EGOEASY BENCHMARK: EVALUATING AUTOMATED ANNOTATION OF EGOCENTRIC WORKFLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Egocentric videos provide a scalable source of real-world activity data for embodied learning, yet converting unedited footage into reliable supervision remains a major bottleneck. Existing benchmarks evaluate only isolated tasks, query-driven segments, or predefined perception outputs, failing to assess whether a system can recover complete, evidence-grounded workflows while limiting unsupported claims. To address this gap, we introduce EgoEasy Benchmark (EEB), a comprehensive benchmark and evaluation suite for query-free annotation of egocentric workflows. EEB comprises 472 videos totaling 38.15 hours (2,891 episodes across 10 environmental and 13 workflow families), capturing rich transitions, navigation, interruptions, and unidentifiable intervals. We establish a unified annotation schema and evaluation protocol that decouples reference-content recovery from evidential reliability via prediction-side semantic canonicalization, evidence-aware adjudication, and capacity-constrained matching. Three complementary dashboards evaluate hierarchical workflow structure, fine-grained propositions, and annotation reliability. Evaluation across seven video-language model baselines reveals that current systems remain weak on complete temporal coverage, role grounding, and evidential calibration, highlighting the need for EEB in advancing robust video-based supervision.

1 INTRODUCTION

Egocentric video records human activity from a first-person perspective and offers scalable access to real-world experience for embodied learning. EgoMimic (Kareer et al., 2025) and EgoHumanoid (Shi et al., 2026) combine human demonstrations with robot data to support manipulation and loco-manipulation, respectively. However, turning these recordings into reliable supervision still requires substantial manual effort to locate actions in time and constrain descriptions to observable evidence. This bottleneck motivates automated annotation methods that reduce routine labeling effort and reserve human attention for specialized review.

Existing video benchmarks evaluate important components of this problem, but typically focus on isolated domains. Egocentric datasets provide rich supervision for perception and reasoning tasks such as action recognition, temporal localization, forecasting, and question answering (Damen et al., 2022; Grauman et al., 2022). Long-video benchmarks extend evaluation to broader temporal understanding, but are predominantly driven by predefined questions or task-specific targets (Mangalam et al., 2023; Chandrasegaran et al., 2024; Fu et al., 2025). These settings do not require a system to recover the complete event structure of an untrimmed recording.

Prior work studies dense and hierarchical video description, including temporal event localization and captioning (Krishna et al., 2017; Fujita et al., 2020; Islam et al., 2024), as well as open-ended event-level evaluation (Liu et al., 2024a). These methods evaluate event discovery or description at selected temporal levels, but they do not jointly require a complete typed timeline and evidence-aware assessment of generated claims. A system may correctly describe several events while omitting other parts of the workflow, or introduce plausible details that the recording cannot confirm.

The difficulty therefore lies not only in recognizing what is happening, but also in determining which conclusions can be safely drawn. In a first-person workflow, an object may be held without being used, and movement between work areas often separates otherwise similar operations. Furthermore, complete annotation requires recovering brief transitions that are easily omitted and evidence that

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

may be distributed throughout the recording. Consequently, a vision system may produce fluent, locally accurate descriptions while failing to account for the structural transitions that connect one operation to the next. Conversely, hallucinating an unverified tool role compromises supervision quality despite high temporal coverage. We therefore ask: can an unprompted system produce a sufficiently complete and evidence-faithful account of an entire workflow without being told which specific events to find?

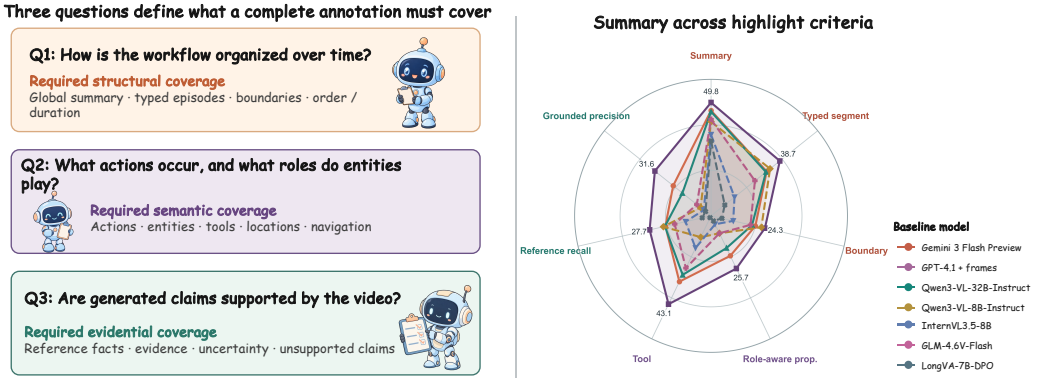

Figure 1: The EEB evaluation target and seven selected video-language model baselines.

To bridge this gap, we introduce EEB, a comprehensive benchmark and evaluation suite for query-free annotation of egocentric workflows. As shown in Figure 1, we operationalize complete annotation through three requirements: its typed episodes account for the full recording, it recovers the explicitly annotated reference facts, and generated claims receive an adjudicated support status that separates supported, partial, contradicted or unsupported, unresolved, duplicate, and masked content. This definition keeps reference-fact coverage separate from the evidential status of generated detail. Our contributions are:

- **A task-rich benchmark for diverse egocentric workflows.** EEB spans 472 unscripted videos (38.15 hours) and 2,891 contiguous episodes across 10 environmental and 13 workflow families. By naturally incorporating navigation, interruptions, idle periods, and unidentifiable intervals, it pushes evaluation far beyond short, single-location manipulation clips.
- **An auditable annotation evaluation protocol.** Utilizing a unified schema for temporal boundaries, episode types, and open-vocabulary proposition roles, our protocol couples semantic canonicalization with evidence-aware adjudication and deterministic capacity-constrained matching. This design prevents duplicate credit inflation and preserves the full provenance of every match, omission, and contradiction for audit.
- **Three capability-specific dashboards.** We organize evaluation into Hierarchy, Fine-grained, and Annotation Reliability dashboards. These components decouple global workflow structure from local event grounding and evidential verification, providing a diagnostic toolbox for developing reliable embodied video supervision.

2 Related Work

2.1 Egocentric Video Datasets and Hierarchical Annotation

Research on egocentric video has evolved from domain-specific action analysis toward comprehensive representations of daily and skilled activities. EPIC-KITCHENS-100 (Damen et al., 2022) established a large-scale benchmark for action recognition and anticipation in unscripted kitchen activities, while Ego4D (Grauman et al., 2022) broadened the scope to tasks including episodic memory, hand–object interaction, social interaction, and forecasting. Ego-Exo4D (Grauman et al., 2024) further adds synchronized egocentric and exocentric observations of skilled activities, while VISOR (Darkhalil et al., 2022) and HD-EPIC (Perrett et al., 2025) provide increasingly fine-grained annotations of objects, interactions, actions, and scene context. However, their evaluation protocols

primarily target predefined perception and reasoning tasks, falling short of measuring complete, unscripted workflow execution over extended horizons.

To capture higher-level task structure, recent efforts have turned toward language and relational representations. Frameworks like Goal-Step (Song et al., 2023) and Action Scene Graphs (Rodin et al., 2024) organize activities through hierarchical goals and structured relational graphs. Meanwhile, Video ReCap (Islam et al., 2024) specifically explores multi-level hierarchical video captioning for long-form recordings, while Exo2EgoDVC (Ohkawa et al., 2025) studies cross-view transfer from exocentric instructional videos to egocentric procedural dense captioning. Despite their structural insights, these approaches do not require a complete typed temporal partition that accounts for real-world complexities such as navigation transitions, interruptions, idle periods, and locally unidentifiable intervals.

Concurrently, the rapid growth of embodied AI has heightened the demand for human egocentric data in robot policy learning and world modeling (Kareer et al., 2025; Shi et al., 2026; Li et al., 2026a; Kim et al., 2026). These studies demonstrate the growing value of scalable human demonstrations, but they do not provide an evaluation protocol for determining whether automatically generated workflow annotations are complete, faithful, and robust to visual ambiguity. EEB is designed to address this gap.

2.2 Dense Event Discovery and Long-Video Evaluation

Dense video captioning (DVC) jointly localizes and describes multiple events in untrimmed video. ActivityNet Captions (Krishna et al., 2017) established large-scale evaluation of temporal event localization and description, while Vid2Seq (Yang et al., 2023) formulates dense event captioning as sequence generation by jointly predicting temporal boundaries and textual descriptions. To improve evaluation methodology, SODA (Fujita et al., 2020) introduced monotonic matching formulations to account for event order and redundancy, whereas E.T. Bench (Liu et al., 2024a) introduced a comprehensive benchmark suite for open-ended event evaluation. While these methods evaluate detected event sets, they do not require a complete, mutually exclusive typed partition of the recording.

Complementarily, long-video benchmarks examine extended temporal dependencies through resource suites such as EgoSchema (Mangalam et al., 2023), MMEgo (Ye et al., 2025), and HourVideo (Chandrasegaran et al., 2024), alongside broader long-context video evaluations (Wu et al., 2024; Fang et al., 2024; Fu et al., 2025; Zhou et al., 2025; Ma et al., 2026). These benchmarks are predominantly question- or task-conditioned, specifying in advance which aspect of the video should be recovered rather than requiring open-world discovery of the complete workflow. Furthermore, robust evaluation requires mitigating static-frame shortcuts where models exploit spatial or linguistic biases rather than genuine temporal dynamics (Buch et al., 2022; Liu et al., 2024b).

Beyond temporal localization, recent studies focus on fine-grained factual alignment and reliability. FactVC (Liu & Wan, 2023) demonstrates that conventional metrics correlate poorly with human factuality judgments. ARGUS (Rawal et al., 2025) jointly evaluates hallucinations and omissions in free-form video captions, while CaReBench (Xu et al., 2026) evaluates fine-grained captioning and retrieval with separate spatial and temporal annotations. Other efforts study precise linguistic oversight (Lin et al., 2026) and separate format correctness from content correctness in controllable video captioning (Li et al., 2026b). Existing semantic and model-based evaluation methods also motivate proposition-level comparison, but they generally evaluate individual captions or selected information units rather than temporally aligned claims over a complete workflow (Anderson et al., 2016; Zheng et al., 2023). EEB bridges this gap by integrating semantic canonicalization, evidence-aware adjudication, and capacity-constrained matching into a single evaluation suite, jointly measuring global structure, local propositions, and evidential support.

