and object identification (Cheng et al., 2024b; Majumdar et al., 2024; Jia et al., 2022), offering only short-term assessments of reasoning and cognition. Some works consider more complex tasks like task planning (Chen et al., 2023b), which better probe reasoning and memory. As shown in Table. 1, our work focus on egocentric procedure

[1]More details about A2A protocl are available at https://a2a-protocol.org/latest/

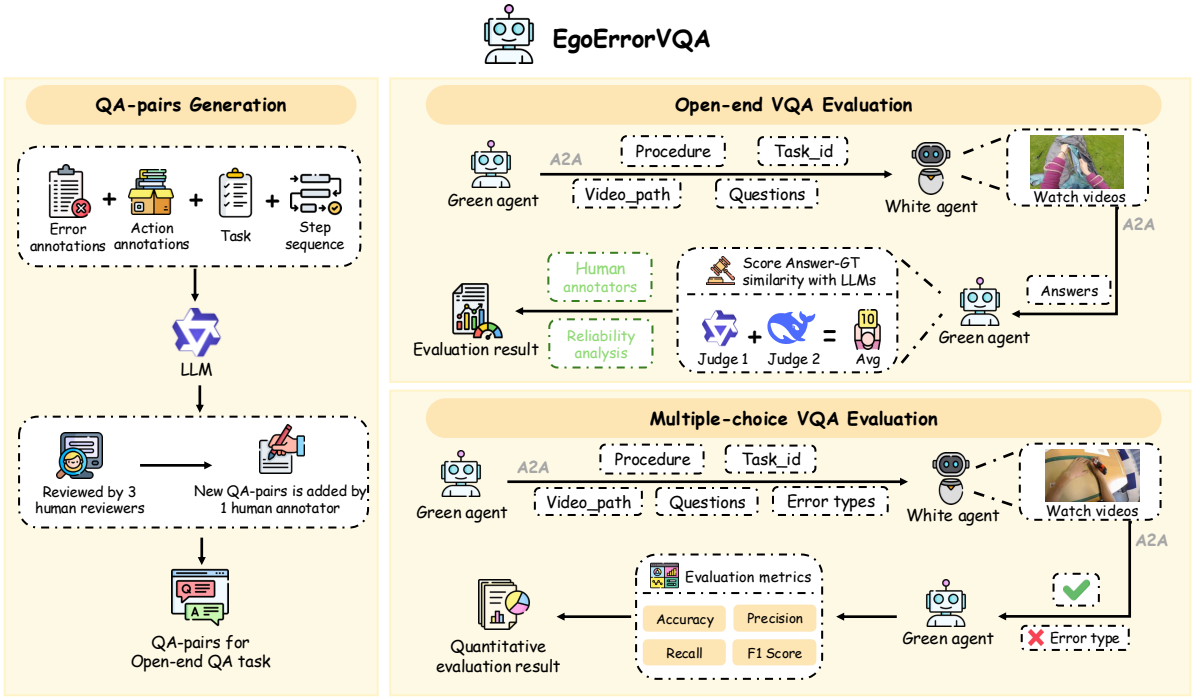


Figure 2: Overview workflow of EgoErrorVQA. In the QA-pairs Generation stage, it outlines the overall procedure for constructing QA-pairs and highlights the stages and roles of human involvement, we use Qwen2.5-7B-Instruct to generate in this work. Subsequently, the Open-end VQA Evaluation and Multiple-choice VQA Evaluation sections illustrate the overall pipelines for those two evaluation tasks, respectively.

comprehension, a task that is more challenging and insightful, covering the two most common question types in VQA.

Egocentric procedural error detection. As egocentric datasets proliferate (Haneji et al., 2025; Peddi et al., 2024), downstream tasks have become increasingly diverse, with egocentric procedural error detection (Wang et al., 2023) emerging as a key challenge. Such task is central to enabling AI assistants (Li et al., 2025b) to effectively support daily activities and even help blind people. Our benchmark targets this task, providing a framework to assess these abilities, to evaluate traditional methods (Lee et al., 2024; Huang et al., 2025) and further supports the research on interpretable error detection and classification.

LLM reasoning and in-context learning. With the rapid advancement of Large Language Models (LLMs), their reasoning capabilities have substantially improved (Ma et al., 2025), and recent studies show that chain-of-thought (CoT) prompting can further enhance their reasoning performance (Wei et al., 2022). Nevertheless, LLMs still encounter considerable difficulties in handling complex tasks, such as advanced mathematical problems(Chen et al., 2023a). In this work, we employ LLMs only for relatively simple semantic-level tasks, supported by rigorous reliability analysis. Also, we appropriately integrates CoT and in-context learning. In-context learning (Dong et al., 2024) allows models to improve task performance without fine-tuning by using contextual information or examples in the prompt.

3 Open-end VQA in EgoErrorVQA

Given that video question answering (VQA) is already a core capability of current agents and VLMs, and is highly likely to remain so in the future, we design our evaluation procedure in the form of VQA. To enable a more comprehensive and objective evaluation, EgoErrorVQA incorporates both open-end and multiple-choice VQA as shown in Fig. 2. In this section, we focus on the open-ended setting. We first introduce the data collection, the specific task formulation and the construction of QA-pairs, then present several illustrative examples for better understanding.

3.1 Data Collection

In Table. 2, Fig. 3 and Fig. 4, we select four egocentric procedural task datasets: **CaptainCook4D** (Peddi et al., 2024), **EgoOops** (Haneji et al., 2025), **Epic-Tent** (Jang et al., 2019) and **Assembly101** (Sener et al., 2022). These are among the few ego-

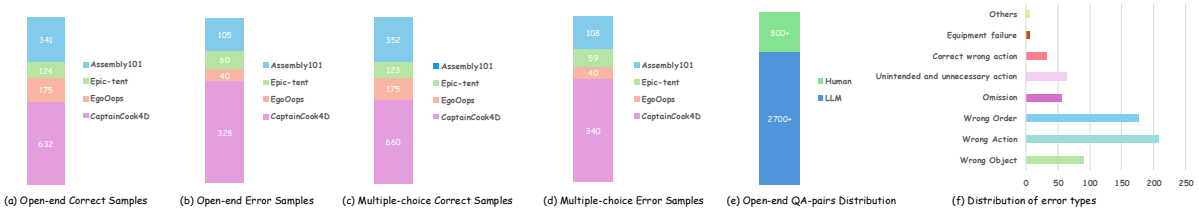


Figure 3: (a), (b), (c), and (d) respectively show the proportions of correct and incorrect samples in open-end VQA and multiple-choice VQA. (e) shows the proportion of QA-pairs in the open-end VQA that are generated by the LLM versus those added by human annotator. (f) shows the distribution of each error type.

Figure 4: Examples from the four selected datasets.

centric video datasets composed entirely of procedural tasks with explicit error annotations, and they all provide action labels, matching our needs for dataset construction and benchmarking. CaptainCook4D covers 24 cooking recipes; EgoOops includes five procedural scenarios (e.g., electrical circuits, ionic reactions); Epic-Tent focuses on tent setup; and Assembly101 on toy car assembly. Considering potential future expansion of the training set, we subsample each dataset proportionally to its size, obtaining 1,000 samples from CaptainCook4D, 215 from EgoOops (40%), 184 from Epic-Tent, and about 460 from Assembly101 (40%).

