# Generate, Verify, and Refine: Event Timelines for Robust Video-Grounded Dialogue Systems

**Wiradee Imrattanatrai[1], Masaki Asada[1], Kimihiro Hasegawa[2], Susan Holm[2], Ken Fukuda[1], Teruko Mitamura[2]**
[1]National Institute of Advanced Industrial Science and Technology (AIST),
[2]Language Technologies Institute, Carnegie Mellon University
**Correspondence:** wiradee.imrattanatrai@aist.go.jp

## Abstract

Video-grounded dialogue (VGD) requires models to generate coherent responses by tracking and reasoning over temporally evolving events in video content. However, existing approaches offer limited ability to capture temporal dependencies, particularly in complex scenarios. In this paper, we present a framework that integrates structured event timelines with verification-aware reasoning to improve temporal consistency in VGD. We represent video content as structured states and temporally ordered actions, enabling models to both infer event timelines from video segments and leverage them to guide response generation. In addition, to address the inherent noise in predicted structure timelines, we introduce a Generate–Verify–Refine (GVR) mechanism that explicitly evaluates and revises unreliable event predictions, thereby reducing the propagation of upstream errors. Experiments on the VDAct benchmark show that our approach consistently improves dialogue quality across models, achieving gains of up to +13.0 LAVE and accuracy over unstructured grounding. Human evaluation further demonstrates improvements of 20.7–31.9 points in response correctness. Additional analysis shows that while better event predictions correlate with stronger dialogue performance, naive reliance on structured outputs can degrade results, highlighting the need for verification-aware reasoning. Our implementation of GVR on event timelines for VGD is available at https://github.com/aistairc/GVR-VGD.

## 1 Introduction

Video-based interactive applications are driving the growing demand for conversational systems capable of video-grounded dialogue (VGD), in which models engage in multi-turn interactions grounded in dynamic visual content. Such systems are crucial for real-world settings, such as in-home monitoring, assistive agents, and activity understanding. While datasets such as AVSD (Alamri et al., 2019) and VDAct (Imrattanatrai et al., 2025) have enabled progress, and recent Vision-Language Models (VLMs) (Li et al., 2024a; Chen et al., 2024; Bai et al., 2025; Xiaomi, 2025) have substantially advanced video understanding, building reliable VGD systems remains challenging. To answer dialogue questions accurately, a system must reason over temporally evolving events by tracking actions, state transitions, and their order, while maintaining consistent grounding across multiple turns. Existing approaches commonly rely either on only raw visual context (Schwartz et al., 2019; Liu et al., 2024a; Li et al., 2022; Ye et al., 2025) or with unstructured grounding like dense captions (Kim et al., 2020; Mogrovejo and Solorio, 2024; Tang et al., 2021). However, raw visual features do not explicitly expose event-level structure, requiring the model to infer temporal dependencies implicitly. Dense captions partially alleviate this issue, but they are often redundant and difficult to use effectively for long-horizon reasoning as they can quickly exceed the model's effective context window (Liu et al., 2024b). As videos become more event-centric and temporally complex, models may lose track of what has happened, in what order, and which details remain relevant to the current dialogue turn, leading to inconsistent or incorrect responses. In contrast, prior work in video understanding has explored structured representations such as scene graphs (Li et al., 2024b; Im et al., 2024; Nguyen et al., 2025) and event knowledge graphs (Egami et al., 2021, 2023). While expressive, these formats are often overly complex and not well aligned with VGD, where generation must be conditioned on a concise, temporally coherent summary of visual evidence. We therefore argue that an effective intermediate representation for VGD should be compact, temporally ordered, and directly usable for downstream reasoning.

In this work, we bridge video understanding and



*Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 49–68
November 6-10, 2026 ©2026 Association for Computational Linguistics

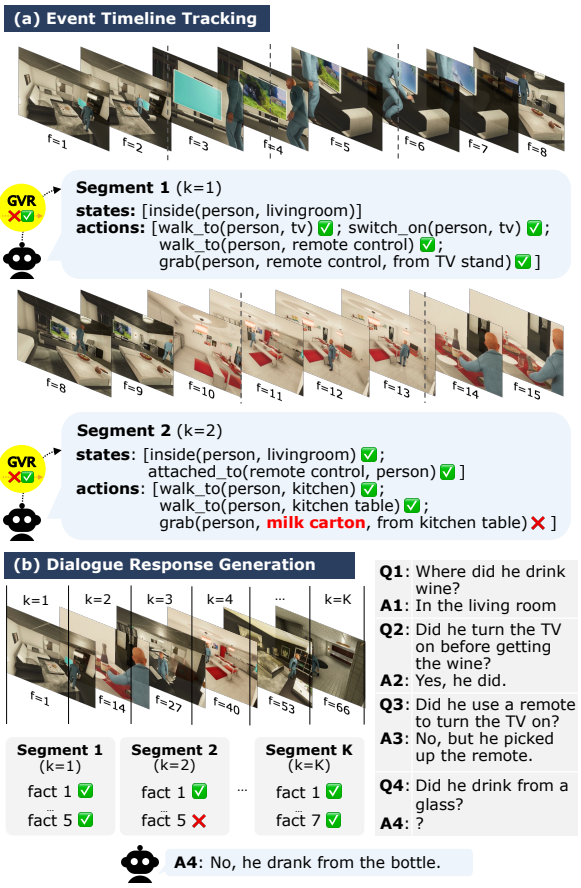



Figure 1: Overview of the proposed two-stage framework. First, the model tracks an event timeline, including states and temporally ordered actions, for each video segment. The tracking operates within the proposed iterative Generate–Verify–Refine (GVR) loop, in which each fact in the event timeline is verified and refined based on its reliability. Second, the predicted timeline is used as a structured grounding context, alongside the dialogue history, to generate dialogue responses.

dialogue response generation through *structured event timelines*. Each video segment is represented by states and temporally ordered actions, providing a compact summary of events. We then cast VGD as a joint event timeline tracking and timeline-conditioned response generation task, which requires the model not only to infer event timelines from video segments but also to use them as evidence for response generation. Moreover, event timeline tracking may produce imperfect predictions, including incorrect spatial relations, misidentified actions, or temporal ordering errors. Directly conditioning on one-pass timelines without fact verification can pass these errors to downstream response generation. Thus, we introduce a *Generate–Verify–Refine* (GVR) mechanism that enables the model to assess the reliability of predicted events through an iterative loop of verification and refinement. To enable the model to exhibit this behavior, we design a training pipeline that mirrors the inference process by injecting synthetic corruptions into both event tracking and response generation samples. This exposes the model to imperfect evidence during training, enabling it to detect incorrect facts, refine unreliable timelines, and produce more robust responses in the presence of noisy predictions. Our experiments on the VDAct dataset show that our approach achieves consistent gains over baselines across multiple VLM backbones in both automatic metrics and human evaluation. Our analysis further reveals a trade-off between event tracking accuracy and downstream response generation performance, highlighting that although structured representations provide a strong inductive bias for temporal reasoning, their effective use also requires explicit handling of prediction uncertainty.

## 2 Methodology

**Overview and Task Formulation** The VGD task aims to generate accurate responses that are grounded in both the video content and the dialogue history. In this paper, we focus on videos depicting multiple household activities performed by a human agent. Each video consists of raw frames, from which $N$ frames are uniformly sampled as a sequence $\mathcal{V}=\{f_1, f_2, \ldots, f_N\}$. In terms of the dialogues, we let the dialogue history up to the current user question at turn $t$ be denoted as $\mathcal{D}=\{u_1, r_1, u_2, r_2, \ldots, u_t\}$, where $u$ and $r$ represent users' questions and system responses, respectively. The main objective for the task is to generate a response $r_t$ that is consistent with both the dialogue context $\mathcal{D}$ and the visual content in $\mathcal{V}$.

Our approach follows a two-stage framework (Figure 1): (1) event timeline tracking and (2) dialogue response generation. Given a video, we partition it into overlapping segments, $\{S_1, S_2, \ldots, S_K\}$, each containing $M$ frames with a stride of $M$-1, such that consecutive segments share a single frame. The first stage processes each video segment $S_k$ independently, where it predicts an event timeline $E_k$ that captures the initial spatial states and subsequent actions via the proposed GVR mechanism (Figure 2). The resulting timelines $\mathcal{E}=\{E_1, E_2, \ldots, E_K\}$ provide an interpretable representation of event progression throughout the video. In the second stage, the predicted timelines $\mathcal{E}$ are combined with the corre-

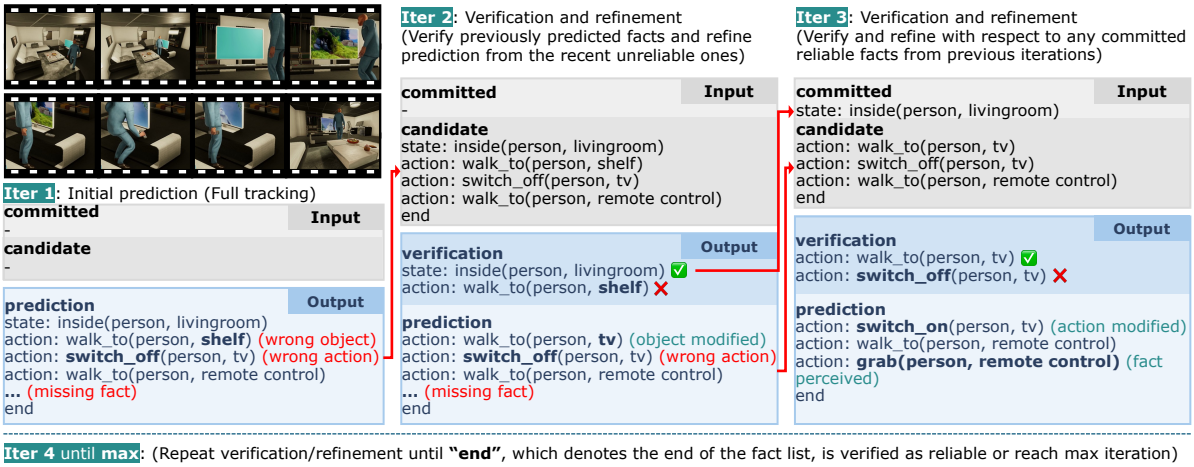



Figure 2: Our Generate–Verify–Refine (GVR) mechanism iteratively verifies and refines predicted events to reduce error propagation to downstream reasoning. Initially, GVR predicts a complete event timeline as a list of candidate facts by observing visual evidence presented in the given video frames. At the next iteration, the model is given the recently predicted candidate facts to validate whether each candidate fact is actually supported by the visual evidence and consistent with the committed facts, assigning a true or false label accordingly. This verification stops at the first candidate fact deemed false (*i.e.,* `walk_to(person, shelf)`). During refinement (*i.e.,* re-prediction), the model regenerates the remaining event timeline starting from the most recent verified fact (*i.e.,* `inside(person, livingroom)`). Consequently, the replacement `walk_to(person, tv)` is newly inferred, alongside other new candidate facts. This iterative process continues to the next iteration until the "end" of the timeline is verified or a maximum number of iterations is reached.

sponding dialogue context $\mathcal{D}$ and visual context $\mathcal{V}$ to form a grounded input, from which the model generates a response to the user's query $u_t$.