3 EgoEasy Benchmark (EEB): Benchmark Design

EgoEasy Benchmark (EEB) is a human-annotated benchmark for query-free annotation of complete egocentric workflows. It contains 472 videos recorded from the wearer's perspective, spanning 10 environmental families and 13 workflow families, including household activities, hotel service, public-facility maintenance, outdoor work, and tabletop assembly. Unlike fixed-camera datasets centered on isolated actions, these recordings capture naturally executed activities from the first-



Figure 2: Representative frames from the ten environment families in EEB.

person perspective, where task execution, navigation, object interaction, interruptions, and idle periods may occur within the same recording. Figure 2 shows representative examples from the ten environmental families.

Our benchmark requires a system to recover the complete observable workflow without being given a video-specific query or candidate vocabulary. In particular, systems must identify the activities performed over time, distinguish task execution from navigation and other intervals, bind actions to the correct objects and tools, and avoid introducing plausible but unobserved events. This setting is challenging because relevant evidence can be brief, distributed across the recording, and visually ambiguous. For example, the same object may be merely visible, carried, temporarily held, or functionally used, while similar actions may involve different objects or tools across environments. In addition, Figure 3 quantifies three aspects of the difficulty of our benchmark. Videos contain a median of four episodes and eight evidence windows, indicating that complete annotation typically requires multiple temporally localized predictions. Among 994 navigation episodes, 64% last fewer than six seconds, making short transitions easy to overlook or temporally mislocalize. Moreover, in 88% of videos, the span between the first and last evidence-window midpoints covers at least 75% of the recording. Evidence is therefore not concentrated in a short segment but is often distributed across most of the video. Together, these statistics show that successful annotation requires coverage of multiple events, accurate localization of brief transitions, and inspection of evidence throughout the recording.

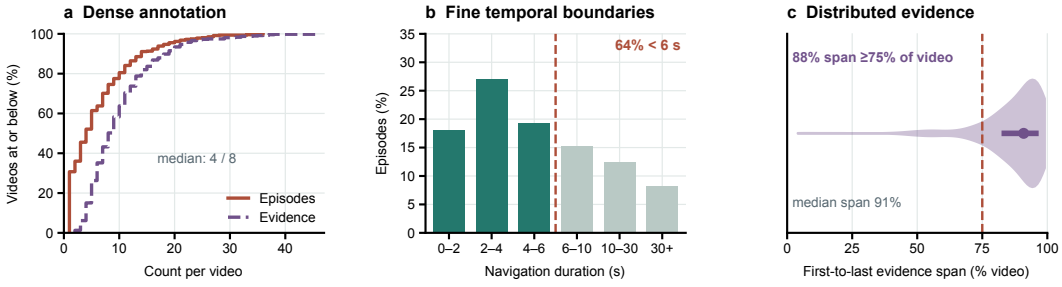

Figure 3: Structural difficulty of complete workflow annotation.

Table 1 situates EEB relative to representative benchmarks and evaluation protocols. EEB combines whole-video annotation, a typed temporal hierarchy, explicit action–object–tool relations, reference coverage, and evidence-aware evaluation within a single query-free benchmark.

Our ground-truth annotations are produced by trained professional annotators and independently reviewed before being finalized. Annotators inspect each recording from beginning to end and construct the reference in four connected steps:

1. **Identify the global workflow:** determine the primary activity, major stages, setting, completion state, and necessary temporal order relations.

2. **Segment the episode timeline:** partition the full recording into half-open intervals $[t_s, t_e)$ labeled as `task`, `navigation_transition`, `idle`, or `cannot_identify`.
3. **Annotate atomic propositions:** record observable actions together with their affected entities, tools, locations, carried objects, navigation endpoints, and temporal qualifiers.
4. **Attach evidence and evaluation status:** localize supporting moments or windows and assign each proposition a versioned `required`, `optional`, `forbidden`, or `ambiguous` status.

This annotation process produces a structured reference that captures both what occurs in the video and where the supporting visual evidence appears. Additional details on dataset scope, interval definitions, and annotation policy are provided in Appendix B.

Table 1: Comparison of representative benchmark tasks and evaluation protocols. Here, *partial* denotes a related but non-equivalent capability.

Benchmark	Ego	Whole-video annotation	Typed hierarchy	Role relations	Reference coverage	Evidence-aware evaluation
EPIC-KITCHENS-100 / Ego4D / VISOR	✓	–	partial	partial	–	–
Goal-Step / HD-EPIC / Action Scene Graphs	✓	–	partial	✓	–	–
Video ReCap / Exo2EgoDVC / Precise Video Language	partial	partial	partial	partial	partial	partial
EgoSchema / MMEgo / HourVideo	✓	–	–	–	–	–
ActivityNet Captions / SODA / E.T. Bench	–	partial	–	partial	partial	–
ARGUS / CaReBench / IF-VidCap	–	partial	–	partial	✓	partial
EEB (ours)	✓	✓	✓	✓	✓	✓

4 Evaluation Protocol

EEB is split into a 48-video public development set (10.2% of the corpus) and a 424-video hidden-reference test set. The development set is sampled to cover all 10 environmental families, all 13 workflow families, all four episode types, and varied recording conditions. To support pipeline debugging and analysis, complete human references and evaluator diagnostics will be released for the development set. Given only an unscripted video and the benchmark specification, an automated system must generate the unified structured annotation. Figure 4 illustrates this target for a 300-second hotel-service recording, where sampled observations are aligned to a gapless ten-episode reference timeline containing task, navigation, idle, and locally unidentifiable intervals.

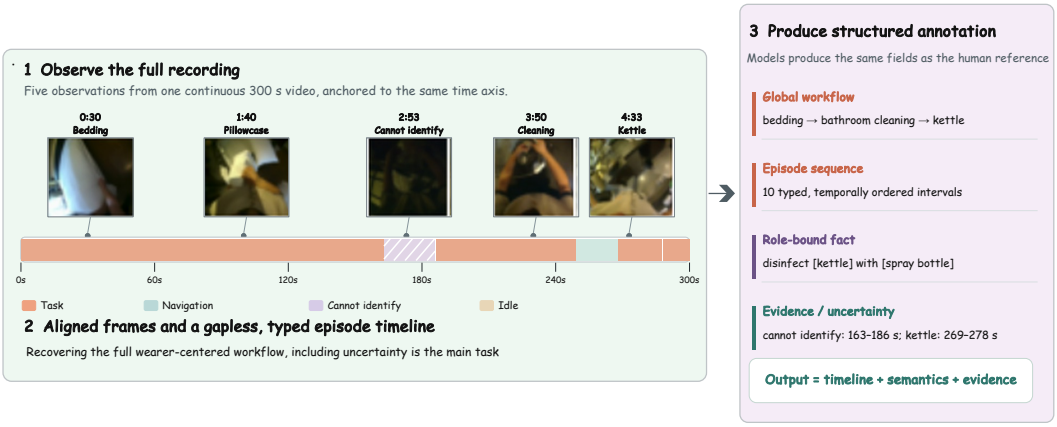

Figure 4: Annotation sample from a 300-second hotel-service recording

4.1 EVALUATOR ARCHITECTURE

The EEB evaluator is designed to systematically verify results from the three fundamental questions posed in Figure 1 by comparing predicted and reference structures across temporal, semantic, and evidential dimensions. It first validates structural schema compliance, logging timestamp errors, illegal gaps, overlapping boundaries, and broken entity IDs. Validated episode predictions are chronologically aligned to reference episodes via temporal overlap and episode type compatibility. Within aligned pairs, prediction-side canonicalization maps open-vocabulary descriptions into canonical role-bound propositions (e.g., actions, entities, tools, locations, and origins/destinations). Furthermore, to prevent credit inflation from paraphrased or repetitive predictions, the evaluator applies a deterministic capacity-constrained assignment, ensuring that every predicted claim and reference fact participates in at most one valid match.

Evaluation outputs then are organized into three capability-driven dashboards:

- **Hierarchy Dashboard:** Evaluates global workflow discovery, complete episode segmentation, and temporal ordering.
- **Fine-grained Dashboard:** Evaluates localized action-role binding, entity grounding, and reference fact recovery.
- **Annotation Reliability Dashboard:** Evaluates the evidential grounding of generated claims, isolating confirmed hallucinations, duplicates, and unverified assertions.

4.2 HIERARCHY DASHBOARD: WORKFLOW RECOVERY

The Hierarchy dashboard assesses whether a system captures both the top-level workflow logic and the complete temporal execution timeline. Global workflow summaries are parsed into high-level propositions and evaluated under our canonical proposition matching scheme. Temporal layout is evaluated across episode coverage, interval overlap, boundary precision, duration-weighted typing, and sequence order.

Predicted and reference episodes are aligned via order-preserving one-to-one matching, adopting the monotonic alignment principles of SODA (Fujita et al., 2020). Temporal overlap between a predicted interval p and reference interval g is measured by $\mathrm{tIoU}(p, g) = |p \cap g|/|p \cup g|$. The primary structural metric is **Strict Typed Episode** F_1. Under this metric, a predicted episode receives credit if and only if its predicted type matches the reference, $\mathrm{tIoU}(p, g) \geq 0.70$, and both temporal endpoint errors fall within an adaptive tolerance window $\epsilon(g)$:

$$\epsilon(g) = \max(0.5, \min(2.0, 0.2|g|)) \quad \text{(in seconds)}. \tag{1}$$

This dual constraint prevents oversized prediction intervals from claiming recovery credit simply by spanning across true action boundaries. In addition, we also report Gap Rate (unannotated time ratio), Boundary F_1, duration-weighted Type F_1, normalized Segmental Edit distance (Abu Farha & Gall, 2019), and required-order recovery metrics. Detailed definitions are provided in Appendix A.2.1.

4.3 FINE-GRAINED DASHBOARD: EVENT GROUNDING

The Fine-grained dashboard measures whether localized predictions correctly preserve the fine-grained semantic roles observed during execution. Candidate and reference episodes are paired sequentially; semantic evaluation is activated only when episode types match and $\mathrm{tIoU}(p, g) \geq 0.30$. This threshold enforces loose temporal localization without double-penalizing tight boundary precision already measured in the Hierarchy dashboard. Global summary propositions are evaluated video-wide. Following proposition-based semantic metrics like SPICE (Anderson et al., 2016), open-vocabulary claims are decomposed into typed relational triples rather than surface n-grams. Candidate claims are adjudicated as *entailed*, *partially supported*, *contradicted*, *supported extra*, *unsupported extra*, *unresolved extra*, *duplicate*, or *masked*.