3.2 Task Formulation

In the open-ended setting, EgoErrorVQA first transmits to the evaluated White agent, via a communication protocol, a Procedure that outlines the complete workflow for the task, specifying the main steps required to achieve the goal. White agent is then asked to answer carefully designed questions that probe, from multiple perspectives, whether specific steps and their ordering are appropriate. The answers are sent back to EgoErrorVQA, which performs scoring and evaluation metrics; for each question, EgoErrorVQA also provides an explanation of the rationale behind the assigned score.

3.3 QA-Pairs Generation

Rigorous evaluation of a model's understanding of procedural tasks cannot rely on generic questions such as "Is there anything wrong in this video?" Because the video clips contain a lot of redundant information, the agent struggles to focus on critical procedural elements. These responses thus provide little evidence of hierarchical procedural understanding or of stepwise logical reasoning, and do not support meaningful evaluation. Targeted and deliberately confounding questions about specific actions and steps are therefore required to obtain data that more accurately reflects the agent's understanding of procedures and procedural errors.

For example, for a correctly executed step, a question such as "Did I add chopped cilantro to the ramen bowl as instructed?" directly targets that action and requires the model to reason about it. Similarly, one might ask "Did I use the wrong part when attaching the roof to the body?", while the actual error occurs elsewhere in the video, thereby introducing a controlled distractor.

We adopt an **LLM–Human collaboration strategy** for QA-pair generation. **Qwen2.5-7B-Instruct** (Bai et al., 2025) first labels each procedural step by task type and existing annotations, then generates three diverse QA-pairs per sample. Around 6,000 QA-pairs are independently checked by three human annotators over three weeks, who remove or revise those that fail to test procedural error understanding or focus on irrelevant content. A final annotator adds QA-pairs for missing evaluation aspects and introduces deliberately misleading questions about correctly executed steps (e.g., "Did I add any unnecessary steps when I attach engine to chassis?").

In total, EgoErrorVQA contains 3,560 QA-pairs covering 1,805 samples, with each sample associated with 1–3 QA-pairs from which one is randomly selected during evaluation. Approximately

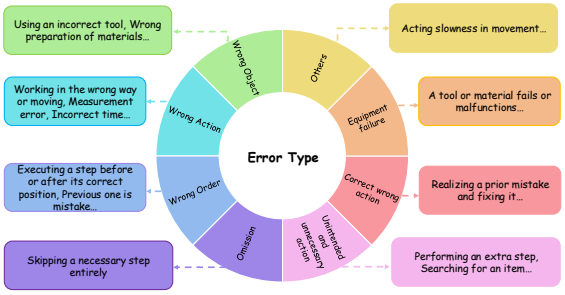


Figure 5: Eight error types defined in this work and, for each type, provides several concise illustrative examples.

2,700 QA-pairs are generated by an LLM and subsequently refined through human review, while about 800 are manually authored, shown in Fig. 3.

3.4 Evaluation Metric

Given the extensive evidence supporting the effectiveness of LLM-as-a-Judge (Li et al., 2025a), we adopt a novel scoring scheme in our evaluation. To facilitate reproducibility and broader adoption, we employ two open-source, memory-efficient LLMs, **Qwen2.5-7B-Instruct** (Bai et al., 2025) and **DeepSeek-LLM-7B-Chat** (Bi et al., 2024) as judges.

To ensure reliability and mitigate subjective bias, the judging models do not perform complex reasoning over the answers. Instead, they only assess the semantic similarity between the model-generated answer and the ground truth. Also, we design a rigorous reliability analysis of LLM scoring in subsequent sections. To further reduce model-specific bias, both Qwen and DeepSeek provide Sim. that represent semantic similarity, and we take their average as the final score. Sim. is rated on a 0–5 scale, where **5 = Perfect match**, **4 = Minor error**, **3 = Partially correct**, **2 = Mostly wrong**, **1 = Wrong**, and **0 = No response**. Under this scheme, the resulting scores are both reasonable and credible, effectively reflecting a model's understanding of procedural errors.

4 Multiple-choice VQA in EgoErrorVQA

Semantic-similarity-based metrics prevent fair comparison between current agents and traditional egocentric procedural error detection methods. To address this, we introduce a multiple-choice VQA task to more rigorously assess VLMs' and agents' understanding and recognition of error categories, thereby mitigating the key limitation of interpretability in downstream error detection.

Method	frames	Cook	Tent	Assem	Oops	Avg-Sim.
Human	-	3.79	3.89	3.91	3.47	3.77
GPT-4o-mini	8f	2.84	3.18	3.33	3.33	3.05
Gemini-2.5-flash	8f	2.82	2.48	2.89	3.39	2.87
7B / 8B						
LLaVA-OneVision	8f	2.85	3.17	3.82	3.34	3.18
	16f	2.89	3.19	3.75	3.35	3.19
	24f	2.91	3.25	3.73	3.37	3.20
	32f	2.94	3.22	3.83	3.42	3.25
Vinci	8f	2.98	3.42	3.74	3.61	3.29
	16f	2.97	3.36	3.82	3.64	3.30
	24f	2.97	3.20	3.70	3.63	3.25
EgoGPT	8f	2.55	2.96	3.54	3.24	2.92
	16f	2.59	2.89	3.52	3.21	2.92
	24f	2.54	2.92	3.56	3.21	2.91
Video-LLaVA	8f	2.85	3.12	3.08	3.04	2.96
Video-LLaMA2	8f	2.89	2.41	3.27	3.16	2.97
	16f	2.85	2.67	3.22	3.16	2.96
	24f	2.86	2.83	3.18	3.18	2.97
	32f	2.82	2.59	3.10	3.14	2.90
Qwen2-VL-7B-Instruct	8f	3.01	3.37	3.83	3.62	3.32
	16f	3.02	3.27	3.81	3.59	3.31
	24f	3.01	3.26	3.94	3.56	**3.33**
	32f	3.01	3.14	3.87	3.57	3.30
Qwen2.5-VL-7B-Instruct	8f	3.11	2.90	2.55	3.53	3.00
	16f	3.14	2.95	2.57	3.47	3.02
	24f	3.18	3.05	2.59	3.50	3.06
	32f	3.14	3.01	2.62	3.44	3.03
Qwen3-VL-8B-Instruct	8f	3.03	2.75	2.74	3.48	2.98
	16f	3.02	2.85	2.80	3.56	3.01
	24f	3.00	2.84	2.80	3.47	2.99
	32f	3.01	2.85	2.78	3.56	3.00
30B / 32B						
Qwen3-VL-32B-Instruct	8f	3.09	2.70	2.75	3.28	2.99
	16f	3.08	2.94	2.80	3.29	3.02
InternVL3.5-38B-Instruct	8f	2.38	2.54	2.93	2.95	2.60
	16f	2.35	2.49	2.90	2.87	2.56

Table 3: Models' performance on open-end VQA. **Cook** denotes results on CaptainCook4D, **Tent** on Epic-tent, **Assembly** on Assembly101. **Avg-Sim.** denotes the sample-size–weighted average over all tasks. Details of Human performance shown in Appendix. A.