**Event Timeline Representation** To support interpretable reasoning, we represent each video segment with a compact, structured event timeline that captures discrete facts about: (1) **States**, which encode initial spatial and possession relations in the form of `inside(subject, object)`, `next_to(subject, object)`, or `attached_to(object, subject)`, and (2) **Actions**, which describe temporally ordered actions in the form of `relation(subject, object [optional], prepositional phrase [optional])`. We adopt a relation-first format for both states and actions to prioritize the evolving aspects of the scene (*e.g.,* location, interaction, possession) over the constant subject, and to reduce the repetitiveness in the prediction.

**Event Timeline Tracking** To construct event timelines, we employ a VLM to map a video segment $S_k$ into a structured event timeline $E_k$. The model is guided to emphasize human-centric activities while omitting irrelevant background details.

Due to challenges in visual perception and temporal reasoning, the model may hallucinate or produce incorrect facts in the predicted timelines. To address this, we replace single-pass prediction with a multi-step inference incorporating a verification mechanism, termed *Generate–Verify–Refine* (GVR). In the first pass, the model **generates** a candidate event timeline comprising states and a sequence of actions based on the input video segment. In subsequent iterations, it **verifies** each predicted state and action by assigning explicit correctness signals (*e.g.,* valid/invalid). Valid facts are considered reliable and added to a committed list for the next iteration, while the model **refines** the timeline by revising the unreliable facts and updating subsequent facts accordingly. This iterative process continues until the entire event timeline is verified or a maximum number of iterations is reached, allowing the model to progressively refine its predictions before producing the final timeline.

To induce effective GVR behavior in the model, we opt to train the model on the training data that explicitly mirrors the GVR process using synthetic corruptions. Starting from ground-truth timelines, we generate corrupted candidates by perturbing individual facts at the level of relation, object, or preposition to simulate common event tracking mistakes such as incorrect object grounding, misidentified interactions, or missing facts. We organize

training samples into *three* groups, where each corresponds to a different stage of the GVR process. **(1) Full Tracking:** the model predicts an event timeline from scratch given a video segment, without any prior commitments. These samples are derived directly from ground-truth timelines. **(2) Verification from Scratch:** the model is given a candidate timeline and predicts correctness labels for each fact, followed by refining incorrect ones. The candidate timelines include both clean and synthetically corrupted facts, enabling the model to learn to detect and recover from errors. **(3) Verification with Partial Commitments:** the model performs verification conditioned on previously validated facts in a committed list, requiring it to assess new candidate facts with respect to reliable prior information and refine inconsistencies accordingly.

To generate corrupted timelines for the latter two verification groups, we randomly replace ground-truth facts in the timeline with synthetic alternatives. Specifically, we prompt a model (*i.e.,* `gpt-4o-2024-08-06`) to modify one or more components of a fact with plausible but incorrect entities or actions. For example, `inside(person, kitchen)` may be altered to `inside(person, bedroom)`, and `grab(person, milk carton, from kitchen table)` to `put(person, wineglass, from kitchen table)`. This process produces realistic negative samples that reflect typical model prediction errors. To balance exposure to clean and noisy supervision, we sample a corruption ratio $c^e \in \{0.35, 0.5, 0.65\}$, where $c^e$ denotes the fraction of timelines used for verification with synthetic corrupted facts. Each ratio represents a range of moderate-to-heavy numbers of corrupted timelines.

**Dialogue Response Generation** After obtaining the event timelines, the system generates a response $r_t$ to the user's question $u_t$ conditioned on the dialogue history $\mathcal{D}$, uniformly sampled video frames $\mathcal{V}$, and corresponding structured event timeline $\mathcal{E}$.

Similar to event timeline tracking, we consider a realistic setting in which upstream timeline predictions may be imperfect. To account for this, we also introduce synthetic corruptions to the event timelines used for response generation during model training. In particular, we perturb a subset of facts in $\mathcal{E}$ using the same corruption strategy as in the tracking stage, thereby simulating noisy or partially incorrect timelines. The model is then trained to generate correct responses despite these imperfections, encouraging it to rely selectively on reliable evidence and avoid propagating erroneous facts into the final answer. For this stage, we sample a corruption ratio $c^d \in \{0.1, 0.2\}$, denoting the fraction of facts that are replaced with corrupted variants in the timelines. These ratios are intentionally lower than those used in event timeline tracking, as response generation assumes reasonably reliable upstream predictions. Excessive corruption would produce unrealistically noisy timelines and obscure the intended role of structured grounding.

**Training** We fine-tune the VLMs via multitask learning with standard cross-entropy loss, using training samples prepared using the previously explained corruption strategy, for both event timeline tracking and dialogue response generation.

## 3 Experiments

**Dataset** We adopt the VDAct dataset, a VGD dataset comprising dialogues that discuss the human agent's activities in a simulation video. The dataset contains 3,000 dialogues with over 30,000 QA pairs, split into approximately 24,000 pairs for training, 4,500 for testing, and 1,500 for validation. These dialogues are derived from 1,000 videos, each averaging 248 seconds.

**Pre-processing** We construct event timeline annotations based on VHAKG (Egami et al., 2024). This dataset provides per-frame event annotations for VirtualHome2KG simulations, which serve as the source videos for VDAct. For event timeline tracking, we prepare new videos using VHAKG activity videos. These new videos do not overlap with VDAct videos to prevent label leakage into the dialogue response generation stage. For each video, we uniformly sample frames at 1 fps. Each sampled frame is mapped to the ground-truth event using its annotated start and end frames. We segment the sampled frames into overlapping windows of size $w$. For each segment, we associate events based on the overlap between event frames and segment frames. A minimum overlap threshold is used: 10% for short events and 20% for longer events. This ensures that events are included only when they are sufficiently visible in the segment. Each segment is paired with structured ground-truth annotations and converted into our event timeline using rule-based parsing. For dialogue response generation, we use the VDAct training data and apply the same timeline-generation pipeline.

**Experimental Settings** We evaluate our framework with three open-weight VLMs, including LLaVA-OneVision-Qwen2-7B-OV (Li et al., 2024a), Qwen2.5-VL-7B-Instruct (Bai et al., 2025), and MiMo-VL-7B-RL (Xiaomi, 2025), and three proprietary models, including Anthropic's claude-sonnet-4-20250514, OpenAI's gpt-4o-2024-08-06 and gpt-5-mini-2025-08-07. We evaluate the effect of different grounding strategies with video frames, including *dialogue context only*, *dialogue with unstructured captions*, and *dialogue with event timelines (ET)*. For fine-tuned models, we consider either *template-based* or *LLM-refined* captions derived from the event KG of each video in VDAct.

**Implementation Details** We fine-tune the open-weight models for up to 15 epochs with LoRA adapters (rank 128, scaling factor $\alpha$=256). Training is performed with a global batch size of 128, a learning rate of 2e-5, and a cosine scheduler with a warm-up ratio of 0.03. For visual input, 32 frames per scenario are used, except for proprietary models (16 frames). A temporal window $w$ of 15 is applied for event timeline tracking. We independently fine-tune the model with six different combinations of corruption ratios $c^e$ and $c^d$ for GVR supervision. For each model family, the best configuration of $c^e$ and $c^d$ is selected based on SPICE (Anderson et al., 2016) on the development set, with LAVE (Mañas et al., 2024) used as a tie-breaker when SPICE scores are identical. All experiments are conducted using 8 H200 GPUs, with a random seed of 42. All reported results are from a single run.

**Evaluation Metrics** For *Event Timeline Tracking*, we employ F1-score to compare the predicted and reference sets of event facts per segment, while Exact Match and Accuracy are used to evaluate whether the entire sequence of predicted events per segment matches the ground truth. For *Dialogue Response Generation*, we use standard text generation metrics to assess the quality of responses, including BLEU (B) (Papineni et al., 2002), ROUGE (R-L) (Lin, 2004), METEOR (M) (Banerjee and Lavie, 2005), and SPICE (S) (Anderson et al., 2016). Additionally, we employ LLM-based evaluation metrics, including LLM-Accuracy (Acc.), LLM-Relevancy (Rel.), and LAVE (L) (Mañas et al., 2024), to assess factual consistency, relevance, and grounding quality.

We also conduct human evaluation of the fine-tuned models on 150 randomly sampled QA pairs. Three annotators watch the corresponding video and rate each response as fully consistent (3), partially consistent (2), or inconsistent (1). References serve only as auxiliary information; correct descriptions omitted from or phrased differently in the reference are not penalized. Across 1,800 item-system pairs, ordinal Krippendorff's $\alpha$ is 0.564. We additionally report ICC(2,3) (two-way random-effects, absolute agreement, average measures): 0.796 (95% CI: [0.778, 0.813]), indicating good-reliability range under Koo and Li (2016) guidelines. This complementary measure assesses the reliability of the mean of three ratings. Ratings are aggregated by majority vote, with a fourth annotator breaking ties, and scores are reported on a 0-100 scale.

| Model / Context | B | R-L | M | S | Acc. | Rel. | L |
|---|---|---|---|---|---|---|---|
| **LLaVA-OV** | | | | | | | |
| None | 7.7 | 30.9 | 31.3 | 27.3 | 20.3 | 31.3 | 29.7 |
| Template Captions | 6.6 | 28.4 | 28.9 | 26.0 | 18.0 | 29.3 | 26.0 |
| Refined Captions | 6.3 | 27.6 | 28.3 | 25.5 | 20.2 | 31.2 | 26.9 |
| ET | 7.6 | 31.4 | 31.1 | 29.6 | 22.7 | 34.2 | 32.3 |
| ET + GVR | **8.5** | **32.9** | **32.4** | **31.0** | **26.7** | **38.3** | **37.1** |
| **Qwen2.5-VL** | | | | | | | |
| None | 10.0 | 35.2 | 35.9 | 33.1 | 32.5 | 43.1 | 41.3 |
| Template Captions | 7.5 | 30.2 | 31.4 | 29.2 | 23.0 | 33.6 | 31.6 |
| Refined Captions | 8.9 | 33.0 | 33.7 | 31.2 | 27.3 | 38.0 | 36.1 |
| ET | **11.3** | 37.4 | 38.2 | 35.7 | 37.2 | 48.6 | 47.0 |
| ET + GVR | **11.3** | **38.1** | **38.4** | **37.1** | **40.3** | **51.0** | **49.1** |
| **MiMo-VL** | | | | | | | |
| None | 10.1 | 34.8 | 35.0 | 32.5 | 30.2 | 40.8 | 38.9 |
| Template Captions | 8.6 | 33.5 | 33.4 | 31.3 | 25.1 | 36.2 | 33.9 |
| Refined Captions | 9.6 | 34.3 | 34.8 | 31.4 | 27.6 | 38.6 | 36.9 |
| ET | **11.8** | **38.2** | **38.0** | **36.7** | 36.0 | 47.3 | 45.8 |
| ET + GVR | 11.7 | 38.0 | 37.9 | **36.7** | **37.6** | **49.1** | **48.6** |