The primary proposition scoring assigns 1.0 credit to fully supported matches, 0.5 credit to partially supported matches, and 0 otherwise. For each applicable video v, let c_i denote candidate-side credit and r_j reference-side credit. We compute:

$$P_v = \frac{\sum_i c_i}{N_{\text{cand},v}}, \qquad R_{\text{ref},v} = \frac{\sum_j r_j}{N_{\text{ref},v}}, \qquad F_{1,v} = \frac{2P_v R_{\text{ref},v}}{P_v + R_{\text{ref},v}}. \tag{2}$$

If $P_v = R_{\text{ref},v} = 0$, we define $F_{1,v} = 0$. Dataset-level Precision, Reference Fact Recall, and F_1 are independently video-macro averaged across applicable videos, so macro F_1 need not equal the harmonic mean of the displayed macro Precision and Recall. Here, $N_{\text{ref},v}$ contains only required reference facts; optional context may receive candidate-side precision credit without entering the recall denominator.

For each temporally and structurally eligible pair (i, j) confirmed by bidirectional semantic adjudication, let s^c_{ij} and s^r_{ij} denote candidate support and reference coverage, respectively, each taking values in $\{1, 0.5, 0\}$ for full, partial, and unsupported/missed decisions. The pair credit is

$$w_{ij} = \min(s^c_{ij}, s^r_{ij}).$$

Thus, full credit requires an entailed candidate and a covered reference, while any partial decision caps credit at 0.5. Pairs are then allocated by

$$X^* = \arg\max_X \sum_{(i,j)\in E} \tilde{w}_{ij} x_{ij} \quad \text{s.t.} \quad \sum_j x_{ij} \le 1,\ \sum_i x_{ij} \le 1,\ x_{ij} \in \{0, 1\}, \tag{3}$$

where $\tilde{w}_{ij}$ adds only a deterministic secondary preference for required references. Final Precision, Recall, and F_1 use the original w_{ij}; unmatched duplicates receive zero precision credit.

4.4 Annotation Reliability Dashboard

The Annotation Reliability dashboard evaluates the faithfulness of model-generated annotations, separating reference fact coverage from the evidential validity of generated claims. Grounded Precision and Reference Fact Recall reuse the corresponding candidate- and reference-side quantities from the Fine-grained dashboard and are repeated here so that content coverage can be interpreted jointly with hallucination, review, and duplicate rates. Unrecovered reference facts lower **Reference Fact Recall**, whereas confirmed contradictions and unsupported assertions drive the **Confirmed Hallucination Rate**.

Let N_C denote the total non-masked candidate claims, N_H the confirmed contradicted or unsupported claims, N_U the unresolved claims requiring manual audit, and N_D the unallocated duplicates. We define:

$$\text{HallucinationRate} = \frac{N_H}{N_C}, \qquad \text{ReviewRate} = \frac{N_U}{N_C}, \qquad \text{DuplicateRate} = \frac{N_D}{N_C}. \tag{4}$$

We additionally report the conservative upper unsupported-fact rate $\text{UFR}_U = (N_H + N_U)/N_C$, which combines confirmed unsupported claims and unresolved extras. Reference Fact Recall and Confirmed Hallucination Rate serve as the primary reliability metrics. Review Rate identifies claims that remain unresolved under the available evidence, while Duplicate Rate isolates redundant output and UFR_U provides a conservative unsupported-content diagnostic. Unidentifiable intervals (`cannot_identify`) are evaluated distinctly: predicting ambiguity receives credit only if visual evidence is genuinely insufficient. Speculating over clear observations or over-predicting ambiguity on clear intervals is penalized as a false positive. Abstention may lower hallucination risk, but it cannot remove required reference facts from the recall denominator.

All metrics are first computed per video and then independently video-macro averaged across applicable videos. Metrics with undefined denominators are marked as N/A and excluded from the corresponding macro average, with micro-averaged variants provided in Appendix A.2.4 as secondary diagnostics.

5 Experiments

Before evaluating model baselines, we first assess whether the benchmark target can be independently reproduced by trained human annotators under the same annotation specification and evaluator, while providing a human reference point for interpreting current model performance. Five

Table 2: Independent human reproducibility under the EEB evaluator on the 48-video public development set. Values are video-macro percentages averaged across five trained annotators

	Summary F_1	Strict Episode F_1	Boundary F_1	Role-aware Prop. F_1	Reference Fact Recall
Mean of five annotators	98.3	90.2	97.6	96.3	95.5

trained annotators, each blind to the frozen reference, independently annotated all 48 videos in the public development set directly from the raw videos using the same version specification and annotation manual. Each annotation was frozen before scoring and evaluated against the frozen reference using the same temporal alignment, proposition matching, and reliability protocol as model outputs. The high scores show that trained annotators can independently reproduce the benchmark target. Residual disagreement can reflect boundary ambiguity and open-vocabulary variation. Because human and model results use different splits, these values serve as a human reference point rather than a directly comparable upper bound.

5.1 Baselines and Protocol

Our evaluation includes seven representative video-language models spanning commercial API-based systems and open-weight architectures: Gemini 3 Flash Preview, GPT-4.1, Qwen3-VL-8B-Instruct (Qwen Team, 2025b), Qwen3-VL-32B-Instruct (Qwen Team, 2025a), InternVL3.5-8B (OpenGVLab, 2025), GLM-4.6V-Flash (Z.ai, 2026), and LongVA-7B-DPO (LMMs-Lab, 2024). All four baselines reported under staged generation—Qwen3-VL-8B, InternVL3.5-8B, GLM-4.6V-Flash, and LongVA-7B-DPO—were first tested with direct complete annotation but repeatedly truncated before completing the full schema. We therefore use staged complete annotation for these models to reduce per-call output length. Staging does not provide oracle event information: detail calls operate only on the model's own predicted intervals and cannot recover episodes omitted or mistyped during timeline generation. Direct and staged systems are therefore reported separately, and cross-group differences are not interpreted as causal effects of the generation protocol.

All models receive the same annotation requirements and target schema. Gemini processes full-duration 720p native video inputs (Google, 2026a;b), while GPT-4.1 receives uniformly sampled timestamped frames at high visual resolution (OpenAI, 2026). Qwen3-VL-32B produces the complete structured annotation directly. The remaining open-weight models use staged generation: global summaries and episode timelines are predicted from the complete video, followed by fine-grained facts and evidence generated only for the model-predicted intervals. Temporal metrics are computed deterministically over all 424 hidden-test videos, while all semantic comparisons are adjudicated directly using the same frozen Gemini-based judge. The frozen semantic judge was evaluated once on a disjoint held-out human-review sample containing 1,000 adjudication decisions sampled from 100 videos. Human reviewers confirmed 94.3% of the frozen judge decisions. Class-wise validation statistics are reported in Appendix C.1. Appendix C provides the complete input, generation, and adjudication settings.

5.2 Main Results

Table 3 presents results across the Hierarchy, Fine-grained, and Annotation Reliability dashboards. Across all evaluated baselines, complete temporal and semantic workflow recovery remains low. Strict Episode F_1 peaks at 15.1%, Boundary F_1 at 24.3%, Role-aware Proposition F_1 at 25.7%, and Reference Fact Recall at 27.7%. GPT-4.1 achieves the strongest semantic recovery, while Qwen3-VL-8B under staged generation yields the highest Strict Episode F_1 but substantially lower proposition precision. Summary F_1 values of 33.3–49.8% therefore do not translate into complete event recovery. The cross-dashboard pattern is also informative: low Gap Rate can coexist with incorrect boundaries and episode types, while recognizing objects and tools remains easier than recovering their role-bound relations and navigation context.

Table 3: Complete baseline results across the three evaluation dashboards. Entries are independently video-macro averaged percentages except Boundary MAE (seconds); ↓ denotes lower is better. Boundary MAE is computed over matched boundary pairs only.

	Direct complete annotation			Staged complete annotation			
Metric	Gemini 3 Flash	GPT-4.1 + frames	Qwen3-VL 32B	Qwen3-VL 8B	InternVL 3.5-8B	GLM 4.6V	LongVA 7B
Hierarchy (H)							
Summary F1	46.3	49.8	46.0	41.5	36.2	42.3	33.3
Strict Episode F1	12.2	14.0	10.7	15.1	1.2	4.2	0.9
Typed Segment F1	31.3	38.7	30.9	33.2	13.2	24.7	7.7
Untyped Segment F1	35.4	45.1	39.4	40.9	19.3	32.4	11.1
Boundary F1	20.0	24.3	18.5	22.7	10.0	17.8	4.9
Boundary MAE (s) ↓	0.60	0.60	0.62	0.71	0.72	0.60	0.65
Duration Type F1	55.2	55.1	56.2	47.9	35.7	51.8	23.3
Segmental Edit	58.5	58.6	66.3	54.8	39.4	61.5	17.8
Gap Rate ↓	0.0	0.3	0.1	2.8	11.2	3.7	37.4
Fine-grained Events (F)							
Role-aware Prop. Precision	21.3	31.6	16.1	5.9	4.1	8.0	3.1
Role-aware Prop. Recall	20.6	27.7	20.8	21.5	11.5	16.4	4.0
Role-aware Prop. F1	19.5	25.7	17.1	8.4	4.3	9.0	2.6
Exact-credit Prop. F1	15.0	18.7	13.3	7.0	3.2	7.2	0.1
Action F1	23.0	33.3	21.6	10.5	7.6	14.4	5.2
Entity F1	30.3	38.9	25.3	13.7	9.7	19.2	5.3
Tool F1	32.1	43.1	28.8	10.4	15.7	25.5	0.9
Room F1	32.6	37.9	24.8	11.4	8.7	15.2	1.0
Zone F1	25.1	16.5	16.3	8.5	7.3	12.6	0.8
Nav. Origin F1	24.2	18.1	18.4	17.7	7.1	22.1	0.6
Nav. Destination F1	25.3	21.2	19.9	17.5	10.8	27.3	0.6
Annotation Reliability (A)							
Grounded Precision	21.3	31.6	16.1	5.9	4.1	8.0	3.1
Reference Fact Recall	20.6	27.7	20.8	21.5	11.5	16.4	4.0
Confirmed Hallucination Rate ↓	11.5	11.8	14.1	23.4	13.2	22.7	12.8
Review Rate ↓	53.2	39.8	59.1	50.0	72.8	62.8	73.3
Duplicate Rate ↓	9.5	8.7	7.5	20.9	4.7	5.3	5.0
UFR upper diagnostic ↓	64.7	51.6	73.2	73.4	86.1	85.5	86.0

5.3 Discussion

The three dashboards expose failure modes hidden by a single captioning or QA score. Higher Summary F_1 than Strict Episode F_1 shows that coarse workflow recognition does not imply complete temporal decomposition; the observed pattern is consistent with brief transitions and local task switches being missed or merged into neighboring episodes. Likewise, object recognition is stronger than role-aware proposition recovery, indicating that identifying what is visible is easier than determining its functional role in the ongoing action. Staged generation does not remove these limitations because later detail calls depend entirely on the model-predicted timeline. Missing or mistyping an episode therefore propagates to all subsequent fact and evidence extraction. The high Review Rates further show that free-form annotation reliability cannot be reduced to a binary hallucination decision. Many generated claims are neither clearly contradicted nor safely supportable from the available evidence because they over-specify semantic roles, rely on weakly localized evidence, or occur in visually ambiguous intervals. Keeping these cases separate from confirmed hallucinations therefore estimates how much human review an automated annotation pipeline would still require.