4.1 Error Type

We categorize the error types into the following eight classes: **Wrong Object**, **Wrong Action**, **Wrong Order**, **Omission**, **Unintended and Unnecessary Action**, **Correct Wrong Action**, **Equipment Failure** and **Others**. Details of error type classification can be found in Appendix. B and Fig. 5.

4.2 VQA Setting

Multiple-choice VQA supplies the white agent with the task-specific procedures and error types' definitions. To limit distraction from irrelevant content, each video sample is annotated with action labels, and the agent is instructed only to determine whether an error occurs in the specified step and, if so, to identify its type. The evaluation set comprises approximately 1,859 samples and share the same data source with open-end VQA, covering 31 procedural tasks.

4.3 Evaluation Metric

For a fair comparison with traditional egocentric procedural error detection approaches (Lee et al., 2024) and to comprehensively assess the model's

Method	frames	TP	FP	TN	FN	Accuracy	Precision	Recall	F1
Human	-	19	4	66	10	85.9	82.6	65.5	73.1
GPT-4o-mini	8f	117	1173	413	140	28.5	9.1	45.5	15.1
Gemini-2.5-flash	8f	68	544	887	344	51.4	11.1	16.5	13.3
7B / 8B									
LLaVA-OneVision	8f	14	240	1183	406	64.5	5.5	3.3	4.2
	16f	19	207	1198	419	65.5	8.4	4.3	5.7
	24f	32	194	1197	420	66.2	14.2	7.1	9.4
	32f	28	219	1182	414	65.2	11.3	6.3	8.1
Vinci	8f	22	321	1093	407	60.0	6.4	5.1	5.7
	16f	25	336	1080	402	59.5	6.9	5.9	6.4
	24f	24	343	1068	408	58.8	6.5	5.6	6.0
EgoGPT	8f	12	149	1221	461	**66.4**	7.5	2.5	3.8
	16f	14	170	1209	450	65.9	7.6	3.0	4.3
Video-LLaVA	8f	57	731	763	292	44.2	7.2	16.3	10.0
Video-LLaMA2	8f	141	1417	200	85	18.4	9.1	62.4	15.8
	16f	156	1302	250	135	21.9	10.7	53.6	17.8
	24f	161	1332	212	138	20.1	10.8	53.9	18.0
Qwen2-VL-7B-Instruct	8f	139	1469	186	49	17.5	8.6	73.9	15.5
	16f	128	1447	214	54	18.4	8.1	70.3	14.6
	24f	139	1409	241	54	20.5	9.0	72.0	16.0
	32f	136	1390	253	64	21.0	8.9	68.0	15.8
Qwen2.5-VL-7B-Instruct	8f	179	1601	58	5	12.8	10.1	97.3	18.2
	16f	176	1591	70	6	13.3	10.0	96.7	18.1
	24f	186	1577	75	5	14.1	10.6	**97.4**	19.0
	32f	193	1564	79	7	14.7	11.0	96.5	19.7
32B / 38B									
Qwen3-VL-32B-Instruct	8f	145	890	589	219	39.5	14.0	39.8	20.7
	16f	139	820	636	248	41.7	**14.5**	35.9	**20.7**
InternVL3.5-38B-Instruct	8f	50	566	859	368	49.0	8.1	12.0	9.7
	16f	48	570	864	361	49.1	7.8	11.7	9.4

Table 4: Performance of different models on multiple-choice VQA; for each model, **frames** denotes the number of input sampled frames. Details of Human performance shown in Appendix. A.

understanding of different error categories, we compute **Precision**, **Accuracy**, **Recall**, and **F1 Score**, and report the corresponding confusion matrix.

Precision represents the proportion of samples predicted as procedural error by the model whose error types are correctly matched. **Accuracy** represents the proportion of correctly predicted samples among all samples of the model. **Recall** is the proportion of correctly predicted samples among all procedural error samples, and **F1 Score** is the harmonic mean of precision and recall, it is used to comprehensively evaluate the performance of binary classification models, and is particularly suitable for scenarios with imbalanced datasets. In confusion matrix, we will count TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative).Detailed definitions can be checked in Appendix. C.

5 EgoError-CoT

Procedural tasks span diverse real-world scenarios, and training on narrow, scenario-specific data cannot adequately improve model performance and may even hinder transfer to other contexts. To more accurately assess egocentric procedural error detection and classification performance, we propose EgoError-CoT, a training-free framework that uses prompt engineering and in-context learning, allows agents to flexibly adapt specific tasks and scenarios, accommodating varied situations and objectives.

5.1 Few-shot Setting

Shown in Appendix. E, through repeated experiments, we found that in few-shot learning, providing all error types with correct examples both increases GPU memory usage and biases the model, leading it to label most segments as erroneous due to the imbalance between correct and incorrect examples. In addition, using examples from a single scenario improves performance only in that scenario. Therefore, our framework provides, for each sample in a dataset, one correct and one incorrect example from the same dataset, with the incorrect example randomly selected for each test round.

5.2 CoT Setting

Different Chain-of-Thought (CoT) structures can yield different performance outcomes. We design three CoT variants: Check-list, Two-stage, and No-distraction. In Check-list, the agent systematically tests the current segment against each error type, discards mismatches, and selects the most appropriate type. In Two-stage, the agent first decides

whether an error is present and, if so, then identifies it step by step. In No-distraction, the agent first describes what occurs in the video and then compares it to what should occur, without using error-type terminology during reasoning. Full CoT prompts are provided in Appendix. E.

6 Experiment

6.1 Experiment Setting

Baseline. To conduct a comprehensive evaluation, we select two closed-source models: GPT-4o-mini (Hurst et al., 2024) and Gemini-2.5-flash (Comanici et al., 2025), three popular open-source video VLMs: LLaVA-OneVision (Li et al., 2024), Video-LLaMA2 (Cheng et al., 2024c), and Video-LLaVA (Lin et al., 2024), as representative models for general video understanding. Additionally, we include two video agents fine-tuned on egocentric data, EgoGPT (Yang et al., 2025b) and Vinci (Huang et al., 2024), to represent specialized egocentric vision agents. To support future research, we test three widely-used versions of Qwen-VL (Bai et al., 2025; Wang et al., 2024; Yang et al., 2025a), highlighting potential performance variations across iterations. Two larger models are evaluated, Qwen-VL-32B-Instruct and InternVL-38B-Instruct (Chen et al., 2024), demonstrating performance differences across parameter sizes.

Benchmark Setting. To assess the impact of frame sampling, we uniformly sample 8, 16, 24, or 32 frames within the action interval (if allowed by GPU memory), using GPUs with 24 GB memory in 7B/8B evaluation.

EgoError-CoT Setting. We evaluated Vinci in the EgoError-CoT framework with 8-frame inputs, compared three settings: few-shot, zero-shot+CoT, and few-shot+CoT, and further examined three CoT strategies under zero-shot conditions.