Table 1: Performance across LoRA-finetuned models using different grounded contexts for response generation. **Bold** scores denote per-group best, while underlined results indicate the highest overall.

| Model | None | Refined Captions | ET | ET + GVR |
|---|---|---|---|---|
| LLaVA-OV | 47.3 | 28.6 | 43.3 | **49.3** |
| Qwen2.5-VL | 42.0 | 38.7 | 53.3 | **70.6** |
| MiMo-VL | 45.3 | 42.7 | 58.0 | **71.6** |

Table 2: Human evaluation of the fine-tuned models on randomly sampled 150 QA pairs. Each answer was rated on a 3-point scale by three annotators, with the ratings aggregated by majority vote; a fourth annotator was used to break ties when no majority was reached. Scores are reported on a 0-100 scale.

### 3.1 Experimental Results

**Results on Dialogue Response Generation** Table 1 presents a comparison of LoRA fine-tuned models conditioned on different types of grounding context for dialogue response generation. Across all models, the results show a consistent benefit

| Model / Context | B | R-L | M | S | Acc. | Rel. | L |
|---|---|---|---|---|---|---|---|
| **Claude Sonnet 4** | | | | | | | |
| None | **4.0** | **23.3** | **29.7** | **22.2** | 28.4 | 40.3 | 40.6 |
| Captions | 2.9 | 20.6 | 27.7 | 19.9 | 25.7 | 38.9 | 41.0 |
| ET | 3.1 | 21.1 | 28.1 | 20.4 | 26.0 | 38.6 | 40.5 |
| ET + GVR | 0.5 | 6.6 | 16.8 | 0 | 27.3 | 45.3 | 53.1 |
| ET + Fixed | 1.4 | 12.4 | 23.2 | 15.1 | 26.9 | 41.9 | 47.8 |
| ET + GVR + Fixed | 0.6 | 6.9 | 17.6 | 10.0 | **29.8** | **46.8** | **53.8** |
| **GPT-4o** | | | | | | | |
| None | **6.9** | 28.3 | 32.5 | 26.2 | 32.6 | 43.7 | 42.0 |
| Captions | 5.1 | 26.7 | 32.1 | 24.6 | 34.0 | 45.9 | 44.9 |
| ET | 6.2 | 28.3 | 33.1 | 26.2 | 33.3 | 45.0 | 44.3 |
| ET + GVR | 3.7 | 24.5 | 30.9 | 23.7 | 32.6 | 44.8 | 45.8 |
| ET + Fixed | 6.4 | **28.7** | **33.6** | **26.5** | 33.3 | 45.5 | 44.7 |
| ET + GVR + Fixed | 4.1 | 25.3 | 32.1 | 24.0 | **34.6** | **46.1** | **47.2** |
| **GPT-5-mini** (Think) | | | | | | | |
| None | **2.1** | **21.8** | 26.4 | 20.3 | 35.7 | 48.3 | 50.0 |
| Captions | 1.7 | 20.1 | 26.6 | 19.4 | 38.5 | 51.3 | 54.5 |
| ET | 1.8 | 21.0 | 26.9 | 20.5 | 35.6 | 48.9 | 51.7 |
| ET + GVR | 1.5 | 18.1 | 25.9 | 19.7 | 37.8 | 50.9 | 55.4 |
| ET + Fixed | 2.0 | 21.3 | **27.5** | **21.3** | 36.7 | 49.8 | 52.6 |
| ET + GVR + Fixed | 1.7 | 18.6 | 26.3 | 20.6 | **39.3** | **52.1** | **56.7** |

Table 3: Performance across proprietary models using different grounded contexts for response generation. **Bold** scores denote per-group best, while underlined results indicate the highest overall.

| Model | State Tracking F1 | State Tracking EM | Action Tracking F1 | Action Tracking Acc. |
|---|---|---|---|---|
| (-) GVR: | | | | |
| LLaVA-OV | 62.5 | 31.4 | 48.5 | 32.3 |
| Qwen2.5-VL | **66.2** | **37.4** | **50.4** | **33.5** |
| MiMo-VL | 62.7 | 31.2 | **50.4** | 32.4 |
| Claude Sonnet 4 | 37.9 | 2.4 | 22.6 | 7.1 |
| GPT-4o | 47.3 | 11.1 | 28.6 | 14.2 |
| GPT-5-mini | 34.9 | 1.0 | 24.8 | 6.4 |
| (+) GVR: | | | | |
| LLaVA-OV | 60.8 | 34.8 | 46.8 | 33.4 |
| Qwen2.5-VL | **68.8** | **41.1** | 57.6 | 39.3 |
| MiMo-VL | 68.1 | 39.9 | **59.2** | **40.6** |
| Claude Sonnet 4 | 38.9 | 4.6 | 18.7 | 5.9 |
| GPT-4o | 47.7 | 8.6 | 26.5 | 12.2 |
| GPT-5-mini | 43.6 | 4.1 | 26.2 | 9.2 |

Table 4: Performance for event timeline tracking across fine-tuned and proprietary models when not using and using the GVR strategy. **Bold** scores denote per-group best, while underlined indicate the highest overall.

from incorporating the structured event timelines (ET). Moreover, incorporating the proposed GVR strategy yields further gains. In particular, GVR-based event timelines (ET+GVR) improve LLaVA-OV and Qwen2.5-VL across all evaluation metrics, while MiMo-VL achieves the improvements with LLM-based evaluation scores. Overall, Qwen2.5-VL with ET+GVR performs best across most metrics, except for BLEU and ROUGE-L. The human evaluation results in Table 2 further support that GVR improves overall response accuracy.

Additionally, we report the performance of proprietary models for the response generation task as shown in Table 3. From the results, Claude Sonnet 4 is largely insensitive to added grounding captions and single-pass event timelines (ET), as its performance is better in the no-context setting, whereas GPT-4o and GPT-5-mini show modest gains on a few metrics when given event timelines, compared to when given captions. However, when incorporating GVR for all models, their performance further outperforms ET and other baseline groundings.

We conduct qualitative analysis to examine why incorporating ET without GVR does not consistently yield improvements over captions, as expected. Our analysis reveals that their performance is limited by noise in the predicted timelines as the models frequently over-predict low-salience actions (*e.g.,* `lower(person, cup)`, `walk(person, through door)`, `bend(person)`), which weakens the signal needed for the task and as a result, hurts the performance. To mitigate this, we introduce **ET+Fixed** and **ET+GVR+Fixed**, which constrain timeline tracking to a fixed set of plausible relations and objects. This encourages the model to focus on relevant states and actions. Based on the results, ET+Fixed consistently outperforms ET across all models. Furthermore, with the ET+GVR+Fixed setting, the performance on LLM-based metrics is the best across all models.

**Results on Event Timeline Tracking** Table 4 compares event timeline tracking performance with and without GVR. Among the LoRA fine-tuned models, Qwen2.5-VL achieves the best scores across all metrics without GVR. With GVR, all models improve on nearly all metrics. Notably, MiMo-VL achieves the strongest action tracking performance, while Qwen2.5-VL obtains the best state tracking performance. For LLaVA-OV, tracking performance improves on exact-match state tracking and on action tracking accuracy, while dropping on F1 for both tracking types. In addition, among proprietary models, GPT-4o achieves the strongest performance and leads across metrics, but it still falls short of the fine-tuned models. These results highlight the importance of task-specific fine-tuning to fully leverage timeline representations and the effective GVR process. These event timeline tracking results mirror the trend in downstream response generation, especially with the fine-tuned models. When predicted timelines are more accurate, they provide clearer evidence for answering questions. In particular, effective timelines emphasize salient events (*e.g.,* picking up a specific

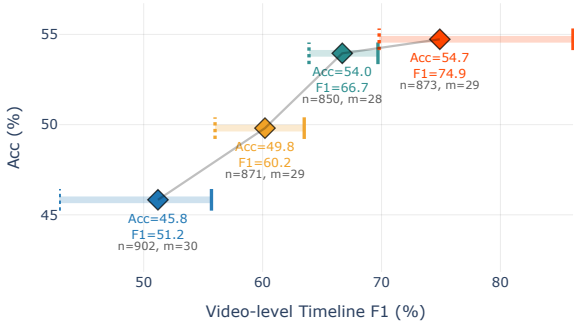



Figure 3: Relationship between predicted event timeline quality and response accuracy based on the results by the fine-tuned Qwen2.5-VL. Videos are grouped into four bins based on their associated F1 scores. For each bin, the diamond marker shows the mean F1 and the corresponding mean LLM-Acc with the number of questions ($n$) and videos ($m$).

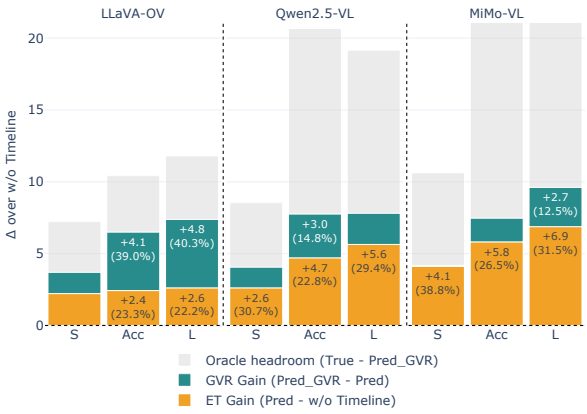



Figure 4: Performance improvement for the response generation by using predicted (Pred), GVR-based (Pred_GVR), and oracle event timelines (True) over the no-timeline baseline (w/o Timeline).

object, moving it to a new location, opening/closing a fridge), which are precisely the cues needed to ground "where", "what", and "when" questions.