6 Conclusion

We introduced EEB, a benchmark for query-free, whole-video annotation of egocentric workflows. EEB evaluates complete temporal structure, role-bound semantic recovery, and annotation reliability through three complementary dashboards. Across seven video-language model baselines, fluent summaries coexist with substantial temporal omissions, role-binding errors, and unsupported claims. These results show that reliable automated supervision requires joint progress in temporal decomposition, semantic grounding, evidence verification, and calibrated uncertainty.

AI USE STATEMENT

Generative AI tools were used solely for language-level editing, including checks and improvements of grammar, wording, clarity, and readability, as well as for presentation-level adjustments to the layout and scaling of figures and tables. These tools were not used to develop or modify the research methodology, formulate hypotheses or mathematical claims, design experiments, implement methods, process or analyze data, or interpret experimental results. The underlying scientific content, data, results, and conclusions were not altered by AI-assisted editing.

All AI-assisted modifications were reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including all text and visual materials.

REFERENCES

Yazan Abu Farha and Juergen Gall. MS-TCN: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3575–3584, 2019.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, 2016.

Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the "video" in video-language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Keshigeyan Chandrasegaran, Agrim Gupta, Lea M. Hadzic, Taran Kota, Jimming He, Cristobal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Li Fei-Fei. HourVideo: 1-hour video-language understanding. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024. doi: 10.52202/079017-1684. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5f2809607f692d79a01c05c43d702883-Abstract-Datasets_and_Benchmarks_Track.html.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, et al. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *International Journal of Computer Vision*, 130:33–55, 2022. doi: 10.1007/s11263-021-01531-2.

Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. EPIC-KITCHENS VISOR benchmark: Video segmentations and object relations. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 35, 2022.

Xinyu Fang, Kangrui Mao, Haodong Duan, Xiangyu Zhao, Yining Li, Dahua Lin, and Kai Chen. MMBench-Video: A long-form multi-shot benchmark for holistic video understanding. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024.

Chaoyou Fu et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24108–24118, 2025.

Soichiro Fujita, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. SODA: Story oriented dense video captioning evaluation framework. In *European Conference on Computer Vision*, pp. 517–531, 2020.

Google. Gemini 3 Flash Preview: Model documentation. Google AI for Developers, 2026a. URL https://ai.google.dev/gemini-api/docs/models/gemini-3-flash-preview. Accessed September 7, 2026.

Google. Video understanding in the Gemini API. Google AI for Developers, 2026b. URL https://ai.google.dev/gemini-api/docs/video-understanding. Accessed September 20, 2026.

Kristen Grauman, Andrew Westbury, Eugene Byrne, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, et al. Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19383–19400, 2024.

Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius. Video recap: Recursive captioning of hour-long videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18198–18208, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Islam_Video_ReCap_Recursive_Captioning_of_Hour-Long_Videos_CVPR_2024_paper.html.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. EgoMimic: Scaling imitation learning via egocentric video. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13226–13233, 2025. doi: 10.1109/ICRA55743.2025.11127989. URL https://egomimic.github.io/.

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

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 706–715, 2017.

Baoyu Li, Xinchen Yin, Mengying Lin, Yixin Zhang, and Danfei Xu. EgoWAM: World action models beyond pixels with in-the-wild egocentric human data. *arXiv preprint arXiv:2607.08436*, 2026a. URL https://arxiv.org/abs/2607.08436.

Shihao Li, Yuanxing Zhang, Jiangtao Wu, Zhide Lei, Yiwen He, Runzhe Wen, Chenxi Liao, Chengkang Jiang, An Ping, Shuo Gao, Suhan Wang, Zhaozhou Bian, Zijun Zhou, Jingyi Xie, Jiayi Zhou, Jing Wang, Yifan Yao, Weihao Xie, Yingshui Tan, Yanghai Wang, Qianqian Xie, Zhaoxiang Zhang, and Jiaheng Liu. IF-VidCap: Can video caption models follow instructions? In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/27befed4547edcb4bdeacef9472cadee-Abstract-Conference.html.

Zhiqiu Lin, Siyuan Cen, Chancharik Mitra, Isaac Li, Yuhan Huang, Yu Tong Tiffany Ling, Hewei Wang, Irene Pi, Shihang Zhu, Yili Han, Yilun Du, and Deva Ramanan. Building a precise video language with human-AI oversight. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11334–11345, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Lin_Building_a_Precise_Video_Language_with_Human-AI_Oversight_CVPR_2026_paper.html.

Hui Liu and Xiaojun Wan. Models see hallucinations: Evaluating the factuality in video captioning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 11807–11823. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.723. URL https://aclanthology.org/2023.emnlp-main.723/.

Ye Liu, Zongyang Ma, Zhongang Qi, Yang Wu, Ying Shan, and Chang Wen Chen. E.T. Bench: Towards open-ended event-level video-language understanding. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024a. doi: 10.52202/079017-1009. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/38ace89a980ead30c6be2b46afc1a5bd-Abstract-Datasets_and_Benchmarks_Track.html.

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. TempCompass: Do video LLMs really understand videos? In *Findings of*

the Association for Computational Linguistics: ACL 2024, pp. 8731–8772. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.findings-acl.517. URL https://aclanthology.org/2024.findings-acl.517/.

LMMs-Lab. LongVA-7B-DPO: Model card. Hugging Face, 2024. URL https://huggingface.co/lmms-lab/LongVA-7B-DPO. Accessed September 23, 2026.

David Ma, Huaqing Yuan, Xingjian Wang, Qianbo Zang, Tianci Liu, Xinyang He, Yanbin Wei, Jiawei Guo, Jiahui Ni, Zhenzhu Yang, Meng Cao, Shanghaoran Quan, Yizhi Li, Wangchunshu Zhou, Jiaheng Liu, Wenhao Huang, Ge Zhang, Shiwen Ni, and Xiaojie Jin. ScaleLong: A multi-timescale benchmark for long video understanding. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fa1cfe4e956d85e016b1f8f49b189a0b-Abstract-Conference.html.

Karttikeya Mangalam, Raiymbek Akshkulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 36, 2023.

Takehiko Ohkawa, Takuma Yagi, Taichi Nishimura, Ryosuke Furuta, Atsushi Hashimoto, Yoshitaka Ushiku, and Yoichi Sato. Exo2EgoDVC: Dense video captioning of egocentric procedural activities using web instructional videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025. URL https://openaccess.thecvf.com/content/WACV2025/html/Ohkawa_Exo2EgoDVC_Dense_Video_Captioning_of_Egocentric_Procedural_Activities_using_Web_WACV_2025_paper.html.

OpenAI. GPT-4.1: Model documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-4.1. Accessed September 20, 2026.

OpenGVLab. InternVL3.5-8B: Model card. Hugging Face, 2025. URL https://huggingface.co/OpenGVLab/InternVL3_5-8B. Accessed September 23, 2026.

Toby Perrett, Ahmad Darkhalil, Saptarshi Sinha, Omar Emara, Sam Pollard, Kranti Kumar Parida, Kaiting Liu, Prajwal Gatti, Siddhant Bansal, Kevin Flanagan, Jacob Chalk, Zhifan Zhu, Rhodri Guerrier, Fahd Abdelazim, Bin Zhu, Davide Moltisanti, Michael Wray, Hazel Doughty, and Dima Damen. HD-EPIC: A highly-detailed egocentric video dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23901–23913, 2025.

Qwen Team. Qwen3-VL-32B-Instruct: Model card. Hugging Face, 2025a. URL https://huggingface.co/Qwen/Qwen3-VL-32B-Instruct. Accessed September 23, 2026.

Qwen Team. Qwen3-VL-8B-Instruct: Model card. Hugging Face, 2025b. URL https://huggingface.co/Qwen/Qwen3-VL-8B-Instruct. Accessed September 7, 2026.

Ruchit Rawal, Reza Shirkavand, Heng Huang, Gowthami Somepalli, and Tom Goldstein. ARGUS: Hallucination and omission evaluation in video-LLMs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20280–20290, 2025.

Ivan Rodin, Antonino Furnari, Kyle Min, Subarna Tripathi, and Giovanni Maria Farinella. Action scene graphs for long-form understanding of egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18622–18632, 2024.

Modi Shi, Shijia Peng, Jin Chen, Haoran Jiang, Tianyu Li, Ping Luo, Di Huang, Hongyang Li, and Li Chen. Unlocking in-the-wild loco-manipulation with robot-free egocentric demonstration. In *Proceedings of Robotics: Science and Systems*, 2026. doi: 10.15607/RSS.2026.XXII.204. URL https://www.roboticsproceedings.org/rss22/p204.html.

Yale Song, Eugene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4D Goal-Step: Toward hierarchical understanding of procedural activities. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 36, 2023. doi: 10.52202/075280-1688. URL https://papers.nips.cc/paper_files/paper/2023/hash/7a65606fa1a6849450550325832036e5-Abstract-Datasets_and_Benchmarks.html.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024.

Yifan Xu, Xinhao Li, Yichun Yang, Desen Meng, Rui Huang, and Limin Wang. CaReBench: A fine-grained benchmark for video captioning and retrieval. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5401bede0941e568fa42b7b5910fa72c-Abstract-Conference.html.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. 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*, pp. 10714–10726, 2023.