6.2 Benchmark Results

Open-end VQA. Table. 3 shows that the highest Avg-Sim. 3.33 is obtained by Qwen2-VL-7B-Instruct (24-frame input), followed by Vinci (16-frame input) getting 3.30, whereas humans reach 3.77. This indicates only moderate performance on open-end VQA targeting specific actions, neither of the two closed-source models showed a clear advantage and increasing the number of input frames alone yields no substantial gains. Our evaluation is primarily QA-based, and open-ended questions better match the generative nature of LLMs than multiple-choice formats. Their prompts explicitly localize specific actions or processes within the video, helping models suppress irrelevant content and thereby achieve relatively higher scores. Nonetheless, given a maximum of five, the gap between 3.77 and 3.33 remains large, revealing significant limitations in procedural understanding. Analysis of raw outputs from strong reasoning models such as Qwen3-VL shows that their step-by-step reasoning is easily distracted by irrelevant details and selected large-parameter model InternVL3.5 produces abnormally terse responses, which harms performance on this task. Consequently, open-end results alone are insufficient to assess procedural error detection and must be interpreted jointly with multiple-choice outcomes.

Multiple-choice VQA. As shown in Table. 4, all models (even two closed-source models) perform poorly on procedural error detection and classification compared with humans, indicating substantial room for improvement. The highest accuracy, 66.4% (EgoGPT, 8-frame input), reflects only moderate overall correctness. The highest precision, 14.5% (Qwen3-VL-32B-Instruct, 16-frame input), indicates that current models struggle to accurately classify procedural errors and identify error types. The highest recall, 97.4% (Qwen2.5-VL-7b-Instruct, 24-frame input), shows that most errors are detected, but mainly at the level of recognizing deviations from correct procedures rather than providing fine-grained error categorization. The best F1 score, 20.7% (Qwen3-VL-32B-Instruct), further confirms that all models perform inadequately on this task. Meanwhile, human performance reached 85.9%, 82.6%, 65.5%, and 73.1% on these metrics, respectively. In addition, no single metric adequately reflects model performance: relatively high accuracy may arise from class imbalance (e.g., predicting all samples as correct), while high recall alone only signifies error detection, and low precision reveals limited explanatory capability for the detected errors. Even if the model's recall exceeds that of humans, markedly lower performance on other metrics merely indicates a tendency to over-label samples as errors under the given prompts, rather than a genuine understanding of procedural errors and error types.

Across both evaluation tasks, current VLMs and agents exhibit limited egocentric perspective understanding, hindering complex procedures such as procedural error detection. Under identical inputs, agents show systematic biases: higher accu-

racy often coincides with lower recall, reflecting a tendency to label most samples as either correct or erroneous. Only large-parameter models like Qwen3-VL-32B-Instruct attain more balanced performance and thus higher F1 scores. Persistently low precision and weak Avg-Sim. further indicate that agents lack strong explanatory capacity for their predictions.

6.3 EgoError-CoT Results

As shown in Fig. 6, we adopt Precision, the metric most aligned with our task objectives, and F1, which reflects overall performance, as evaluation metrics. Introducing visual examples increases Precision from 6.4 to 10.5, F1 from 5.7 to 10.5, indicating clear performance gains. However, the agent still lacks robust error-category recognition, likely because procedural errors are intrinsically difficult to model: video clips contain substantial irrelevant information that must be filtered, and even with examples, such errors are not fully captured. In addition, the task includes multiple error types, and each example illustrates only one type, which may further confuse the agent's judgment.

Under the zero-shot setting, we tested three CoT variants. Precision increases to 7.7 and 8.3 with Check-list and No-distraction CoT prompts, respectively, but drops to 4 with Two-stage CoT. Analysis of the raw outputs shows that, with Check-list and Two-stage reasoning, the model tends to bias toward specific error types mentioned in the CoT and to select them when uncertain, thereby degrading its judgment. In contrast, No-distraction CoT is designed to avoid such misleading cues and yields the best performance gains. Appendix. E results further show that the model guided by No-distraction CoT achieves the highest TP value, i.e., the most correctly identified error types. Finally, we evaluate the few-shot setting combined with No-distraction CoT, which gives the highest Precision. Compared with zero-shot, the F1 score increases to 14.5, indicating a marked improvement.

In summary, diverse prompts and Chain-of-Thought (CoT) strategies can partially improve the agent's ability to interpret egocentric procedural errors. Although this remains insufficient for direct deployment, combining prompt engineering with in-context learning offers an efficient means to rapidly adapt the agent to new scenarios and tasks.

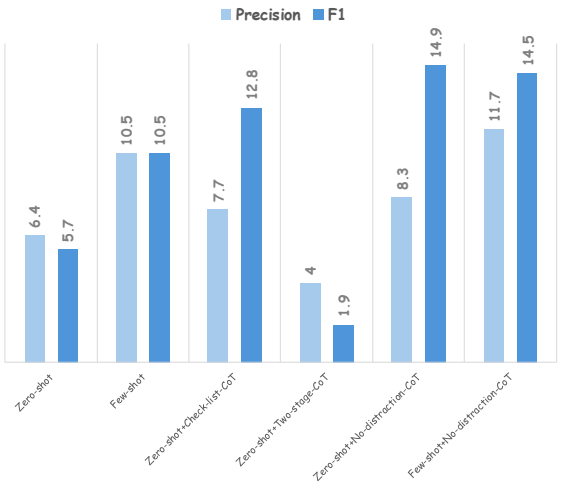


Figure 6: Figure shows the performance improvement of Vinci under the EgoError-CoT framework, mainly presented in terms of Precision and F1-score.

6.4 Agreements between Human and Evaluators

To assess the reliability of our LLM-as-a-Judge strategy, three human annotators and two judge models conducted the same scoring task. The Cohen's Kappa among the annotators was 0.711, indicating strong inter-rater agreement. Human scores were then aggregated via majority voting to obtain a consensus label. The Pearson correlation between this consensus score and the judge scores was 0.851 for Qwen and 0.781 for DeepSeek, suggesting close alignment with human judgment and supporting the reliability of the judge models. Further experimental details are provided in Appendix. D.

7 Conclusion

To thoroughly evaluate the egocentric perspective understanding of agents and VLMs, we propose EgoErrorVQA, a novel benchmark featuring a new VQA dataset. Focusing on procedural error detection tasks that require in-depth reasoning, we design both open-end and multiple-choice evaluations to comprehensively assess agents' understanding of procedural errors. We further implement EgoErrorVQA as an evaluator agent, enabling automated, dialogue-based assessment. The evaluation results suggest that agents now are still struggle with this task. Therefore, we propose EgoError-CoT, a training-free in-context learning framework that provides a promising direction for improving agents' performance on procedural tasks. We hope our work offers valuable insights for the relevant research communities.

Limitations

Our dataset covers a wide range of scenarios and tasks but has several limitations. First, the four source datasets differ in size, so their contributions to the benchmark are not exactly equal. Meanwhile, the present work only involves evaluation data and does not include training data. Anticipating future training set expansion, we ensured that each dataset retains sufficient samples for extension. Because the original datasets contain many more correct than incorrect samples, the benchmark maintains a similar ratio, with correct samples dominating. This reduces score differentiation among models in open-end VQA. In multiple-choice VQA, accuracy and related metrics should therefore not be interpreted in isolation and require cautious analysis. Also, during detailed data annotation and experiments, we observed that the video dataset contains instances requiring fine-grained action and object recognition, such as choosing an incorrect heating time by pressing the small button on the microwave. Although these cases occur in only a small fraction of clips, they can still affect the model's ability to classify error types. In EgoError-CoT, we only conduct evaluations on multiple-choice VQA because this setting uses general quantitative metrics to clearly demonstrate the performance gains brought by our method, and avoids misjudgment caused by the inherent creativity of answers in open-end VQA.