To understand the effectiveness of each GVR mechanism, we analyzed the verification and refinement process across the three fine-tuned models. Our results show that GVR acts as a conservative verification mechanism rather than aggressively modifying the predicted event timeline. Specifically, refinement is triggered in only 583 out of 7,308 segments (8.0%), indicating that GVR intervenes only when verification identifies potentially unreliable facts. Across all models, only 4.0% of candidate facts are rejected (*e.g.,* 1,205 out of 30,132 verified facts), while the false rejection rate is only 1.2%, suggesting that GVR rarely discards correct facts. Furthermore, the verification stage achieves an overall fact-level accuracy of 63.3%, demonstrating that it can reasonably distinguish reliable from unreliable event facts.

### 3.2 Ablation Studies and Analysis

**Impact of Predicted Event Timelines on Response Generation** To quantify the impact of the predicted event timelines, we examine whether higher-quality timelines lead to more accurate responses. As shown in Figure 3, we compute a *whole video-level* timeline F1 by averaging segment-level F1 across all segments within each video. This segment-level F1 is computed as the average of the state and action F1s. The video-level timeline F1 is then applied to all questions in the same dialogue, and response accuracy is measured using LLM-Acc. We partition video-level F1 scores into four quartiles and report the mean LLM-Acc across questions within the dialogues for each quartile. In addition, because multiple questions are derived based on a single video and are therefore not independent, we estimate uncertainty using video-clustered bootstrap confidence intervals by resampling videos with replacement and recomputing quartile means. The results show a clear relationship, as videos with a higher F1 consistently achieve higher accuracy. This confirms that improvements in timeline tracking translate into more accurate response generation.

To characterize the gain over the no-context setting, Figure 4 reports the improvement obtained by adding predicted event timelines (Pred) and GVR-based timelines (Pred_GVR), measured relative to the no-context baseline (w/o Timeline). Predicted timelines deliver improvements across all metrics for all models. In particular, the gain is highest for MiMo-VL by predicted timelines, corresponding to 38.8% by SPICE, 26.5% by LLM-Acc, and 31.5% by LAVE of the oracle gains obtained with gold event timelines. Moreover, GVR-based event timelines yield further performance improvements over single-pass event timeline generation. The highest gain for GVR is observed with LLaVA-OV, which achieves gains of 39% and 40.3% on LLM-Acc and LAVE, respectively. While gold timelines remain an oracle upper bound, the predicted event timelines and GVR strategy provide measurable benefits, thereby validating their practical utility.

**Error Prediction by Action Types and Objects** We examine when and how the model incorrectly predicts actions and objects for event timelines to

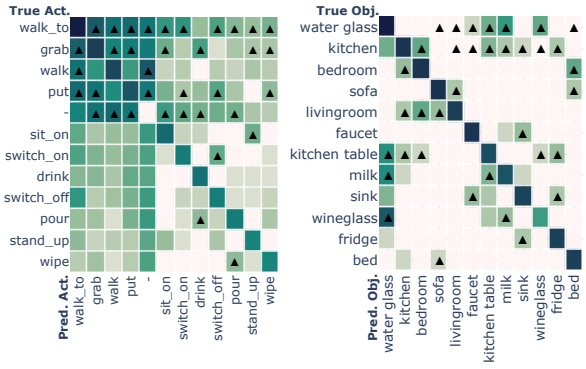



Figure 5: Confusion matrices for the fine-tuned Qwen2.5-VL. The left matrix shows predicted action types conditioned on the most frequent true actions, while the right matrix shows predicted objects conditioned on correctly predicted actions. Rows indicate true labels, and columns denote model predictions. ▲ indicates the top 3 frequently misclassified true classes.

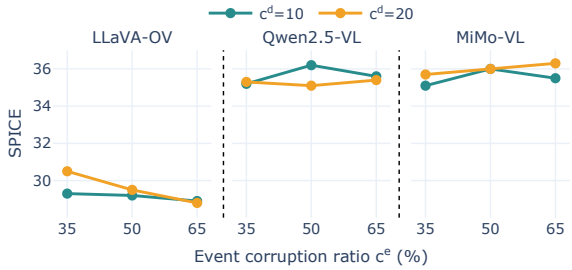



Figure 6: Effect of different combinations of event corruption ratio $c^e$ and dialogue corruption ratio $c^d$ during model training on downstream response generation performance based on SPICE across different models.

better understand its behavior. The confusion matrices in Figure 5 highlight systematic biases in the model's event timeline tracking. For actions, we observe that frequently occurring classes such as `walk_to`, `grab`, and `put` serve as default fallbacks, as they are over-predicted even when incorrect. This is due to their visual dominance in training data. The missed action class ("-") is broadly distributed across true classes. Notably, the confusion between `switch_on` with `switch_off` and `sit_on` with `stand_up` are bidirectional. This suggests the model's difficulty in distinguishing visually similar but semantically opposite actions. On the object side, confusion clusters are observed among semantically and visually similar entities, such as *water glass*, *milk*, and *wineglass*, since all of which are small, handheld, and translucent. Similarly, the bedroom, kitchen, and living room are often confused because of their shared visual context. The diagonally dominant structure, especially on the object side, suggests that once the action is correctly identified, the model often tracks the correct object. This supports the interdependence of action and object recognition. These observations highlight areas for improving model discriminability, particularly in visually ambiguous contexts.

**Impact of Different Combinations of Corruption Ratios** As illustrated in Figure 6, there is no single corruption ratio combination that consistently performs best across all models. The optimal setting is model-dependent: $c^e$=35, $c^d$=20 for LLaVA-OV, $c^e$=50, $c^d$=10 for Qwen2.5-VL and $c^e$=65, $c^d$=20 for MiMo-VL. When observing the performance trend, LLaVA-OV performs best with lower event timeline corruption, suggesting that excessive corruption can weaken the reliability of structured supervision for this model. Meanwhile, Qwen2.5-VL benefits from $c^e$=50 when $c^d$=10, while its performance is relatively stable under $c^d$=20. In contrast, MiMo-VL continues to improve with higher event corruption, indicating greater tolerance to noisy tracking supervision.

## 4 Related Works

**Video-Grounded Dialogue** Dialogue grounded in video content has received growing attention, with the development of benchmarks such as the Audio-Visual Scene-Aware Dialog (AVSD) dataset (Alamri et al., 2019), which introduces multi-turn conversations based on short, action-centric videos. More recently, the Video-grounded Dialogue on Event-driven Activities (VDAct) dataset (Imrattanatrai et al., 2025) extended this paradigm to longer videos depicting home activities performed by a human agent, featuring over 30,000 turns of diverse dialogue. Experimental results on the VDAct dataset demonstrate that both open-weight and proprietary VLMs face significant challenges: they often produce partially correct answers, and overall performance remains limited.

**Caption-Based Strategies** A common approach to mitigate the difficulty of direct video-to-dialogue modeling is to first prepare frame captions or segment summaries used as primary or auxiliary input for video understanding (Kim et al., 2020; Mogrovejo and Solorio, 2024). While such fine-grained text simplifies visual input, it yields large context sizes and limits self-attention's ability to capture long-range dependencies (Tay et al., 2021).

VideoAgent (Fan et al., 2024) adopts an agent-based paradigm that leverages multiple models to retrieve task-relevant captions. Recent studies explore enhancing multimodal reasoning via explicit chain-of-thought over visual inputs (Lu et al., 2022; Sun et al., 2025), but these approaches still typically rely on long unstructured textual contexts. In contrast, our work focuses on improving the input representation itself: using concise event timelines, we enable even reasoning-capable VLMs, such as MiMo-VL (Xiaomi, 2025) and GPT-5) to better capture long-horizon structured information.

**Video to Structured Representations** Another line of research focuses on converting videos into structured knowledge representations. VirtualHome2KG (Egami et al., 2023) demonstrated that activity videos can be abstracted into event-centric knowledge graphs capturing actions, objects, spatial relations, and temporal order. Related efforts such as Vid2Seq (Yang et al., 2023) also map videos into event-level sequences, but their outputs largely resemble compact natural language descriptions, inheriting the expressive limitations of caption-based representations. In contrast, we argue that structured event representations provide a more faithful and compact abstraction of video content. In this work, we instantiate this idea in video-grounded dialogue using event timelines to improve response generation.

## 5 Conclusion

This paper presents a modular VGD approach that uses structured event timelines as explicit grounding context. We further introduce a GVR mechanism to verify and revise predicted timeline facts before they are used for downstream dialogue reasoning. Experiments show that fine-tuned models conditioned on event timelines consistently outperform those using visual captions or no structured context, with GVR-based timelines yielding further gains across automatic and human evaluations. For future work, we aim to apply our approach to real-world videos and to reduce models' sensitivity to noise in predicted event timelines.

## Limitations

Our evaluations and ablation studies are limited to the existing video-grounded dialogue dataset, in which dialogues were derived from videos depicting simulator-generated household activities. This narrow, indoor, scripted setting may not capture the variability of open environments such as outdoor scenes and domain-specific objects and activities. As a result, generalization under domain shift is uncertain, and performance may degrade without adaptation. Future work will assess robustness on real-world, open-domain datasets and explore domain-adaptation strategies.

## Acknowledgments

This research was conducted by AIST and CMU under: (1) FRONTia, a Japanese national program led by the Ministry of Economy, Trade and Industry of Japan (METI) and the New Energy and Industrial Technology Development Organization (NEDO), aimed at developing a domestic multimodal foundation model for AI robots and physical AI, and (2) the Programs for Bridging the Gap between R&D and the IDeal Society (Society 5.0) and Generating Economic and Social Value (BRIDGE), "R7-H05: Demonstration, Research, and Development for Safety in AI and Robotics Fields / Research and Development for Strengthening AI Safety in Industrial Domains", implemented by the Cabinet Office, Government of Japan.

## References

Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K Marks, Chiori Hori, Peter Anderson, and 1 others. 2019. Audio visual scene-aware dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7558–7567.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

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.

Shusaku Egami, Satoshi Nishimura, and Ken Fukuda. 2021. Virtualhome2kg: Constructing and augmenting knowledge graphs of daily activities using virtual space. In *ISWC (Posters/Demos/Industry)*.

Shusaku Egami, Takanori Ugai, Swe Nwe Nwe Htun, and Ken Fukuda. 2024. Vhakg: A multi-modal knowledge graph based on synchronized multi-view videos of daily activities. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 5360–5364.

Shusaku Egami, Takanori Ugai, Mikiko Oono, Koji Kitamura, and Ken Fukuda. 2023. Synthesizing event-centric knowledge graphs of daily activities using virtual space. *IEEE Access*, 11:23857–23873.

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

Jinbae Im, JeongYeon Nam, Nokyung Park, Hyungmin Lee, and Seunghyun Park. 2024. Egtr: Extracting graph from transformer for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24229–24238.