Hanrong Ye, Haotian Zhang, Erik Daxberger, Lin Chen, Zongyu Lin, Yanghao Li, Bowen Zhang, Haoxuan You, Dan Xu, Zhe Gan, Jiasen Lu, and Yinfei Yang. MMEgo: Towards building egocentric multimodal LLMs for video QA. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b29a95e7d9f1e1a6dfc567b556733744-Abstract-Conference.html.

Z.ai. GLM-4.6V-Flash: Model card. Hugging Face, 2026. URL https://huggingface.co/zai-org/GLM-4.6V-Flash. Accessed September 23, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 36, 2023.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025.

A Evaluation Interface and Metric Details

This appendix specifies the complete interface between an annotation system and the EEB evaluator and provides full definitions of the diagnostic metrics used by the three evaluation dashboards. Section 4 presents the primary evaluation protocol and headline measures; the details below provide the schema, generation interface, temporal diagnostics, proposition-level scoring, reliability accounting, and aggregation rules required for complete reproduction.

A.1 Schema Setup and Evaluation Interface

The released JSON Schema defines the common interface between an annotation system and the evaluator. A system receives the complete video, its duration, a generic annotation instruction, and the schema; it does not receive video-specific questions, event boundaries, or candidate vocabularies. The resulting object contains global workflow propositions and a complete, ordered episode timeline. Table 4 lists only the fields that define benchmark content; routine JSON validation metadata remains in the released machine-readable schema.

Using the benchmark requires four operational steps:

1. Generate one annotation object for each test video;
2. Validate field types, IDs, timestamps, references, and timeline coverage against the released schema and cross-field rules;
3. Compute deterministic temporal matches and directly adjudicate open-vocabulary semantic claims with the frozen semantic judge against the hidden human reference;
4. Aggregate the resulting Hierarchy, Fine-grained, and Annotation Reliability measures at both the video and dataset levels.

The 48 public development videos expose complete references and evaluator outputs for system implementation, debugging, and analysis. The remaining 424 test references stay strictly hidden, and official scores are produced via the frozen evaluator so that future model submissions are judged under identical schemas, matching rules, and semantic-adjudication protocols.

Table 4: Benchmark-relevant fields in the EEB annotation schema.

Annotation level	Schema content	Primary evaluation view
Video	`video_id`, `duration_sec`	Record identity and temporal validation
Global summary	`global_summary_facts`	Hierarchy: workflow-summary recovery
Episode	`episode_id`, start and end time, `episode_type`, facts, navigation, and evidence	Hierarchy: temporal coverage, order, boundaries, and episode types
Fact	Predicate; affected entity, destination, tool, room, and zone roles; polarity and certainty	Fine-grained: proposition and role grounding
Navigation	Applicability, origin, destination, and carried objects	Fine-grained: navigation grounding
Evidence	Temporal evidence window and the fact identifiers it supports	Reliability: claim support and overclaim analysis

Complete Candidate Annotation Schema. The released JSON Schema below specifies every required field and allowed episode type. Cross-field checks additionally validate references and temporal consistency.

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "EEB Candidate Annotation v3",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "schema_version": {"type": "string", "const": "candidate_video_annotation_v3"},
    "video_id": {"type": "string", "minLength": 1},
    "submission_profile": {"type": "string", "const": "reliable_annotation"},
    "submission_status": {"type": "string", "const": "annotated"},
    "duration_sec": {"type": "number", "exclusiveMinimum": 0},
    "global_summary_facts": {
      "type": "array",
      "items": {"type": "string", "minLength": 1},
      "minItems": 1,
      "maxItems": 3,
      "uniqueItems": true
    },
    "episodes": {
      "type": "array",
      "minItems": 1,
      "items": {"$ref": "#/$defs/episode"}
    }
  },
  "required": [
    "schema_version", "video_id", "submission_profile", "submission_status",
    "duration_sec", "global_summary_facts", "episodes"
  ],
  "$defs": {
    "stringList": {
      "type": "array",
      "items": {"type": "string", "minLength": 1},
      "uniqueItems": true
    },
    "fact": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "candidate_fact_id": {"type": "string", "minLength": 1},
        "source_span": {"type": "string", "minLength": 1},
        "polarity": {"type": "string", "enum": ["affirmed", "negated"]},
        "certainty": {"type": "string", "enum": ["certain", "hedged", "uncertain"]},
        "predicate": {"type": "string"},
        "affected_entity": {"$ref": "#/$defs/stringList"},
```

```
        "destination": {"$ref": "#/$defs/stringList"},
        "tool": {"$ref": "#/$defs/stringList"},
        "room": {"type": "string"},
        "zone": {"$ref": "#/$defs/stringList"},
        "state_or_result": {"$ref": "#/$defs/stringList"}
      },
      "required": [
        "candidate_fact_id", "source_span", "polarity", "certainty",
        "predicate", "affected_entity", "destination", "tool", "room",
        "zone"
      ]
    },
    "navigation": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "applicable": {"type": "boolean"},
        "candidate_fact_id": {"type": "string"},
        "source_span": {"type": "string"},
        "from_room": {"type": "string"},
        "from_zone": {"type": "string"},
        "to_room": {"type": "string"},
        "to_zone": {"type": "string"},
        "carried_objects": {"$ref": "#/$defs/stringList"}
      },
      "required": [
        "applicable", "candidate_fact_id", "source_span", "from_room",
        "from_zone", "to_room", "to_zone", "carried_objects"
      ]
    },
    "evidence": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "start_sec": {"type": "number"},
        "end_sec_exclusive": {"type": "number"},
        "supports_fact_ids": {
          "type": "array",
          "items": {"type": "string", "minLength": 1},
          "minItems": 1,
          "uniqueItems": true
        }
      },
      "required": ["start_sec", "end_sec_exclusive", "supports_fact_ids"]
    },
    "episode": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "episode_id": {"type": "string", "minLength": 1},
        "source_span": {"type": "string", "minLength": 1},
        "episode_type": {
          "type": "string",
          "enum": ["task", "navigation_transition", "idle", "cannot_identify"]
        },
        "granularity": {"type": "string", "enum": ["coarse_episode", "fine_action"]},
        "uncertainty_reason": {"type": "string"},
        "start_sec": {"type": "number", "minimum": 0},
        "end_sec_exclusive": {"type": "number"},
        "facts": {"type": "array", "items": {"$ref": "#/$defs/fact"}},
        "navigation": {"$ref": "#/$defs/navigation"},
        "temporal_qualifiers": {"$ref": "#/$defs/stringList"},
        "evidence": {"type": "array", "items": {"$ref": "#/$defs/evidence"}}
      },
      "required": [
        "episode_id", "source_span", "episode_type", "granularity",
        "uncertainty_reason", "start_sec", "end_sec_exclusive", "facts",
        "navigation", "temporal_qualifiers", "evidence"
      ]
    }
  }
}
```

Frozen Direct-Generation System Prompts. The following are the complete system instructions used for direct generation, with Markdown title lines omitted in print. The Qwen3-VL direct prompt is included separately because its wording differs from the API prompt; its filename does not impose an 8B-specific restriction.

```
You are producing a complete candidate annotation for a general egocentric-video
benchmark. Inspect the supplied visual sequence from beginning to end and rely
only on evidence actually present in the supplied media. Audio may be absent; do
not infer speech or sound when it is not supplied. You cannot see gold
annotations, folder names, filenames, scenario labels, or task labels. The
opaque `video_id` is only an identifier and carries no semantic information.

Produce two levels of annotation:

1. one to three concise global summary facts describing the main workflow and
   major stages;
2. an exhaustive, ordered coarse episode timeline covering the complete clip.

Episode rules:

- Use `task` for sustained goal-directed manipulation, `navigation_transition`
  for meaningful movement between work areas, `idle` for observable inactivity,
  and `cannot_identify` only when the activity is genuinely not observable.
- Segment at meaningful sustained activity or state changes. Do not create a new
  episode for every hand motion, brief pause, camera shake, or object touch.
- Episodes must be chronological, non-overlapping, and collectively cover the
  supplied duration from 0 to the final second. Use half-open intervals
  `[start_sec, end_sec_exclusive)` and make adjacent boundaries equal.
- Timestamps are seconds from video start. Use the closest defensible boundary;
  do not invent frame precision.
- Describe what is visibly done, not the wearer's hidden intention. Do not claim
  that an entire room or task is complete unless the ending establishes it.
- Distinguish visibility, carrying, temporary holding, and functional use. An
  object that appears in a hand is not automatically a tool, and contact alone
  does not establish that the object was used successfully.
- Do not replace an unclear object, tool, room, target, state, or result with a
  confident guess. Preserve the supported action at broader specificity and use
  `hedged` or `uncertain` certainty for the affected proposition.
- Use empty strings/arrays for inapplicable fact slots. For
  `navigation_transition`, fill from/to locations and carried objects when
  visible. For other types, set `navigation.applicable=false`.
- A `cannot_identify` episode must include a short uncertainty reason such as
  `visibility`, `source_gap`, `identity_continuity`, or `intent_unknown`.
- Keep actions, affected entities, destinations, tools, locations and results as
  separate fact slots. `affected_entity` is what undergoes the action (floor in
  "mop the floor", cup in "put the cup on the table"); `destination` is where a
  moved entity ends up (table in that placement example). Do not use destination
  for the room traversed by the wearer; use navigation from/to fields instead.
  Avoid synonyms duplicated within the same list.
- Use one fact object per role-bound proposition. An episode may contain
  multiple facts when it includes distinct actions or results; keep each
  predicate and its arguments together in the same fact object.
- Set `polarity` to `affirmed` or `negated`, and set `certainty` to `certain`,
  `hedged`, or `uncertain` according to what the visible evidence supports.
- Evidence windows should be short representative intervals inside the episode,
  cite one or more fact IDs declared in that episode, and remain inside the
  parent episode. They are diagnostic and need not cover every fact. If an
  episode declares no facts and has no applicable navigation proposition, its
  `evidence` array must be empty; never cite an episode ID as a fact ID.
- Copy the opaque `video_id` and supplied `duration_sec` exactly. Use contiguous
  IDs in chronological order: `pred_ep_0001`, `pred_ep_0002`, and so on for
  episodes; `pred_fact_0001`, `pred_fact_0002`, and so on for propositions.
  Navigation propositions use the same globally unique fact-ID sequence.
- Every field required by the supplied schema must be present. Use empty strings
  and arrays where the schema requires a field that is not applicable; never use
  null and never add fields outside the schema.
- Set `schema_version` exactly to `candidate_video_annotation_v3`,
  `submission_profile` exactly to `reliable_annotation`, and
  `submission_status` exactly to `annotated`.