Ethical Considerations

We build our VQA dataset on publicly available egocentric datasets: CaptainCook4D, EgoOops, Epic-Tent, and Assembly101, all licensed for research use and all comply with ethical standards. We further verify that the constructed dataset contains no violent, illicit, or otherwise harmful content and does not disclose any private information. Besides, the annotators (all students) involved in this work are all listed among the authors and have been compensated appropriately.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Siddhant Bansal, Chetan Arora, and CV Jawahar. 2022. My view is the best view: Procedure learning from egocentric videos. In *European Conference on Computer Vision*, pages 657–675. Springer.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023a. Theoremqa: A theorem-driven question answering dataset. *arXiv preprint arXiv:2305.12524*.

Yi Chen, Yuying Ge, Yixiao Ge, Mingyu Ding, Bohao Li, Rui Wang, Ruifeng Xu, Ying Shan, and Xihui Liu. 2023b. Egoplan-bench: Benchmarking egocentric embodied planning with multimodal large language models. *CoRR*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198.

Sijie Cheng, Kechen Fang, Yangyang Yu, Sicheng Zhou, Bohao Li, Ye Tian, Tingguang Li, Lei Han, and Yang Liu. 2024a. Videgothink: Assessing egocentric video understanding capabilities for embodied ai. *arXiv preprint arXiv:2410.11623*.

Sijie Cheng, Zhicheng Guo, Jingwen Wu, Kechen Fang, Peng Li, Huaping Liu, and Yang Liu. 2024b. Egothink: Evaluating first-person perspective thinking capability of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14291–14302.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and 1 others. 2024c. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, and 1 others. 2024. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 1107–1128.

Chenyou Fan. 2019. Egovqa-an egocentric video question answering benchmark dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.

Alessandro Flaborea, Guido Maria D'Amely Di Melendugno, Leonardo Plini, Luca Scofano, Edoardo De Matteis, Antonino Furnari, Giovanni Maria Farinella, and Fabio Galasso. 2024. Prego: online mistake detection in procedural egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18483–18492.

Pascale Fung, Yoram Bachrach, Asli Celikyilmaz, Kamalika Chaudhuri, Delong Chen, Willy Chung, Emmanuel Dupoux, Hongyu Gong, Hervé Jégou, Alessandro Lazaric, and 1 others. 2025. Embodied ai agents: Modeling the world. *arXiv preprint arXiv:2506.22355*.

Simon Ging, María A Bravo, and Thomas Brox. 2024. Open-ended vqa benchmarking of vision-language models by exploiting classification datasets and their semantic hierarchy. *arXiv preprint arXiv:2402.07270*.

Yuto Haneji, Taichi Nishimura, Hirotaka Kameko, Keisuke Shirai, Tomoya Yoshida, Keiya Kajimura, Koki Yamamoto, Taiyu Cui, Tomohiro Nishimoto, and Shinsuke Mori. 2025. Egooops: A dataset for mistake action detection from egocentric videos referring to procedural texts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2690–2700.

Kimihiro Hasegawa, Wiradee Imrattanatrai, Zhi-Qi Cheng, Masaki Asada, Susan Holm, Yuran Wang, Ken Fukuda, and Teruko Mitamura. 2025. Promqa: Question answering dataset for multimodal procedural activity understanding. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11598–11617.

Wei-Jin Huang, Yuan-Ming Li, Zhi-Wei Xia, Yu-Ming Tang, Kun-Yu Lin, Jian-Fang Hu, and Wei-Shi Zheng. 2025. Modeling multiple normal action representations for error detection in procedural tasks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27794–27804.

Yifei Huang, Jilan Xu, Baoqi Pei, Yuping He, Guo Chen, Lijin Yang, Xinyuan Chen, Yaohui Wang, Zheng Nie, Jinyao Liu, and 1 others. 2024. Vinci: A real-time embodied smart assistant based on egocentric vision-language model. *arXiv preprint arXiv:2412.21080*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Youngkyoon Jang, Brian Sullivan, Casimir Ludwig, Iain Gilchrist, Dima Damen, and Walterio Mayol-Cuevas. 2019. Epic-tent: An egocentric video dataset for camping tent assembly. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.

Baoxiong Jia, Ting Lei, Song-Chun Zhu, and Siyuan Huang. 2022. Egotaskqa: Understanding human tasks in egocentric videos. *Advances in Neural Information Processing Systems*, 35:3343–3360.

Shih-Po Lee, Zijia Lu, Zekun Zhang, Minh Hoai, and Ehsan Elhamifar. 2024. Error detection in egocentric procedural task videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18655–18666.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, and 1 others. 2025a. From generation to judgment: Opportunities and challenges of llm-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791.

Junlong Li, Huaiyuan Xu, Sijie Cheng, Kejun Wu, Kim-Hui Yap, Lap-Pui Chau, and Yi Wang. 2025b. Building egocentric procedural ai assistant: Methods, benchmarks, and challenges. *arXiv preprint arXiv:2511.13261*.

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

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. 2025. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, and 1 others. 2024. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16488–16498.

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

Rohith Peddi, Shivvrat Arya, Bharath Challa, Likhitha Pallapothula, Akshay Vyas, Bhavya Gouripeddi, Qifan Zhang, Jikai Wang, Vasundhara Komaragiri, Eric Ragan, and 1 others. 2024. Captaincook4d: A dataset for understanding errors in procedural activities. *Advances in Neural Information Processing Systems*, 37:135626–135679.

Chiara Plizzari, Gabriele Goletto, Antonino Furnari, Siddhant Bansal, Francesco Ragusa, Giovanni Maria Farinella, Dima Damen, and Tatiana Tommasi. 2024. An outlook into the future of egocentric vision. *International Journal of Computer Vision*, 132(11):4880–4936.

Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. 2022. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106.

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

Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, and 1 others. 2023. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20270–20281.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Benita Wong, Joya Chen, You Wu, Stan Weixian Lei, Dongxing Mao, Difei Gao, and Mike Zheng Shou. 2022. Assistq: Affordance-centric question-driven task completion for egocentric assistant. In *European Conference on Computer Vision*, pages 485–501. Springer.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jingkang Yang, Shuai Liu, Hongming Guo, Yuhao Dong, Xiamengwei Zhang, Sicheng Zhang, Pengyun Wang, Zitang Zhou, Binzhu Xie, Ziyue Wang, and 1 others. 2025b. Egolife: Towards egocentric life assistant. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28885–28900.

Sheng Zhou, Junbin Xiao, Qingyun Li, Yicong Li, Xun Yang, Dan Guo, Meng Wang, Tat-Seng Chua, and Angela Yao. 2025. Egotextvqa: Towards egocentric scene-text aware video question answering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3363–3373.

Appendix

A Details of Experiment and Human Performance Evaluation

Assembly101 provides four egocentric camera views, some of which are incomplete. For each video, we therefore use the e3 view by default and resort to e4 only where e3 is missing, as e3 and e4 together cover all required information.