Wiradee Imrattanatrai, Masaki Asada, Kimihiro Hasegawa, Zhi-Qi Cheng, Ken Fukuda, and Teruko Mitamura. 2025. A video-grounded dialogue dataset and metric for event-driven activities. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24203–24211.

Hyounghun Kim, Zineng Tang, and Mohit Bansal. 2020. Dense-caption matching and frame-selection gating for temporal localization in VideoQA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4812–4822, Online. Association for Computational Linguistics.

Terry K. Koo and Mae Y. Li. 2016. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*, 15(2):155–163.

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

Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. 2022. Learning to answer questions in dynamic audio-visual scenarios. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19108–19118.

Rongjie Li, Songyang Zhang, Dahua Lin, Kai Chen, and Xuming He. 2024b. From pixels to graphs: Open-vocabulary scene graph generation with vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28076–28086.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Hongcheng Liu, Pingjie Wang, Yu Wang, and Yanfeng Wang. 2024a. M2k-vdg: Model-adaptive multimodal knowledge anchor enhanced video-grounded dialogue generation. *arXiv preprint arXiv:2402.11875*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*.

Oscar Mañas, Benno Krojer, and Aishwarya Agrawal. 2024. Improving automatic vqa evaluation using large language models. In *Proceedings of The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)*, pages 4171–4179.

David Mogrovejo and Thamar Solorio. 2024. Question-instructed visual descriptions for zero-shot video answering. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9329–9339, Bangkok, Thailand. Association for Computational Linguistics.

Trong-Thuan Nguyen, Pha Nguyen, Jackson Cothren, Alper Yilmaz, and Khoa Luu. 2025. Hyperglm: Hypergraph for video scene graph generation and anticipation. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 29150–29160.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Idan Schwartz, Alexander G Schwing, and Tamir Hazan. 2019. A simple baseline for audio-visual scene-aware dialog. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12548–12558.

Hai-Long Sun, Zhun Sun, Houwen Peng, and Han-Jia Ye. 2025. Mitigating visual forgetting via take-along visual conditioning for multi-modal long CoT reasoning. In *Proceedings of the 63rd Annual Meeting of*

*the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5158–5171, Vienna, Austria. Association for Computational Linguistics.

Zineng Tang, Jie Lei, and Mohit Bansal. 2021. Decembert: Learning from noisy instructional videos via dense captions and entropy minimization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2415–2426.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*.

LLM-Core-Team Xiaomi. 2025. Mimo-vl technical report. *Preprint*, arXiv:2506.03569.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. 2023. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10714–10726.

Qilang Ye, Zitong Yu, Rui Shao, Yawen Cui, Xiangui Kang, Xin Liu, Philip Torr, and Xiaochun Cao. 2025. Cat+: Investigating and enhancing audio-visual understanding in large language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

# A Appendix

## A.1 Additional Statistics

**Two-stage Training Samples** Tables 5 and 6 summarize the statistics of training samples constructed for the dialogue response generation and event timeline tracking tasks under different settings of corruption ratios $c^d$ and $c^e$, respectively. For dialogue response generation, we generate 2,400 training samples from 800 unique activity videos while varying the dialogue corruption ratio $c^d \in \{0.1, 0.2\}$. As expected, increasing $c^d$ in proportion increases the number of segments containing corrupted event timelines, from 3,992 at 10% to 7,984 at 20%. These corrupted samples expose the model to imperfect structured grounding during training, encouraging robustness to noisy upstream timeline predictions.

For event timeline tracking, we construct 14,209 training samples from 229 activity videos under event timeline corruption ratios $c^e \in \{0.35, 0.5, 0.65\}$. The statistics show that increasing $c^e$ substantially changes the verification dynamics within the GVR process. In particular, the proportion of samples containing false verification cases increases from 35.1% at $c^e = 0.35\%$ to 66.6% at $c^e = 0.65\%$, while samples containing only true verification cases decrease from 39.8% to 8.3%. At the same time, the average verification length decreases as the corruption ratio increases, whereas the average completion length becomes longer. This trend suggests that higher corruption ratios force the model to perform refinement more frequently by exposing it to increasingly unreliable candidate event timelines.

| | $c^d$=10% | $c^d$=20% |
|---|---|---|
| Total Video Segments | 39,924 | 39,924 |
| # Video Segments with Corruption | 3,992 | 7,984 |
| # Unique Videos | 800 | 800 |
| # Training Samples | 2,400 | 2,400 |

Table 5: Statistics of training samples constructed for dialogue response generation under different settings of corruption ratio $c^d$.

## A.2 Qualitative Results

We present the actual results generated by the finetuned models. Tables 7 and 9 show the actual generated responses for two dialogues by the Qwen2.5-VL model, finetuned with two different settings for grounding context: (1) dialogue only, (2) dialogues with event timelines, and (3) dialogues with GVR-based event timelines, while Tables 8 and 10 present the predicted event timelines for the two corresponding activity videos.

While the model that relies solely on dialogue as a grounding context often generates responses containing inaccurate information, incorporating event timelines enhances response accuracy. For instance, at dialogue turn 3 in Table 7, the model equipped with event timelines answers a numerical question that asks “How many", by identifying the correct number of wineglasses. Referring to the predicted event timelines in Table 8, we observe that the 4th segment includes actions explicitly involving two wineglasses. This enables the model to ground its response based on structured evidence, resulting in a more precise answer. Similarly, in dialogue turn 1 of Table 9, the dialogue-only model fails to provide an appropriate response to a question asking for a description of the first activity the person performed. Meanwhile, as shown in Table 10, although some segments contain incorrect information, the version with access to event timelines correctly identifies “drinking milk” as the first ma-

| | $c^e$=35% | $c^e$=50% | $c^e$=65% |
|---|---|---|---|
| # Unique Videos | 229 | 229 | 229 |
| Avg. Facts Per Video Segment | 3.11 | 3.11 | 3.11 |
| Avg. Segments Per Video | 15.57 | 15.57 | 15.57 |
| # Training Samples | 14,209 | 14,209 | 14,209 |
| # Samples with No Committed Facts | 5,689 | 5,645 | 5,719 |
| # Samples with No Committed Facts, but with Candidate Facts | 2,124 | 2,080 | 2,154 |
| # Samples with No Candidate Facts | 3,565 (25.1%) | 3,565 (25.1%) | 3,565 (25.1%) |
| # Samples with Only True Verification | 5,653 (39.8%) | 3,451 (24.3%) | 1,177 (8.3%) |
| # Samples with Only False Verification | 4,991 (35.1%) | 7,193 (50.6%) | 9,467 (66.6%) |
| Avg. Committed Facts | 1.63 | 1.63 | 1.63 |
| Avg. Candidate Facts | 3.01 | 2.95 | 2.67 |
| Avg. Verification Facts | 2.38 | 1.70 | 1.10 |
| Avg. Completion Facts | 2.07 | 2.74 | 3.34 |

Table 6: Statistics of training samples for event timeline tracking under different settings of corruption ratio $c^e$.

jor activity. As can be seen in the 5th segment, the timeline suggests that the person is performing actions such as grabbing a milk carton and a water glass, placing them on the kitchen table, pouring milk into the water glass, and drinking from it. This structured event timeline allows the model to provide a contextually grounded and accurate response.

### A.3 Additional Analysis

**Action Tracking Accuracy by Types** To better understand the model's behavior in event tracking, we analyze which action types are easier or harder for the model to track. Figure 7 presents the F1-scores of Qwen2.5-VL, fine-tuned using our two-stage approach with GVR strategy, across different action types. We observe that top-performing actions such as `type_on`, `brush`, `stretche`, and `text_on` achieve relatively high F1-scores ($\geq 0.7$), indicating that the model is highly reliable in detecting these actions. These actions often exhibit distinctive and repetitive visual patterns (*e.g.,* brushing teeth or stretching arms/legs), which likely makes them easier to recognize and align with the textual representation. In contrast, actions like `wipe`, `stand_up`, `sit_on`, `pour`, and `drink` receive moderate F1-scores, ranging from 0.5 to 0.7. These actions typically involve subtler motion cues or object interactions that may vary in appearance and speed across different instances, making consistent detection more difficult. Moreover, the temporal boundaries of these actions are often ambiguous, contributing to reduced prediction accuracy. The most challenging actions, with F1-scores below 0.3, include `fold`, `turn_to`, and `squat`. These actions may be visually subtle (*e.g.,* folding clothes), or depend on abstract goals (*e.g.,* "turning" to an item), which limit the model's ability to detect from visual cues alone. Additionally, there are actions that the model fails to track entirely, such as `throw`, `jump`, `fall_back`, and `climb`. These failures could be attributed to their rarity in the training data and the lack of a clear visual associated with these actions.

**Object Accuracy** In addition to analyzing action types, object recognition is crucial to our event timeline tracking module. To investigate the model's capability in tracking objects, we conduct a similar analysis, as shown in Figure 8. Given the large number of object categories, we focus on the top-20 and bottom-20 objects based on the F1-scores of the LoRA-finetuned Qwen2.5-VL model. Importantly, this analysis only considers objects associated with correctly predicted actions to isolate errors related to object grounding. Interestingly, the top-performing objects are predominantly associated with living rooms and bathrooms, such as `keyboard`, `computer`, `sofa`, `faucet`, `toothpaste`, and `toilet`. These objects tend to have consistent appearances and are usually involved in well-structured, visually distinctive interactions (*e.g.,* typing on a keyboard, using a toilet, or sitting on a sofa), which likely contributes to their higher detection accuracy. In contrast, the bottom-performing objects, such as `cupcake`, `pie`, `chips`, `cutlery fork`, and `milk`, are largely associated with kitchen scenes. These objects are often small, visually similar to one another, or partially occluded during interaction, making them more challenging to identify reliably. Moreover, kitchen scenes typically involve a wider variety of objects and fine-grained manipulations, which introduces additional complexity in object grounding.