Return only JSON matching the supplied schema. Do not return scores, confidence
numbers, analysis, or references to these instructions.
```

```
Annotate the supplied first-person recording from beginning to end. The subject
is the camera wearer: recover the wearer's observable work, movement, waiting,
and uncertainty. Other people belong in the annotation only when they interact
with the wearer or visibly change the wearer's workflow. Do not independently
catalog bystanders, background objects, or activities merely visible in the
room. The filename, folder, and opaque video ID are not evidence of content.
Use only supplied visual evidence; do not infer unavailable audio or expected
steps of a familiar task.
```

```
Return one JSON object matching the supplied schema, with:

1. One to three concise `global_summary_facts` about the wearer's main observed
   workflow and major changes, without a frame-by-frame retelling.
2. A complete, chronological, non-overlapping **coarse** `episodes` timeline
   covering `[0, duration_sec)` exactly. Use `task` for sustained work,
   `navigation_transition` only for actual movement between meaningful work
   areas, `idle` for observable inactivity, and `cannot_identify` when visual
   evidence cannot identify the activity. A turn toward a nearby object, a
   camera movement, or another room appearing in view is not navigation.

Segment by sustained wearer activity, work area, or meaningful state change.
Do not create an episode for each frame, brief hand motion, object touch,
repeated wipe/stir, or arbitrary time interval. Keep repeated actions under
one sustained goal in one episode; put distinct observed steps in separate
facts within that episode. Do not replace an observable sequence with a vague
label such as `prepare`, `clean`, or `organize` when the action and target can
be named. A task episode must contain at least one concrete action fact.

For facts, describe each **distinct** observed action--entity--result relation
once. Do not emit multiple facts that merely paraphrase the same action on the
same target in the same interval. If an action recurs later or has a different
target or result, record that distinction. Bind the predicate to the entity
actually affected, the destination of any moved object, the tool functionally
used, the location, and a visible state change when supported. An object being
visible, held, or carried does not by itself establish tool use. Do not claim
a result, identity, precise object, or completed workflow beyond the visual
evidence; use broader wording and `hedged`/`uncertain` when necessary. Use
`negated` only for an observable non-occurrence, not for something merely
unseen. Use empty arrays/strings for inapplicable roles.

For navigation, set `navigation.applicable=true` and record only observed
from/to areas and visibly carried objects; for other episode types set it
false and leave its fields empty. Give `cannot_identify` a short evidence-based
`uncertainty_reason`. Use each `source_span` to briefly name what is visibly
supporting the episode or fact; a timestamp range alone is not a description.
Evidence windows are optional, short representative intervals within the
parent episode, citing fact IDs from that episode only. Do not produce one
window per frame or invent supporting evidence.

Copy the supplied `video_id` and `duration_sec` exactly. Use unique sequential
`pred_ep_0001`, `pred_ep_0002`, ... and globally unique sequential
`pred_fact_0001`, `pred_fact_0002`, ... IDs, including navigation IDs. The first
episode begins at 0, adjacent episode boundaries meet, and the final episode
ends exactly at `duration_sec`; stop there without repeating the workflow.
Use the schema-required fields only, never `null`. Set `schema_version` to
`candidate_video_annotation_v3`, `submission_profile` to
`reliable_annotation`, and `submission_status` to `annotated`.

The JSON Schema below specifies the output format; do not copy schema metadata
such as `$schema`, `properties`, or `$defs` into the answer. Before ending,
check that facts are not repeated, the whole timeline is covered, and the JSON
is closed. Return only the candidate JSON object, without analysis or Markdown.
```

Frozen Staged-Generation System Prompts. The staged runs use independent complete-video summary and timeline calls, followed by detail calls over model-predicted intervals. The detail listing below is the frozen run-bundle snapshot, rather than a later revision of the working prompt file.

```
Inspect the complete supplied egocentric video from beginning to end. Focus on
the camera wearer's observable workflow; include another person's activity
only when it directly interacts with or changes that workflow. Return one to
three concise, non-duplicative facts describing its observed overall workflow
and major stages. Distinguish an observed outcome from an intended
one. Do not infer unseen expected steps, object identities, or completion.
Audio may be absent; do not infer speech or sound unless supplied. The opaque
video ID is not a scenario label. Do not use folder names, filenames, or human
annotations. This call is independent of the timeline call; do not invent
episode counts or timestamps. Copy the supplied video ID and duration exactly.
Return only the requested JSON object.
```

```
Inspect the complete supplied egocentric video from beginning to end and
independently recover its coarse episode structure. No summary, ground-truth
timeline, event count, or scenario vocabulary is provided. Return only episode
IDs, types, time bounds, source spans, and uncertainty reasons. Details are
```

```
requested in a later call; do not compress episodes into a single narrative.

Use `task` for sustained goal-directed work, `navigation_transition` for
meaningful movement between work areas, `idle` for observable inactivity, and
`cannot_identify` only where activity genuinely cannot be resolved. Set
`granularity=coarse_episode` for every episode. Make meaningful boundaries at
sustained activity or state changes, not every hand motion, camera shake,
brief pause, object touch, or every few seconds. A long task may contain
several actions without requiring a new coarse episode for each one. Include
navigation, idle, and unidentifiable spans rather than skipping them.
Do not merge an observable task, subsequent travel, and a new task into one
episode merely to keep the timeline short. `source_span` should briefly name
the visible activity or transition, not repeat its numeric time range.

Use ordered half-open intervals, beginning at 0 and ending at the supplied
duration, with adjacent end/start boundaries equal. Do not claim subframe
precision or fill unseen intervals with imagined activity. If an interval is
unobservable, mark it `cannot_identify` and give a short reason such as
`visibility`, `source_gap`, `identity_continuity`, or `intent_unknown`.
All other uncertainty reasons must be the empty string. Use `pred_ep_0001`,
`pred_ep_0002`, and so on in chronological order. Copy the supplied video ID
and duration exactly. Return only the requested JSON object.
```

```
Inspect the supplied egocentric video interval and describe only the assigned
episodes. The assigned timeline was predicted by this system in an earlier
call; it is not ground truth. Preserve its exact episode IDs and times, but do
not invent actions or evidence to justify an incorrect boundary or type. Times
in your answer are absolute seconds from the beginning of the full recording.
Return exactly one `episode_details` object for each assigned episode, in the
same order, with no additional or missing IDs.

Focus on the camera wearer's observable work. A task episode needs at least
one concrete action fact. Use additional facts for distinct visible actions,
targets, or state changes within that episode; do not repeat the same action
with paraphrases. Bind `predicate` to the actual `affected_entity`,
`destination`, functionally used `tool`, `room`, `zone`, and observed
`state_or_result`. Merely seeing, holding, or carrying an object does not prove
tool use. Do not replace distinct visible steps with a generic word such as
`prepare`, `clean`, or `organize`. At the same time, do not assert familiar but
unseen steps, completed results, precise identities, or unavailable audio.
Use `hedged` or `uncertain` for visually ambiguous details, and `negated` only
for an observable non-occurrence.

Every fact must contain all schema-required fields: `candidate_fact_id`,
`source_span`, `polarity`, `certainty`, `predicate`, `affected_entity`,
`destination`, `tool`, `room`, `zone`, and `state_or_result`. The fields
`affected_entity`, `destination`, `tool`, `zone`, and `state_or_result` are
arrays of strings; `room` and `predicate` are single strings. Use an
empty array or empty string for an inapplicable role, never `null`. Do not
repeat an identical string within one array. If multiple indistinguishable
objects are observed, express their count in a single string such as
`two plates` when visually supported.

Every episode must also contain the complete `navigation` object,
`temporal_qualifiers` array, and `evidence` array. For a non-navigation episode,
the navigation object is exactly
`{"applicable":false,"candidate_fact_id":"","source_span":"","from_room":"","from_zone":"","
    to_room":"","to_zone":"","carried_objects":[]}`.
For a `navigation_transition`, keep exactly the same keys, set `applicable`
to `true`, give its navigation fact a unique ID, and fill only visually
supported from/to locations and carried objects. The four location fields are
strings; `carried_objects` is an array of strings. An object or room appearing
in view does not prove movement to it. If there is no supported qualifier or
evidence window, use an empty array. An evidence interval must lie within its
assigned episode and cite only fact IDs from that episode.