To evaluate human performance in open-end VQA, we randomly sampled 75 QA-pairs. Human annotators completed the same tasks as the model, were provided with identical information, viewed the same video clips before answering the questions, and human responses were scored using the same rubric eventually. When evaluating human performance on multiple-choice VQA, we randomly sample 100 samples, assign them the same tasks as the model, and compute the corresponding metrics.

The evaluation results of open-end VQA and multiple-choice VQA are presented more intuitively in Table. 5, Fig. 8 and Fig. 9.

B Details of Error Type Classification

Existing datasets with procedural errors typically define error labels in a scene-specific manner and limit them to the error types observed in that dataset. This heterogeneity in error taxonomies impedes the generalization of downstream methods to new scenarios and complicates the comprehensive evaluation of agents' understanding of procedural errors. For our VQA task, we therefore require a systematic error taxonomy that can cover all potential error types across the selected scenarios. To this end, we exhaustively analyzed all samples from the four chosen datasets and derived a unified error taxonomy that subsumes all identified error types, thereby providing the basis for constructing our multiple-choice VQA evaluation.

Specifically, the detailed definitions of the eight error types are as follows:

• **Wrong Object**: The operator uses an incorrect tool, material, or component, misuses equipment, or performs incorrect preparation of materials before a step.

• **Wrong Action**: The operator performs the correct step in an incorrect manner, works in an inappropriate way or position, or makes measurement errors, uses the wrong temperature or cooking time in culinary tasks, or exhibits motor errors when

pitching a tent.

• **Wrong Order**: The operator executes a step before or after its correct position in the sequence. If an action becomes incorrect because a preceding step was already out of order, it is still considered a Wrong Order error due to the propagated ordering mistake.

• **Omission**: The operator entirely skips a necessary step.

• **Unintended and Unnecessary Action**: The operator performs an extra step that is not part of the procedure, such as searching for an item while pitching a tent or executing an action that should not occur.

• **Correct Wrong Action**: The operator recognizes a prior mistake and actively corrects it. This is acceptable behavior but is explicitly annotated as a correction event.

• **Equipment Failure**: A tool or material fails or malfunctions during the task (e.g., during tent pitching), even when this is not caused by the operator's error.

• **Others**: Any error that does not fit the above categories, such as abnormally slow movements.

Examples of each Error Type are shown in Fig. 7.

C Details of Multiple-choice VQA Metric

The formula definitions of the four evaluation metrics are as follows:

$$\textbf{Precision} = \frac{TP}{TP + FP}, \tag{1}$$

$$\textbf{Recall} = \frac{TP}{TP + FN}, \tag{2}$$

where TP represents the number of samples which model predict they are error, GT are also error, and the error type are match, FP represents the number of samples which model predict are error, but GT are correct or the error type are mismatch.

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3}$$

where TN represents the number of samples which model predict they are correct, GT are also correct. FN represents the number of samples which the model predict they are correct, but GT are some type of error.

$$\textbf{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{4}$$

Also, in simpler terms about confusion matrix, TN: Predict is correct, GT is correct, FP: Predict is error, but GT is correct or the error type is mismatch, FN: Predict is correct, but GT is some type of error, TP: Predict is error, GT is error, and the error type is match.

D Details for Caculate Cohen's Kappa, Pearson and Spearman

When computing the Cohen's Kappa and Pearson coefficients, three human annotators scored the same 170 responses using the same criteria applied to Qwen2.5-7B-Instruct and DeepSeek-LLM-7B-Chat. These 170 responses, randomly sampled and representative, covered outputs from both models across four dataset scenarios. As shown in the table, Cohen's Kappa was first computed pairwise between annotators and then aggregated across all three to obtain an overall inter-annotator agreement. This coefficient reflects substantial consensus among annotators, indicating a broadly representative human scoring standard rather than strong individual bias.

Based on the three annotators' scores, we then applied a majority-voting scheme: if at least two annotators assigned the same score to a response, that score was taken as the final human rating; if all three scores differed, the annotators discussed the case and reached a consensus score. This procedure yielded a unified human scoring standard. Finally, we computed Pearson correlation coefficients between this unified human rating and the scores produced by Qwen2.5-7B-Instruct and DeepSeek-LLM-7B-Chat, respectively. The resulting coefficients show strong agreement between model scores and the unified human standard, supporting the reliability of the model-based ratings. You can refer to Fig. 10 and Table. 6 for detailed visualization results.

E Details for EgoError-CoT

In the CoT setup, we designed three strategies, including Check-list, Two-stage, and No-distraction. You can find the detailed experimental data in Table. 7.

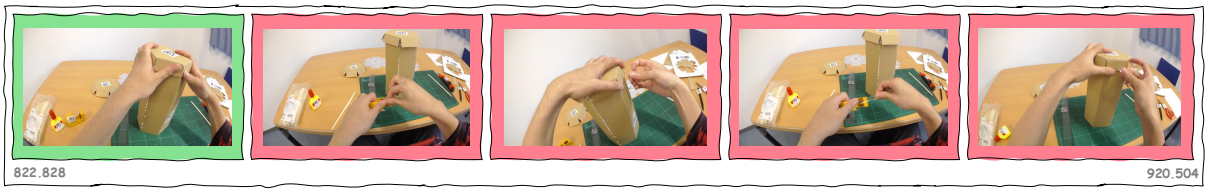
The checklist directs the model to sequentially examine the Action, Order, and Object in each segment and provides hints on potential errors. The Check-list Prompt text is as follows:

```
f"Analyze this step using the following strict checklist. Treat each question as a yes/no decision.\n\n"
f"Step 1 - Action Check:\n"
```

Correct

Wrong order: Chop 1 tsp cilantro should perform after adding tomatoes to the pan

Wrong action: Spilling milk while filling the milk



Wrong object: Use tape but should use glue

Unintended and unnecessary action: Grasp a yellow highlighter and and release it without using

Equipment failure: Guyline broke when placing it

Correct wrong action: Correct a prior error 'insert support tab'

Figure 7: Examples of each Error Type in four selected datasets.

Qwen3-VL-8B-Instruct	Video-LLaVA	EgoGPT	Vinci
Q: What mistake did I make when using the switch? A: You connected switch S2 and lamp L1 in parallel and then wired S2 into the series circuit of the motor and battery box... GT: I used switch S1 instead of switch S2, which is incorrect based on the instructions. Qwen: 2 Qwen Reason: The answer describes a different mistake than the one provided in the ground truth. It suggests using S1 instead of S2, which is not mentioned in the model's answer. Deepseek: 4 Deepseek Reason: The model's answer is mostly correct, but it could be more clear and concise. It provides a detailed explanation of the mistake, but it could be more concise for a general audience.	Q: Did I use the wrong part when I attach track to chassis? A: Yes, you did. You used a wheel instead of a track. GT: No, I used the correct part. Qwen: 1 Qwen Reason: The model's answer is completely incorrect as it states that a wheel was used instead of a track, while the ground truth indicates that the correct part was used. Deepseek: 2 Deepseek Reason: The model's answer is partially correct. It correctly identifies that a wheel was used instead of a track, but it also contains a minor error by stating that the wheel was used instead of a track. The ground truth is that the correct part was used, so the model's answer is mostly wrong.	Q: What went wrong when I tried to detach the bumper from the body? A:The screw was not properly loosened, making it difficult to remove. GT: This action was unnecessary because the bumper was already properly attached to the body. Qwen: 1 Qwen Reason: The model's answer does not match the ground truth. The ground truth states that detaching the bumper was unnecessary, while the model suggests a specific technical issue with the screw. Deepseek: 3 Deepseek Reason: The model's answer is partially correct. It correctly identifies the issue with the screw, but it does not fully address the ground truth that the bumper was already properly attached to the body.	Q: Did I discard both ends of the tortilla without any leftovers? A:You did discard both ends of the tortilla without any leftovers. GT: Yes, I discarded both ends of the tortilla completely. Qwen: 5 Qwen Reason: The model's answer is a perfect match to the ground truth. Deepseek: 4 Deepseek Reason: The model's answer is mostly correct, but it lacks the specific detail of discarding both ends of the tortilla completely. It could be interpreted as partially correct or mostly correct.