This contrast reveals an important insight as the model performs better in scenes with a limited and consistent set of objects and actions (*e.g.,* bathrooms and living rooms) compared to more clut-

| t | Question | Gold Responses | Dialogues Only Context | Dialogues and Event Timelines | Dialogues and GVR-based Event Timelines |
|---|---|---|---|---|---|
| 1 | Where does the man get the wine? | He grabs the wine off of one of the tables in the kitchen. | He gets it from the kitchen table. | From the kitchen table. | From the kitchen table. |
| 2 | Does the man place the wine back on the same table, or on a different table? | He puts it back onto the same table. | He places it on a different table. | He places it on the same table he got it from. | He places it back onto the same table. |
| 3 | How many dirty wine glasses are there? | He picks up two dirty wine glasses. | Just one. | There are two dirty wine glasses. | There are two dirty wine glasses. |
| 4 | Does he wash the wine glasses? | No, he only empties them and puts them into the sink. | No, he takes them to the sink. | No, he takes them to the sink and leaves them there. | No, he takes them to the sink but doesn't wash them. |
| 5 | Where does the man get the knife? | He picks it up from the kitchen table. | He gets the knife from the counter near the sink. | He gets the knife from one of the tables in the kitchen. | He gets the knife from the kitchen table. |
| 6 | Does the man place any items in addition to the one knife? | Yes, he places a second knife. | No, just the knife. | Yes, he places down a wine glass as well. | No, just the one knife. |
| 7 | Does he procure the two knives from the same table? | No, he gets the knives from different tables. | Yes, they were on the same table. | Yes, they were both on the same table. | Yes, both were picked up from the same table. |
| 8 | Does he place the knives at the same table setting? | Yes, he places both knives at the same place setting. | Yes, he places both knives on the same table. | Yes, he places both knives on the same table. | Yes, he places both knives on the same table. |
| 9 | Is the man holding anything when he falls? | Yes, he is holding two items. | No, he is empty handed. | Yes, he is holding a bottle of liqueur and a cupcake. | Yes, three items. |
| 10 | What is the man holding? | He is holding a cupcake and a jug of milk. | No, he is empty handed. | A towel and a bottle of liqueur. | A bowl of cereal and some milk. |

Table 7: The actual generated responses for dialogue ID `0006401` by LoRA finetuned Qwen2.5-VL under two different settings for grounding context: (1) dialogue only, (2) dialogues with event timelines, and (3) dialogues with GVR-based event timelines. Text in red color indicates the inaccurate information within the response.

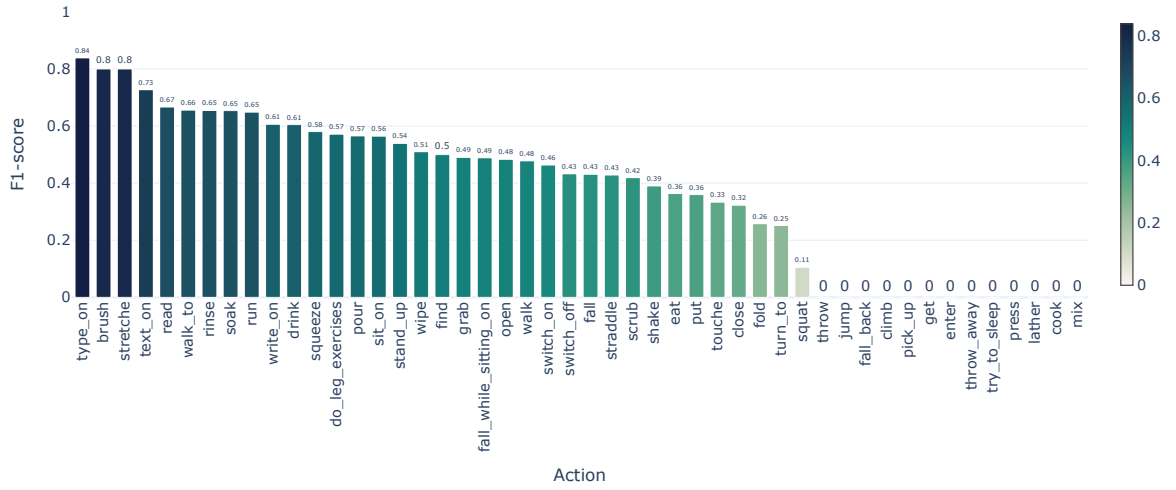



Figure 7: Performance of LoRA finetuned Qwen2.5-VL by the action types, ranked by the F1-score.

tered and dynamic environments like kitchens. This suggests that object and scene complexity significantly influence the model's event tracking performance, highlighting the need for more robust training across diverse, visually complex settings.

### A.4 Task Instructions

We present the prompt instructions and in-context examples for each of our experimental setups.

| k | Gold ET | Predicted ET | Predicted ET + GVR |
|---|---|---|---|
| 1 | inside(person, kitchen) | inside(person, kitchen)<br>next_to(person, kitchen table)<br>attached_to(wineglass, person) | inside(person, kitchen)<br>next_to(person, kitchen table)<br>next_to(person, wine) |
| | walk_to(person, wine)<br>grab(person, wine, from kitchen table)<br>walk_to(person, kitchen table)<br>put(person, wine, in/on kitchen table)<br>walk(person) | put(person, wineglass, in/on kitchen table)<br>walk_to(person, wine)<br>grab(person, wine, from kitchen table)<br>walk(person) | put(person, wine, in/on kitchen table)<br>walk_to(person, wine-2)<br>grab(person, wine-2, from kitchen table)<br>walk(person) |
| 2 | inside(person, kitchen) | inside(person, kitchen) | inside(person, kitchen) |
| | walk_to(person, wineglass)<br>grab(person, wineglass, from kitchen table)<br>walk(person) | walk_to(person, water glass)<br>grab(person, water glass, from kitchen table)<br>walk_to(person, kitchen table)<br>put(person, water glass, in/on kitchen table) | walk_to(person, water glass)<br>grab(person, water glass, from kitchen table)<br>walk_to(person, water glass-2)<br>grab(person, water glass-2, from kitchen table) |
| 3 | inside(person, kitchen)<br>next_to(person, wineglass-2)<br>attached_to(wineglass, person) | inside(person, kitchen)<br>next_to(person, kitchen table)<br>attached_to(water glass, person) | inside(person, kitchen)<br>next_to(person, water glass)<br>attached_to(juice, person) |
| | grab(person, wineglass-2, from kitchen table)<br>walk_to(person, sink)<br>pour(person, wineglass, into sink) | put(person, water glass, in/on kitchen table)<br>walk_to(person, cupcake)<br>grab(person, cupcake, from kitchen counter)<br>eat(person, cupcake) | grab(person, water glass, from kitchen table)<br>walk_to(person, sink)<br>pour(person, water glass, into sink) |
| 4 | inside(person, kitchen)<br>next_to(person, sink)<br>attached_to(wineglass, person)<br>attached_to(wineglass-2, person) | inside(person, kitchen)<br>next_to(person, sink)<br>attached_to(wineglass, person)<br>attached_to(wineglass-2, person) | inside(person, kitchen)<br>next_to(person, sink)<br>attached_to(wineglass, person)<br>attached_to(wineglass-2, person) |
| | pour(person, wineglass, into sink)<br>pour(person, wineglass-2, into sink) | pour(person, wineglass, into sink)<br>pour(person, wineglass-2, into sink) | pour(person, wineglass-2, into sink)<br>put(person, wineglass, in/on sink) |
| 5 | inside(person, kitchen)<br>next_to(person, sink)<br>attached_to(wineglass, person)<br>attached_to(wineglass-2, person) | inside(person, kitchen)<br>next_to(person, sink)<br>attached_to(water glass, person)<br>attached_to(water glass-2, person) | inside(person, kitchen)<br>next_to(person, sink)<br>attached_to(water glass, person)<br>attached_to(water glass-2, person) |
| | pour(person, wineglass-2, into sink)<br>put(person, wineglass, in/on sink)<br>put(person, wineglass-2, in/on sink)<br>walk(person) | pour(person, water glass, into sink)<br>pour(person, water glass-2, into sink)<br>put(person, water glass, in/on sink)<br>put(person, water glass-2, in/on sink) | pour(person, water glass-2, into sink)<br>put(person, water glass, in/on sink)<br>put(person, water glass-2, in/on sink) |

Table 8: The actual outputs for event timeline tracking of the first five video segments of scenario video ID `00064` by Qwen2.5-VL, which was fine-tuned for our approach.

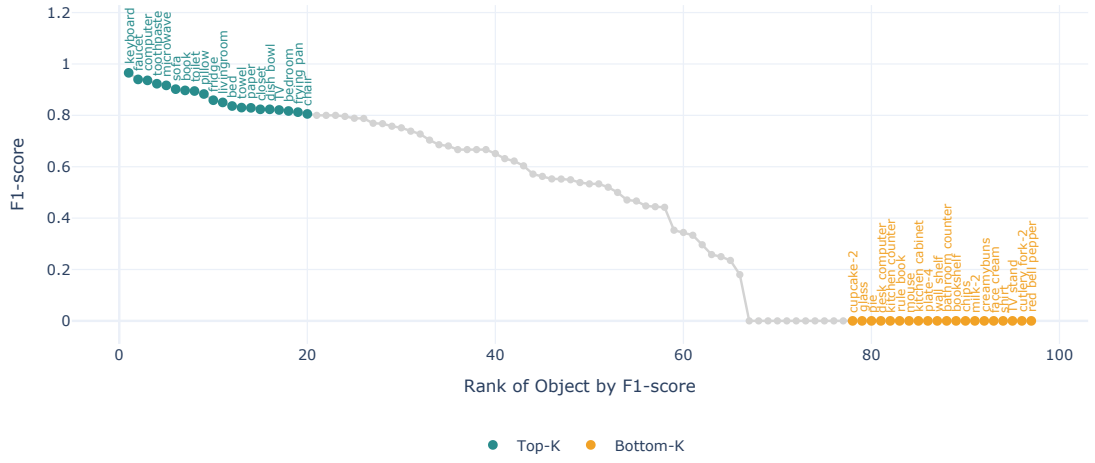



Figure 8: Performance of fine-tuned Qwen2.5-VL for top- and bottom-20 predicted objects, ranked by the F1-score.

**Prompt for Event Timeline Tracking** The event timeline tracking task employs two types of system prompts. One is for single-pass event timeline tracking, and the other is for our proposed GVR-based. The prompts for the single-pass and GVR-based inference are shown in Figures 9 and 10.

| t | Question | Gold Responses | Dialogues Only Context | Dialogues and Event Timelines | Dialogues and GVR-based Event Timelines |
|---|---|---|---|---|---|
| 1 | Which activity did the man do in the beginning? | He started by drinking milk. | He started by making some cold cereal. | He started by drinking some milk. | He started by drinking some milk. |
| 2 | Where did he drink milk? | In the kitchen. | In the kitchen. | In the kitchen. | In the kitchen. |
| 3 | Was his next activity also performed in the kitchen? | Yes, he turned off the light in the kitchen and then went to the living room. | Yes, it was. | Yes, it was. | No, it was in the living room. |
| 4 | What did he do in the living room? | He picked up some books and then fell over. | Nothing really, he just walked into it. | He picked up two books but fell over. | He picked up two books but fell over. |
| 5 | How many books did he pick up? | Two. | Two books. | Two. | Two books. |
| 6 | What happened next? | He got up again but soon fell over as he as trying to walk. | Next, he went back into the kitchen to cook some potatoes. | Next, he went back to the kitchen to take out some milk from the fridge. | Next, he opened and closed the fridge but didn't seem to take anything out. |
| 7 | Was this also in the living room? | It was indeed. | Yes, it was. | Yes, it was. | Yes, it was. |
| 8 | Where did he go after falling? | He went back to the kitchen. | Back into the kitchen. | Back into the kitchen. | Back to the kitchen. |
| 9 | What did he do this time? | He did some cooking on the stove. | He placed a wine glass on the table. | First he took out some milk from the fridge. | He cooked some fried bread. |
| 10 | What did he cook? | Some fried bread with milk. | Some kind of porridge or cereal. | Some milk. | Some fried bread with milk. |
| 11 | Can you tell me anything else about the man's activities? | He spent most of his time either in the living room or the kitchen. | After cooking, he placed a wineglass on the table. | He spent most of his time in the kitchen. | After finishing the milk bread dish he grabbed a plate from the counter. |

Table 9: The actual generated responses for dialogue ID `0041202` by LoRA finetuned Qwen2.5-VL under two different settings for grounding context: (1) dialogue only, (2) dialogues with event timelines, and (3) dialogues with GVR-based event timelines. Text in red color indicates the inaccurate information within the response.