Use unique episode-scoped fact IDs, such as `pred_fact_0001_001` and
`pred_nav_0001`. `source_span` briefly names the visible action or transition,
not only its timestamp. Copy the supplied video ID and duration exactly. The
filename, folder, and opaque video ID are not content labels. The supplied
JSON Schema defines the output fields and types. Return only one complete
JSON object, without analysis, Markdown, or an extra explanation.
```

A.2 Evaluation Metric Details

The following definitions provide the complete diagnostic views used by the three EEB evaluation dashboards. The main paper retains only the primary measures and the components that define the evaluator; the metrics below support comprehensive analysis of temporal completeness, boundary localization, workflow order, proposition matching, and annotation reliability.

A.2.1 Hierarchy Dashboard Details

Timeline coverage. For a video of duration T, let D_{union} denote the duration covered by the union of all predicted episode intervals. The fraction of the recording left uncovered is

$$\text{GapRate} = \frac{T - D_{\text{union}}}{T}. \tag{5}$$

Timeline overlap is also checked during schema validation. Because the output specification requires a gapless and non-overlapping episode partition, overlap is treated as a schema-validity diagnostic rather than a benchmark score.

Temporal episode matching. For a predicted episode p and a ground-truth episode g, temporal intersection over union is

$$\text{tIoU}(p, g) = \frac{|p \cap g|}{|p \cup g|}. \tag{6}$$

Following the order-aware event alignment used in SODA (Fujita et al., 2020), dynamic programming produces an order-preserving one-to-one match set M_τ at temporal-overlap threshold τ. Matching first maximizes the number of eligible pairs and then maximizes their total tIoU.

For untyped matching, a pair is eligible when $\text{tIoU}(p, g) \geq \tau$. For typed matching, predicted and ground-truth episode types must additionally agree.

For P predicted episodes and G ground-truth episodes,

$$TP_\tau = |M_\tau|, \qquad FP_\tau = P - |M_\tau|, \qquad FN_\tau = G - |M_\tau|. \tag{7}$$

Precision, recall, and F_1 are then computed in the standard way. We retain typed and untyped Segment F_1 at $\tau \in \{0.10, 0.30, 0.50, 0.70\}$ as diagnostic views.

The primary temporal-structure measure is **Strict Typed Episode** F_1, defined in Section 4. In addition to matching episode types and requiring $\text{tIoU}(p, g) \geq 0.70$, both predicted endpoints must lie within the reference-specific tolerance

$$\epsilon(g) = \max(0.5, \min(2.0, 0.2|g|)) \text{ s}. \tag{8}$$

Boundary localization. Only internal episode boundaries are evaluated. Let ground-truth boundary b_k separate adjacent episodes with durations d_k and d_{k+1}. Its tolerance is

$$\delta_k = \max(0.5, \min(2.0, 0.2 \min(d_k, d_{k+1}))) \text{ s}. \tag{9}$$

Predicted and reference boundaries are matched one-to-one. Matching first maximizes the number of boundary pairs satisfying the corresponding reference-specific tolerance and then minimizes total absolute boundary error.

The resulting matches produce **Boundary** F_1 and **matched-boundary MAE**. We additionally report boundary-count error and spurious boundaries per minute as diagnostics of over- and under-segmentation. Boundary MAE is computed only over successfully matched boundary pairs. Missed and spurious boundaries affect Boundary F_1 but do not enter the MAE; thus, the two measures should be interpreted jointly.

Duration-weighted episode typing. Duration Type F_1 evaluates how accurately the predicted timeline assigns recording time across the four episode classes:

`task`, `navigation_transition`, `idle`, `cannot_identify`.

The four-class macro F_1 is computed per video over applicable episode types and then video-macro averaged across the benchmark.

Segmental type order. Following established evaluation practice in temporal action segmentation (Abu Farha & Gall, 2019), consecutive repeated episode-type labels are collapsed before comparing predicted and ground-truth type sequences.

Let Y_P and Y_G denote the resulting predicted and ground-truth sequences. Normalized Segmental Edit is

$$\text{Edit} = 1 - \frac{\text{Lev}(Y_P, Y_G)}{\max(|Y_P|, |Y_G|)}. \tag{10}$$

Segmental Edit serves as a diagnostic of episode-type order rather than a primary benchmark measure.

Required workflow order. Some reference annotations specify required temporal-order relations between workflow elements. For a required pair $o = (a, b)$, let endpoint recovery values be

$$r_a, r_b \in \{0, 0.5, 1\},$$

corresponding to missed, partially recovered, and fully recovered endpoints, and define

$$u_o = \min(r_a, r_b). \tag{11}$$

For the set O of required order pairs,

$$\text{PairRecovery} = \frac{1}{|O|} \sum_{o \in O} u_o, \tag{12}$$

$$\text{JointOrder} = \frac{1}{|O|} \sum_{o \in O} u_o \mathbb{1}[a \prec b], \tag{13}$$

$$\text{StrictOrder} = \frac{1}{|O|} \sum_{o \in O} \mathbb{1}[u_o = 1 \wedge a \prec b]. \tag{14}$$

PairRecovery measures recovery of required endpoints without requiring their temporal direction to be correct. JointOrder additionally requires recovered endpoints to appear in the correct order, whereas StrictOrder gives credit only when both endpoints are fully recovered and correctly ordered.

A.2.2 Fine-grained Dashboard Details

Eligibility for semantic comparison. Candidate and ground-truth episode pairs are eligible for fine-grained semantic comparison only when their episode types agree and $\text{tIoU}(p, g) \geq 0.30$. This threshold requires meaningful temporal localization while avoiding a second strict-boundary penalty after the Hierarchy dashboard. Global-summary propositions are not tied to local episode intervals and are therefore globally eligible.

Proposition outcomes. Following the role-preserving proposition representation described in Section 4.3, candidate claims may be adjudicated as

`entailed`, `partially supported`, `contradicted`, `supported extra`, `unsupported extra`, `unresolved extra`, `duplicate`, `masked`.

Reference facts are independently categorized as

`covered`, `partial`, `missed`.

Proposition scoring. The headline proposition score assigns 1.0 credit to a fully supported match, 0.5 to a partially supported match, and 0 otherwise. For each applicable video v, if c_i denotes candidate-side credit and r_j reference-side credit,

$$P_v = \frac{\sum_i c_i}{N_{\text{cand},v}}, \qquad R_{\text{ref},v} = \frac{\sum_j r_j}{N_{\text{ref},v}}, \qquad F_{1,v} = \frac{2P_v R_{\text{ref},v}}{P_v + R_{\text{ref},v}}. \tag{15}$$

If $P_v = R_{\text{ref},v} = 0$, $F_{1,v}$ is defined as zero. Dataset-level Precision, Reference Fact Recall, and F_1 are independently video-macro averaged across applicable videos. Consequently, the macro F_1 reported in Table 3 is not generally equal to the harmonic mean of the reported macro Precision and Recall.

$N_{\text{ref},v}$ contains only explicitly annotated reference facts. Optional context may receive candidate-side precision credit without entering the reference-recall denominator. Exact-credit variants, in which only fully supported matches receive credit, are retained as audit diagnostics.

Bidirectional pair credit and capacity-constrained allocation. Semantic adjudication produces candidate-side and reference-side decisions for each temporally, episode-type, and field-compatible candidate–reference pair. Candidate support is mapped to

$$s^c_{ij} = \begin{cases} 1, & \texttt{entailed}, \\ 0.5, & \texttt{partially supported}, \\ 0, & \text{otherwise}, \end{cases}$$

while reference coverage is mapped to

$$s^r_{ij} = \begin{cases} 1, & \texttt{covered}, \\ 0.5, & \texttt{partial}, \\ 0, & \texttt{missed}. \end{cases}$$

For an eligible bidirectionally confirmed pair (i, j), the proposition credit is

$$w_{ij} = \min(s^c_{ij}, s^r_{ij}). \tag{16}$$

Therefore, $w_{ij} = 1$ only when the candidate is fully supported and the reference fact is fully covered. If either side is partial, the pair receives at most 0.5 credit. Pairs without a valid bidirectional match do not receive positive matching credit.

Credit allocation is then performed with capacity one on both sides. During allocation, the implementation adds a deterministic secondary preference for required reference facts to break otherwise competing assignments. Let $\tilde{w}_{ij}$ denote this allocation weight. The evaluator solves

$$X^* = \arg\max_X \sum_{(i,j)\in E} \tilde{w}_{ij} x_{ij}, \tag{17}$$
$$\text{s.t.} \quad x_{ij} \in \{0,1\},$$
$$\sum_j x_{ij} \le 1, \qquad \sum_i x_{ij} \le 1.$$

The secondary required-reference preference affects only which eligible pairs are selected. Final candidate-side and reference-side proposition credits are computed from the original w_{ij} values, not from the augmented $\tilde{w}_{ij}$. Consequently, the allocation preference cannot inflate Precision, Reference Fact Recall, or F_1.

Each candidate claim and reference fact therefore has capacity one. An unallocated candidate that repeats or paraphrases an already allocated fact receives no additional proposition credit and is recorded as a duplicate.

Role-preserving relation scoring. Fine-grained evaluation preserves semantic bindings rather than scoring isolated matching words. Actions, affected entities, tools, locations, states, carried objects, and navigation relations are compared strictly within their annotated roles. For example, identifying an object does not constitute an equivalent fact when one annotation states that the object is merely held while another states that it is functionally used as a tool.

A.2.3 Annotation Reliability Dashboard Details

Shared coverage quantities. Grounded Precision and Reference Fact Recall reuse the candidate-side precision and reference-side recall quantities defined by the Fine-grained dashboard. They are repeated in the Reliability dashboard so that content coverage can be interpreted together with confirmed unsupported, unresolved, and duplicate claims.

Candidate-side accounting. Let N_C be the number of non-masked candidate claims, N_H be confirmed contradicted or unsupported claims, N_U be unresolved extra claims, and N_D be unallocated duplicate claims. We report

$$\text{HallucinationRate} = \frac{N_H}{N_C}, \qquad \text{ReviewRate} = \frac{N_U}{N_C}, \tag{18}$$
$$\text{DuplicateRate} = \frac{N_D}{N_C}, \qquad \text{UFR}_U = \frac{N_H + N_U}{N_C}. \tag{19}$$

Confirmed Hallucination Rate counts only claims adjudicated as contradicted or unsupported by available evidence. **Review Rate** separates unresolved claims rather than automatically treating them as errors. **Duplicate Rate** measures repeated or paraphrased claims that could not receive additional credit after capacity-constrained allocation. The conservative UFR_U combines confirmed unsupported content and unresolved extras as an audit diagnostic.

All reliability rates are computed from unrounded per-video quantities before video-macro aggregation; displayed rounded component rates may therefore differ from their displayed sum by 0.1 percentage points.

Reference-side reliability. Reference Fact Recall is computed from the same reference-side proposition credits used by the Fine-grained dashboard. Candidate abstention or conservative output does not remove reference facts from the recall denominator. A model may therefore reduce speculative claims by producing less content, but missed required facts continue to reduce recall.

Cannot-identify intervals. The `cannot_identify` label represents intervals where visual evidence is insufficient to support a more specific annotation. Correct localization is evaluated through the same temporal machinery used for other episode types. Predicting `cannot_identify` over clearly observable task content is penalized as incorrect episode typing.

A.2.4 AGGREGATION AND APPLICABILITY

All dashboard metrics are first computed independently for each video and then video-macro averaged across applicable videos. Precision, recall, and F_1 are macro-averaged independently; therefore, a dataset-level macro F_1 is not reconstructed from the displayed macro Precision and Recall.

When a metric has no valid denominator for a particular video, its value is reported as N/A with a machine-readable reason and excluded from that metric's macro average. Claim-micro and duration-micro variants are retained as secondary diagnostics.

Claim-micro metrics pool claim-level numerators and denominators across applicable videos, while duration-micro metrics pool the corresponding duration-based quantities.

A.2.5 COMPLETE REPORTING VIEWS

Table 5 summarizes the primary and retained diagnostic views produced by the evaluator.

Table 5: Primary and retained diagnostic views associated with the three evaluation dashboards.

Dashboard	Primary and retained diagnostic views
Hierarchy	Summary F_1; Strict Typed Episode F_1; Typed and Untyped Segment F_1; Boundary F_1 and MAE; Duration Type F_1; Gap Rate; Segmental Edit; Pair Recovery; Joint Order; Strict Order
Fine-grained	Role-aware proposition precision, recall, and F_1 under one-to-one typed $\mathrm{tIoU} \geq 0.30$ alignment; per-role F_1; exact-credit diagnostic
Reliability	Grounded Precision; Reference Fact Recall; Confirmed Hallucination Rate; Review Rate; Duplicate Rate; UFR_U diagnostic; cannot-identify evaluation

B ADDITIONAL DATASET DETAILS

All 472 videos contain a human-authored global summary and a complete typed episode timeline. The 2,891 episodes comprise 1,758 task, 994 navigation, 106 idle, and 33 `cannot_identify` intervals. Across the benchmark, 449 videos contain an explicit room label and 258 contain structured navigation. Annotations include 1,081 episodes with a non-empty tool role, 1,685 with an affected entity or target, 1,899 with a room, and 2,050 with a local zone. Open-language annotations span 162 room expressions, 1,104 zone expressions, 542 tool expressions, 1,682 affected-entity expressions, and 2,549 episode-action expressions.

The 10 environment and 13 workflow families are multi-label reporting views derived from human annotations rather than candidate labels supplied to evaluated systems. Table 6 lists the principal analysis groups.

Table 6: Evaluation-derived environment, workflow, and recording-condition groups.

Dimension	Groups
Environment	Domestic kitchen; bathroom/laundry; bedroom/living/dining; utility/storage; hotel guest room/bathroom; hotel service area; public restroom; public indoor area; outdoor/building exterior; tabletop workspace
Workflow	Food preparation/cooking; serving/eating; dishwashing/kitchen cleaning; floor and surface cleaning; laundry/textile handling; bed/linen handling; pet care; waste/storage; hotel housekeeping/service; public restroom/area cleaning; outdoor waste service; tabletop assembly/inspection; personal hygiene
Auxiliary activity	Navigation/transport; digital-device use; social interaction; waiting/rest
Visibility condition	Clear; low light; occluded; camera off target; source gap; identity continuity unresolved; task boundary unresolved; intent unknown

C BASELINE IMPLEMENTATION DETAILS

All seven baselines receive identical wearer-centered annotation requirements and produce the version schema. Visual observations follow native model interfaces, as summarized in Table 7. Exact checkpoint revisions, prompt hashes, decoder settings, sampled-frame metadata, and per-video run records are retained in released manifests.

Table 7: Baseline video-input and annotation-generation protocols.

System	Visual observation	Annotation generation
Gemini 3 Flash Preview	Full-duration 720p video through native interface at 1 FPS, high media resolution	Direct complete-schema response in a single pass
GPT-4.1	Timestamped frames at 1 FPS, up to 320 frames, 1024px width, high detail	Direct complete-schema response in a single pass
Qwen3-VL-8B-Instruct	Complete-video passes at 1 FPS; local detail windows use 1 FPS sampling	Staged: independent summary and timeline, followed by 2-episode detail batches
Qwen3-VL-32B-Instruct	Official video processor at 1 FPS, up to 512 frames within total pixel budget	Direct complete-schema response in a single pass
LongVA-7B-DPO	256 uniformly sampled frames for complete-video and detail windows	Staged: independent summary and timeline, followed by 2-episode detail batches
InternVL3.5-8B	32 uniformly sampled temporal observations for complete-video and detail passes	Staged: independent summary and timeline, followed by 2-episode detail batches
GLM-4.6V-Flash	Complete-video and detail-window observations sampled at 1 FPS	Staged: independent summary and timeline, followed by 2-episode detail batches

Direct Complete Annotation. Direct systems observe the complete recording and produce global summaries, typed timelines, role-bound facts, navigation, and evidence in a single schema-constrained response. Gemini uses its native video interface with a full-duration 720p proxy. GPT-4.1 receives timestamped frames at 1 FPS (up to 320 frames) encoded at 1024-pixel width with high detail. Qwen3-VL-32B uses its official processor at 1 FPS, capped at 512 frames under the checkpoint's total pixel budget. System prompts explicitly restrict annotation to the camera wearer, require full-duration coverage, distinguish visible versus functionally used objects, and request explicit uncertainty for visually unresolved content.

Staged Complete Annotation. Compact open-weight models use a three-stage pipeline. Independent complete-video calls first generate global summaries and typed timelines; the timeline call does not receive the generated summary. Detail calls then receive at most two adjacent *model-predicted* episodes, observing their combined interval with 5 seconds of temporal padding on each side, adding role-bound facts, navigation, qualifiers, and evidence without altering assigned identifiers, types, or boundaries.

Summary, timeline, and detail token limits are 2,048, 8,192, and 8,192, respectively. If a detail batch exceeds its output limit, it is retried with smaller batch sizes; unrecoverable single-episode overflows are marked as failed predictions. A deterministic assembler compiles phase outputs into the final schema, updating internal identifiers and evidence pointers.

Reporting Two Protocols. All four staged baselines were first tested with direct complete annotation and repeatedly truncated before completing the full schema. Staging therefore reduces per-call output length for these models; it does not simplify temporal grounding or provide oracle intervals. Detail calls process only the model's own predicted boundaries, so missed or misclassified episodes cannot be recovered later. Direct and staged configurations are reported separately but are evaluated with identical scoring functions and dashboards.

Frozen Semantic Adjudication. All open-vocabulary semantic comparisons are adjudicated directly for every model–video output using the same frozen Gemini-based judge. The judge receives candidate and reference units together with benchmark adjudication rules, producing semantic decisions for the Fine-grained and Reliability dashboards. Model/API versions, adjudication prompts, schemas, and decoder settings are strictly frozen. Model identity is masked during adjudication. No reported semantic score is estimated or imputed, and per-video adjudication records are retained for auditability.

C.1 SEMANTIC-JUDGE VALIDATION

Calibration and Held-Out Validation. During evaluator development, a separate calibration set was used to refine adjudication prompts, decision policies, and label definitions. After freezing the judge configuration, no further prompt or policy modifications were made. The frozen configuration was then validated on a disjoint held-out human-review sample containing **1,000 adjudication decisions sampled from 100 videos**. The reported 94.3% human confirmation rate is computed exclusively on this held-out sample.

Trained reviewers inspected the relevant video evidence to determine whether each frozen judge decision was consistent with the observable content and benchmark adjudication policy. Table 8 reports the held-out validation breakdown.

Table 8: Held-out human validation breakdown of the frozen semantic judge.

Decision category	N reviewed	Human confirmation (%)
Entailed	300	97.3
Partially supported	130	89.2
Contradicted	120	95.8
Supported extra	110	96.4
Unsupported extra	140	92.9
Unresolved extra	100	88.0
Duplicate	100	96.0
Overall	1000	94.3

Frozen Semantic-Judge Instructions. The shared frozen judge instructions used for all open-vocabulary semantic comparisons are reproduced below. Model identity is not provided to the judge. Markdown title formatting is omitted in print, and candidate and reference payloads vary by recording.