Table 5: Performances on Open-end VQA is presented along with its scoring results.

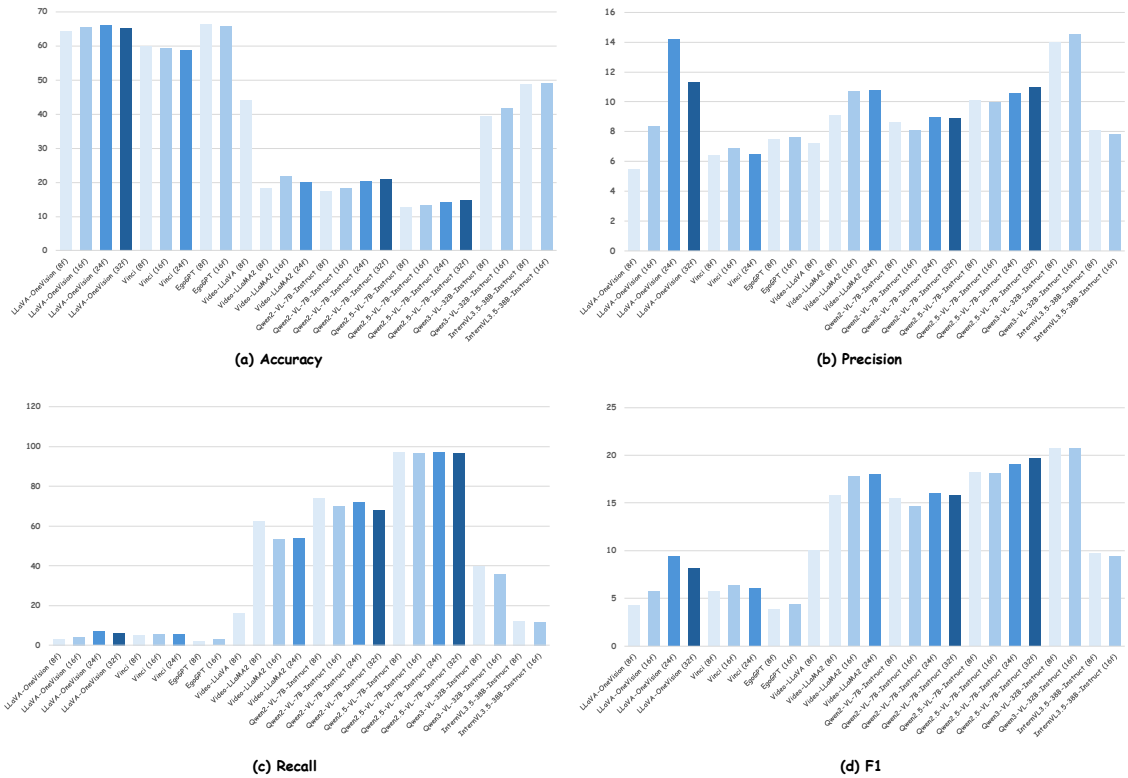


Figure 8: Performance comparison among different agents and models under four metrics in Multiple-choice VQA evaluation.

Coefficient	A vs B	A vs C	B vs C	All
Cohen's Kappa	0.716	0.771	0.646	0.711
Fleiss' Kappa	-	-	-	0.710
Pearson	0.950	0.959	0.934	-

Table 6: Results of three coefficients calculated among annotators A, B, and C. Fleiss' Kappa represents the overall agreement of the three annotators' ratings.

```
  f" - Is the action performed with incorrect technique, measurement, temperature, timing, or force?\n"
  f" - Is this action merely correcting an earlier mistake (and thus not an error itself)?\n"
  f" → If execution is flawed and not a correction, it may be 'Wrong action'.\n\n"
  f"Step 2 - Order Check:\n" f" - Is this step happening at the correct point in the procedure?\n" f" - Has this step been completely omitted (i.e., never performed when it should be)?\n"
  f" → If out of sequence or missing, it may be 'Wrong order' or 'Omission'.\n\n"
  f"Step 3 - Object Check:\n"
  f" - Are the objects, tools, or materials used in the video consistent with what the step requires?\n"
  f" → If a wrong object is used, it may be 'Wrong object'.\n\n"
  f"Step 4 - Final Decision:\n"
  f" - If all checks are 'No' (no issues), the answer is 'correct'.\n"
  f" - If any check is 'Yes', select the SINGLE most applicable error type based on procedure guidelines.\n\n"
  f"Output Format:\n"
  f"Output ONLY the final error type name. No reasoning, no punctuation, no extra text."
```

In the two-stage strategy, the model first determines whether an error exists based on predefined error-type definitions. If no error is detected, it responds "correct"; otherwise, it identifies the specific error type in a fixed sequence. The Two-stage Prompt text is as follows:

```
  f"Perform a two-stage reasoning process:\n\n"
  f"Stage 1 - Is there any procedural error in this step?\n"
  f" - Evaluate based on clear definitions of error types in PROCEDURAL_ERROR_GUIDANCE.\n"
  f" - If NO error is present, immediately conclude with 'correct' and do not proceed further.\n\n"
  f"Stage 2 - If an error exists, determine its primary type:\n"
  f" - Does the error stem from HOW the action is performed? → Likely 'Wrong action'.\n"
  f" - Does the error stem from WHEN the step occurs or if it's missing? → Likely 'Wrong order' or 'Omission'.\n"
  f" - Does the error stem from USING THE WRONG OBJECT? → Likely 'Wrong object'.\n"
  f" → Choose the SINGLE dominant error type that best explains the observed issue.\n\n"
  f"Output Format:\n"
  f"Output ONLY the final error type name. If no error, output 'correct'. Nothing else."
```