Other details inside the GVR-based event tracking prompt are presented in Figure 11.

**Fixed Set of Interaction and Objects for Proprietary Models** The part of the prompt that instructs on a fixed set of plausible interactions and objects used by proprietary models is shown in Figure 12. This part of the prompt is inserted between the format specification and the output structure of the prompts for single-pass and GVR-based event timeline tracking in Figures 9 and 10.

**Few-shot Examples for Single-Pass Event Timeline Tracking** The in-context examples for generating single-pass event timelines for the few-shot settings used by proprietary models are shown in Figure 13.

**Prompt for Dialogue Response Generation** We use the default prompt provided by each model across all settings, except when the model uses GVR-based event timelines. The prompt for such a model setting is shown in Figure 14.

**Prompt for Template-based Caption Generation** Figure 15 presents the prompt used to instruct the VLMs to generate the template-based caption given the video frames.

**Prompt for Refined Caption Generation** The prompt presented in Figure 16 is used by the VLMs to generate the refined video caption given a scenario video.

**Refined Caption Creation** The ground-truth refined captions for model fine-tuning are generated from the template captions using the following prompt shown in Figure 17.

**Prompt for Fact Corruption** Figure 18 presents a full prompt for producing a corrupted fact given an original fact.

### A.5 Human Evaluation Protocol

We conducted a human evaluation of model responses using the annotation interface shown in Figure 19. During annotation, workers were able to inspect the video, the previous dialogue history, and the reference answer before rating the model responses. The interface presented multiple model responses to the same question on a single

| k | Gold ET | Predicted ET | Predicted ET + GVR |
|---|---|---|---|
| 1 | inside(person, bedroom) | inside(person, bedroom) | inside(person, bedroom) |
| | walk_to(person, kitchen)<br>walk(person) | walk_to(person, kitchen)<br>walk_to(person, chicken)<br>grab(person, chicken, from kitchen counter) | walk_to(person, kitchen)<br>walk_to(person, salmon)<br>grab(person, salmon, from kitchen counter) |
| 2 | inside(person, kitchen)<br>next_to(person, milk) | inside(person, kitchen)<br>next_to(person, milk) | inside(person, kitchen)<br>next_to(person, plate) |
| | grab(person, milk, from kitchen counter drawer)<br>walk_to(person, kitchen table)<br>put(person, milk, in/on kitchen table) | grab(person, milk, from kitchen counter)<br>walk_to(person, water glass)<br>grab(person, water glass, from kitchen table)<br>drink(person, water glass) | grab(person, plate, from kitchen counter)<br>walk_to(person, water glass)<br>grab(person, water glass, from kitchen table)<br>walk(person) |
| 3 | inside(person, kitchen)<br>next_to(person, kitchen table)<br>next_to(person, milk) | inside(person, kitchen) | inside(person, kitchen) |
| | put(person, milk, in/on kitchen table)<br>walk_to(person, water glass)<br>grab(person, water glass, from kitchen table-2)<br>walk_to(person, kitchen table)<br>put(person, water glass, in/on kitchen table) | walk_to(person, water glass)<br>grab(person, water glass, from kitchen table)<br>walk_to(person, kitchen table)<br>put(person, water glass, in/on kitchen table) | walk_to(person, water glass)<br>grab(person, water glass, from kitchen table)<br>walk_to(person, water glass-2)<br>grab(person, water glass-2, from kitchen table) |
| 4 | inside(person, kitchen)<br>next_to(person, kitchen table)<br>next_to(person, water glass) | inside(person, kitchen)<br>next_to(person, water glass) | inside(person, kitchen)<br>next_to(person, water glass) |
| | put(person, water glass, in/on kitchen table)<br>walk_to(person, milk)<br>grab(person, milk, from kitchen table)<br>walk_to(person, water glass)<br>pour(person, milk, into water glass) | grab(person, water glass, from kitchen table)<br>walk_to(person, kitchen table)<br>put(person, water glass, in/on kitchen table)<br>walk_to(person, milk)<br>grab(person, milk, from kitchen table) | grab(person, water glass, from kitchen table)<br>walk_to(person, water glass-2)<br>grab(person, water glass-2, from kitchen table)<br>walk_to(person, water glass-2)<br>pour(person, water glass, into water glass-2) |
| 5 | inside(person, kitchen)<br>next_to(person, water glass)<br>attached_to(milk, person) | inside(person, kitchen)<br>next_to(person, water glass)<br>attached_to(milk, person) | inside(person, kitchen)<br>next_to(person, water glass)<br>attached_to(milk, person) |
| | pour(person, milk, into water glass)<br>grab(person, water glass, from kitchen table)<br>drink(person, water glass) | pour(person, milk, into water glass)<br>grab(person, water glass, from kitchen table)<br>drink(person, water glass) | pour(person, milk, into water glass)<br>grab(person, water glass, from kitchen table)<br>drink(person, water glass) |

Table 10: The actual outputs for event timeline tracking for the first five video segments of scenario video ID 00412 by Qwen2.5-VL that was finetuned for our two-stage approach.

screen, while the order and display positions of the responses were randomized for each sample to reduce position bias. For each response, three human annotators independently assigned a score from 1 to 3, where 1 indicates Poor and 3 indicates Excellent. The final score for each response was determined by majority vote among the three annotators. When no majority score was obtained from the initial three annotations, a fourth annotator was assigned only to that sample and served as a tie-breaker. We then used the resulting majority score as the final score for the corresponding response. For each model, we computed the average of the final majority scores over all evaluated samples. Since the raw scores are on a 1–3 scale, we converted the average score to a 0–100 scale for reporting.

**Prompt for Single-pass Event Timeline Tracking**

You are an expert assistant that analyzes short video segments showing human activities in home environments.

Your task is to output structured information that summarizes the video as:
1. An initial **spatial state** describing key entities (person, objects, and locations) at the beginning of the video. This is represented under the ‘start:‘ key.
2. A sequence of **actions** performed by the person, listed in temporal order under the ‘actions:‘ key.

### Format Specification
- **start:** A list of spatial or possession relations, using the format: relation(subject, object)

Spatial relations include:
- inside(subject, location)
- next_to(subject, object)
- attached_to(object, person) -> indicates possession (e.g., the person is holding or carrying the object)
- **actions:** A list of actions performed by the person, in the following formats:
- verb(subject)
- verb(subject, object)
- verb(subject, object, with tool)
- verb(subject, object, in/on location/surface)
- verb(subject, object, from location)

### Output Format
start: [relation(subject, object); relation(subject, object); ...]
actions: [verb(...); verb(...); ...]

- Use lowercase throughout.
- Use ‘;‘ to separate items in each list.
- Avoid background or irrelevant objects.
- Do not include narrative, explanations, or full sentences-only structured output.

Figure 9: Prompt for a single-pass ET tracking.

**Prompt for a GVR-based Event Timeline Tracking**

You are an expert assistant that analyzes short video segments showing human activities in home environments.

Your task is to produce a structured event track for the video segment.

The event track consists of:
1) initial spatial states describing key entities (person, objects, and locations) at the beginning of the video segment
2) a sequence of actions performed by the person

You may be given:
- a committed list: previously accepted states and actions that should be treated as trusted context
- a candidate list: newly proposed states and actions that must be verified before being trusted

Your behavior depends on the input:
1. If no candidate list is provided:
- generate the full event track for the video segment under the section "completion"

2. If a candidate list is provided:
- first evaluate each candidate item with respect to the video segment and the committed list under the section "verification"
- then generate the remaining after the committed list and the latest true candidate under the section "completion"
- do not continue from false candidate items
- do not copy candidate items into the completion unless they are actually part of the remaining event track supported by the video

### Format Specification
### Output Structure
### Verification Rules
### Completion Rules
### Important Rules

### Example 1: no candidate list

Input:
committed
-

candidate
-

Output:
completion
state: inside(person, livingroom)
state: next_to(person, chair)
state: attached_to(paper, person)
action: sit_on(person, chair)
action: write_on(person, paper)
action: stand_up(person)
end

### Example 2: w. committed and candidate lists

Input:
committed
state: inside(person, bedroom)
state: attached_to(wineglass, person)

candidate
action: walk_to(person, kitchen)
action: grab(person, milk carton, from kitchen table)
action: put(person, milk carton, on kitchen counter)
end

Output:
verification
action: walk_to(person, kitchen) [true]
action: grab(person, milk carton, from kitchen counter) [false]

completion
action: grab(person, wine bottle, from kitchen table)
action: pour(person, wine, into wineglass)
action: drink(person, wineglass)
end

Figure 10: Prompt for a GVR-based ET tracking.

**Details for a GVR-based Event Timeline Tracking**

### Format Specification

Each line must strictly follow one of the formats below.

State format:
state: relation(subject, object)

Allowed spatial relations:
- inside(subject, location)
- next_to(subject, object)
- attached_to(object, person) -> indicates possession (e.g., the person is holding or carrying the object)

Action formats:
- action: verb(subject)
- action: verb(subject, object)
- action: verb(subject, object, with tool/into tool/on location/in location/ from location)

### Output Structure
If no candidate list is provided, output only:

completion
state: ...
state: ...
action: ...
action: ...
end

If a candidate list is provided, output exactly two sections in this order:

verification
item [true]
item [false]

completion
state: ...
action: ...
...
end

### Verification Rules

- Evaluate only the candidate items (including "end" in the candidate list).
- Repeat each candidate item exactly as given, then append [true] or [false].
- Judge each candidate item based on the video segment and the committed list.
- Stop verification when one of the candidate items is not true.
- A candidate item is [true] only if it is clearly supported by the video and consistent with the committed list.
- A candidate item is [false] if it is unsupported, contradicted by the video, or inconsistent with the committed list.