The two aforementioned methods introduce explicit error-type terminology, which may interfere with and mislead the model. To mitigate this, we propose a No-distraction CoT strategy. In this setting, the model first describes what actually occurred, then infers what should have occurred according to the prescribed procedure, and finally compares the two to reach a judgment. Throughout this process, no explicit error-type terminology is used. Prompt text is as follows:

```
  f"Please reason step by step based solely on what is observed in the video and the expected procedure:\n\n"
  f"Step 1 - What is actually happening?\n"
  f" - Describe precisely how the action is performed (e.g., hand motion, tool usage, object interaction).\n"
  f" - List the objects involved and the timing relative to other steps.\n\n"
  f"Step 2 - What should be happening?\n"
  f" - According to the full task procedure, what is the correct way to perform this step?\n"
  f" - What objects should be used, and at what point in the sequence?\n\n"
  f"Step 3 - Compare reality vs. expectation:\n"
  f" - Are there any discrepancies in how, when, or with what the step is carried out?\n"
  f" - If everything matches the expected behavior, there is no error.\n\n"
  f"Step 4 - Final judgment:\n"
  f" - If no discrepancy exists, output 'correct'.\n"
  f" - If a discrepancy exists, determine which single error category (as defined in your guidelines) best captures the core issue.\n\n"
  f"Output Format:\n"
  f"Output ONLY the final error type name. No explanation, no intermediate text."
```

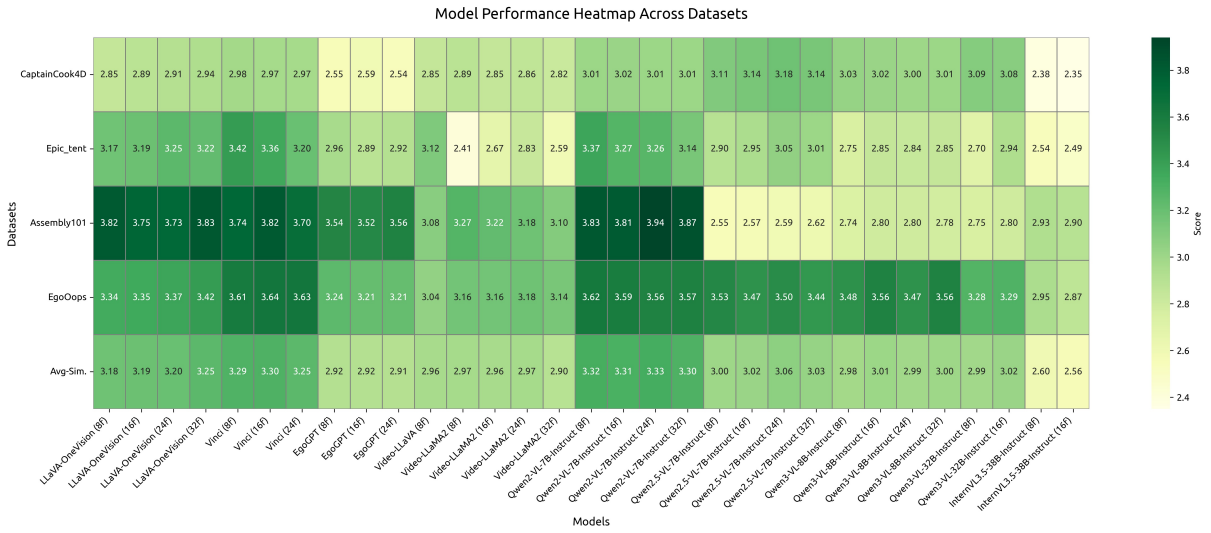


Figure 9: A heatmap of open-end VQA evaluation results.

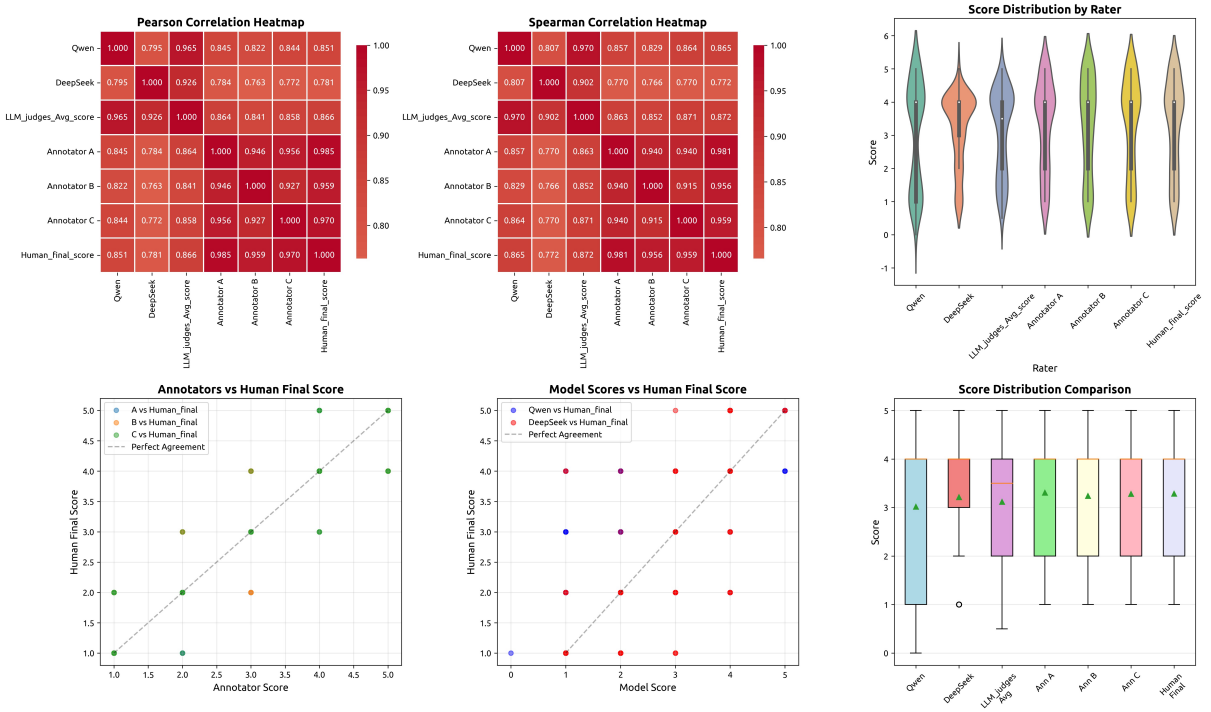


Figure 10: Heatmaps of the Pearson and Spearman correlations, intuitively illustrating the inter-rater relationships. Spearman correlation is a non-parametric statistic that quantifies the strength and direction of a monotonic association between two variables. In addition, it shows the score distribution for each rater, where **Human Final Score** denotes the final score representing the human evaluation standard obtained via the voting mechanism described above, and **LLM Judges Avg Score** denotes the average score of the two judge models.

Method	TP	FP	TN	FN	Accuracy	Precision	Recall	F1
Vinci (zero-shot)	22	321	1093	407	60.0	6.4	5.1	5.7
Vinci (Few-shot)	44	374	1046	379	58.7	10.53	10.4	10.5
Vinci (Zero-shot+Check-list-CoT)	95	1140	452	156	29.5	7.7	37.9	12.8
Vinci (Zero-shot+Two-stage-CoT)	6	144	1221	472	66.1	4.0	1.3	1.9
Vinci (Zero-shot+No-distraction-CoT)	134	1490	183	36	17.1	8.3	78.8	14.9
Vinci (Few-shot+No-distraction-CoT)	76	573	871	323	31.7	11.7	19.1	14.5

Table 7: Table shows how, within the EgoError-CoT framework, zero-shot and few-shot settings, together with different types of CoT, lead to different effects on the results.