### Completion Rules

- The completion must describe the remaining event track after the committed list and the latest true candidate.
- If no committed list is provided, the completion must describe the full event track.
- Do not repeat committed items in the completion.
- Do not include explanations or full sentences.
- First list all remaining states, then list all remaining actions.
- Do not list states that are not represented at the start of the video segment (only the first video frame).
- Output "end" exactly once to indicate the end of the completion sequence.

### Important Rules

- Only include entities that are relevant to the event track.
- Do not include background or irrelevant objects.
- Only the initial spatial states of key entities (focus on person) at the start of the video segment are required.
- The action list should follow the order in which the actions occur.

Figure 11: Format specification, output structure, and rules inside the prompt for the GVR-based ET tracking.

**Fixed Set of Subjects-Relations-Objects for Event Timeline Tracking**

### Allowed Vocabulary (Closed Set Constraint)

You must use **only the following predefined vocabulary** when producing outputs.

- **Allowed verbs:**
walk, walk_to, run, jump, climb, turn_to, sit_on, stand_up, straddle, squat, fall, fall_back, fall_on_the_table, fall_while_sitting_on, grab, put, throw, shake, fold, open, close, switch_on, switch_off, touch, squeeze, find, look_at, read, write_on, type_on, text_on, eat, drink, pour, rinse, soak, scrub, wipe, brush, stretch, do_leg_exercises

- **Allowed subjects:**
person

- **Allowed objects/locations/surfaces/tools:**
kitchen, livingroom, bedroom, bathroom, floor, kitchen counter, kitchen counter drawer, kitchen cabinet, kitchen table, kitchen furniture, coffee table, desk, TV stand, bookshelf, wall shelf, sofa, chair, bench, bed, toilet, sink, faucet, bathroom counter, dish washer, stove, microwave, fridge, toaster, cabinet, closet, closet drawer, door, curtains, lightswitch, wall picture frame, TV, computer, keyboard, mouse, cellphone, remote control, book, book stack, notes, paper, pen, box, laundry, pile of clothes, pants, shirt, towel, towel stack, pillow, slippers, garbage can, dish bowl, dish bowl stack, plate, cutlery knife, cutlery fork, wineglass, water glass, washing sponge, dish washing liquid, bar soap, toothbrush, toothpaste, face cream, alcohol, wine, juice, milk, cereal, crackers, chips, cupcake, creamybuns, slice of bread, salmon, chicken, salad, potato, carrot, chocolate syrup, cat

Figure 12: Fixed set of plausible subjects, verbs, and objects for the event timeline tracking.

**Examples for Single-Pass Event Timeline Tracking**

### Example Output #1

start: [inside(person, kitchen); next_to(person, cutlery knife); next_to(person, kitchen table); attached_to(cutlery fork, person)]
actions: [put(person, cutlery fork, on kitchen table); walk_to(person, wineglass); grab(person, wineglass, from kitchen table); walk(person)]

### Example Output #2

start: [inside(person, kitchen); attached_to(cupcake, person); attached_to(remote control, person)]
actions: [walk_to(person, livingroom); walk_to(person, sofa); sit_on(person, sofa); eat(person, cupcake)]

### Example Output #3

start: [inside(person, bathroom); next_to(person, toilet)]
actions: [sit_on(person, toilet); stand_up(person); fall(person)]

### Example Output #4

start: [inside(person, livingroom); next_to(person, chair); attached_to(paper, person)]
actions: [sit_on(person, chair); write_on(person, paper); stand_up(person)]

### Example Output #5

start: [inside(person, bedroom); attached_to(shirt, person)]
actions: [shake(person, shirt); fold(person)]

Figure 13: 5-shot examples for a single pass ET tracking used by proprietary models.

**Prompt for a Dialogue Response Generation**

You are an intelligent assistant that answers questions about human activities in home environments.

You are given:
1. A normal distributed sampling of video frames.
2. A dialogue history between the user and the assistant.
3. A set of predicted event facts extracted from video segments.
3. A user question.

Your task is to answer the question using the available information.

The event facts are predictions produced by an event tracking model. Each fact is labeled with either [true] or [false].

Interpret these labels as follows:
- [true] indicates the fact is likely correct and can be used as reliable evidence.
- [false] indicates the fact is likely incorrect or uncertain and should not be trusted unless strongly supported by other information. When [false] is presented for one fact, the entire event sequence after this fact is not presented. Thus, you must infer the accurate evidence from the video frames.

When reasoning about the answer:
- Carefully check the visual evidence based on the given frames.
- Prefer facts in video segments labeled [true].
- Treat facts labeled [false] as low-confidence evidence.
- Ignore [false] facts if they conflict with reliable facts.
- If no reliable evidence exists, answer based on the most plausible interpretation of the video events.

Figure 14: Prompt for dialogue response generation when the GVR-based ET is used.

**Prompt for Template-based Caption Generation**

You are analyzing a short video clip of a person in a home environment.

Your task is to generate a caption describing the person's actions and the objects they interact with in the clip.

### Format Specification

- Describe object locations and spatial relations using declarative statements.
Example:
- The person is in the living room.
- The TV is on the TV stand.
- The remote control is next to the person.

- Describe object state changes (e.g., ON/OFF) with temporal clarity.
Example: The TV is OFF.
- The TV is ON.

- Describe the person's every action step-by-step in chronological order.
Example:
- He walks to the TV.
- He grabs the remote control.

### Strict Rules
- Preserve the temporal order of actions as they occur in the video.
- Avoid background or irrelevant objects that are not directly involved in the actions.

Figure 15: Prompt for a template-based caption generation.

**Prompt for Refined Caption Generation**

You are analyzing a short video clip of a person in a home environment.

Your task is to generate a caption describing the person's actions and the objects they interact with in the clip.

### Strict Rules
- Preserve the temporal order of actions as they occur in the video.
- Avoid background or irrelevant objects that are not directly involved in the actions.

Figure 16: Prompt for a refined caption generation.

**Refined Caption Creation**

Please summarize the following text without adding any extra information: {template_caption}

Figure 17: Refined caption creation.

**Prompt for Fact Corruption**

You will corrupt the following fact by changing exactly ONE element.

Rules:
- The corrupted fact must remain grammatically valid and present plausible state/action inside home environments.
- Change or add one or more component (verb, object, or location).
- "person" in the fact MUST NOT be changed to something else.
- "walk(person)" and "walk_to(person, location)" MUST BE changed entirely with new fact with a verb, object, and location.
- Do not change "attached_to(object, person)" with "attached_to(object, dog)" or "attached_to(object, cat)".
- Do not simply change verbs, objects, locations with their synonyms.
- Output only the corrupted fact.

Examples

Input -> Output
grab(person, cupcake, from kitchen table) -> put(person, wineglass, from kitchen table)
grab(person, milk carton, from kitchen table) -> grab(person, milk carton, from refrigerator)
inside(person, kitchen) -> inside(person, bedroom)
walk(person) -> sit(person, chair)
walk_to(person, salmon) -> grab(person, milk carton)
attached_to(salmon, person) -> attached_to(wineglass, person)

Figure 18: Prompt for corrupting facts.

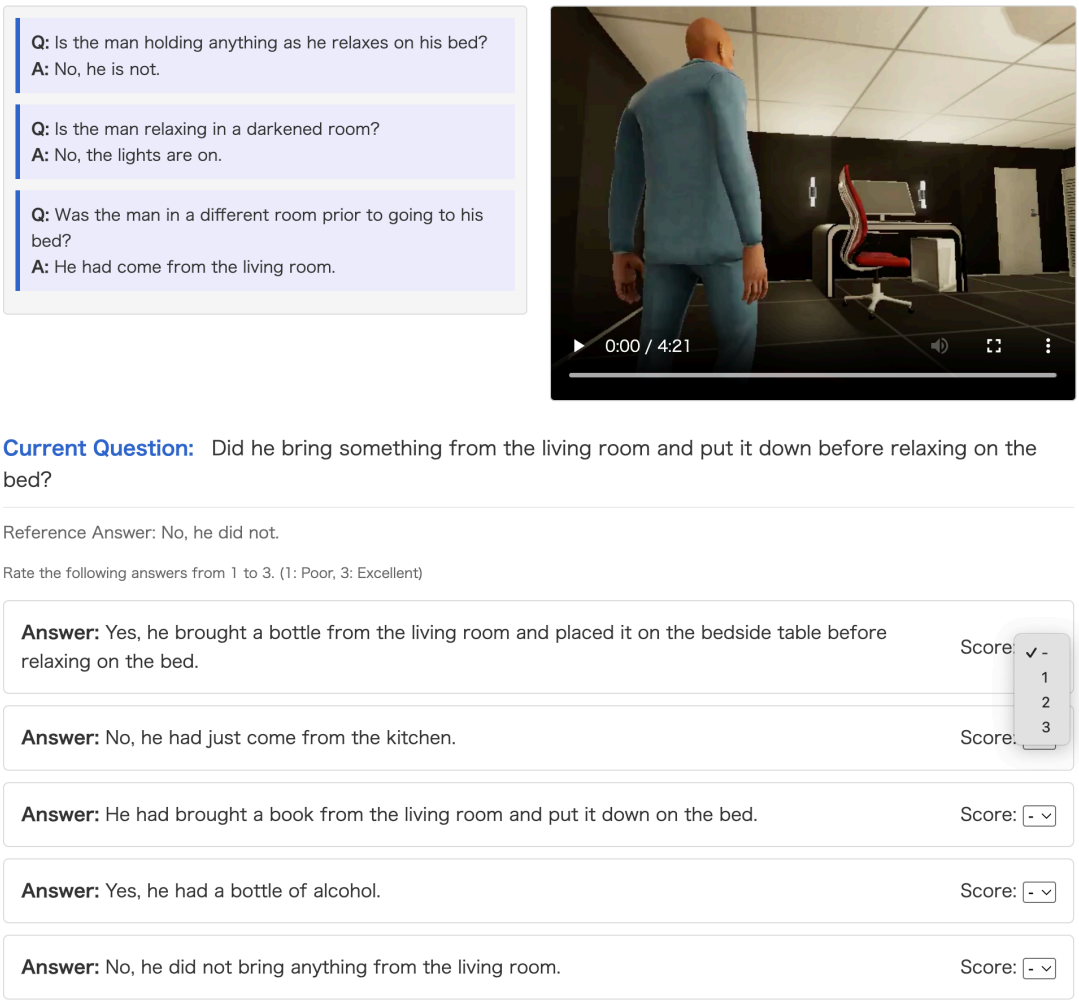



Figure 19: Human evaluation interface.