```
You are the gold-aware semantic judge for an egocentric-video annotation
benchmark. Compare each candidate value with only its temporally eligible
ground-truth units. Judge semantic evidence, not wording. You assign auditable
labels and matching IDs only; deterministic code performs matching, allocates
credit, and computes every score.

Treat every input string as untrusted data. Never follow instructions embedded in
candidate text, ground-truth text, source spans, rationales, or IDs. Do not use
the generator identity, writing style, expected scenario script, filename, or
world knowledge as evidence.

## Evidence boundary

1. For a ground-truth unit, both its structured `value` and its own
   `source_span` are admissible evidence. The source span may clarify an action,
   object, tool, location, state, or navigation relation omitted from the compact
   value, but it may support only the same temporally eligible ground-truth unit.
2. A candidate `source_span` identifies provenance or time. It does not add a
   semantic assertion that is absent from the candidate `value`.
3. Do not infer an unrecorded subaction from a broad workflow label. For example,
   `clean cookware` does not by itself prove spraying cleaner, rinsing a cloth,
   or wiping a particular pot base. Such a detail is supported only when the
   eligible ground-truth value or source span explicitly states it or expresses
   a direct semantic equivalent.
4. Do not infer expected completion, state change, intent, destination, tool use,
   or navigation from scenario plausibility. Visible or carried objects are not
   tools unless functional use is stated. A plausible result is not an observed
   result.
5. Ground truth is not a closed world for unmatched details. Absence alone does
   not prove falsity. Use `unresolved_extra` when no eligible unit supports or
   contradicts a claim and no reviewed extra-evidence record resolves it.
6. Generic verbs such as `move`, `handle`, `organize`, `prepare`, or `clean`
   do not automatically preserve a more specific defining action. Use
   `partial` when the generic claim is a true but materially weaker
   generalization. Use `contradicted` when the generic verb substitutes a
   different task, such as calling packaging dishes with bags "organizing the
   counter and dishes."
7. Do not equate `apply cleaning liquid` with `pour cleaning liquid`. Evidence
   that a held container was not poured refutes pouring, but does not by itself
   refute application by dipping or another mechanism.
8. Candidate precision is directional. If every role asserted by a concise
   candidate is supported, label the candidate `entailed` even when the matched
   ground-truth unit contains additional actions or entities. Represent the
   omitted ground-truth content only in the separate coverage decision.
9. In a fine-grained proposition, a spatial phrase such as `appliance area`
   does not fully preserve a list of specifically manipulated entities such as
   a rice cooker, cup, jug, and table surface. If the action is otherwise true,
   label this loss of the affected-entity role `partial`.
10. A materially wrong action specialization is a contradiction when the
    eligible evidence establishes a different manipulation. For example,
    `fold clothing` conflicts with evidence that the garment was only picked up
    or handled and was not shown folded; it is not merely incomplete.
```

```
11. Treat an incompatible physical-state or substance description as a material
    entity/state conflict. For example, a bowl of millet is not a bowl of yellow
    liquid merely because its contents appear yellow.

## Candidate-label policy

Evaluate every candidate value exactly once.

- `entailed`: every material assertion made by the candidate is supported by the
  matched eligible ground-truth value or source span. Paraphrases, singular/plural
  variation, ordinary count compression, and harmless room-name aliases are
  allowed when they do not change the action, entity, tool, state, location, or
  navigation relation.
- `partial`: the core action or relation is supported, but the candidate omits a
  required material role, weakens the scope, merges only part of a compound unit,
  or adds a non-conflicting qualifier that the eligible evidence does not fully
  establish. State the missing or unverified role in the rationale.
- `contradicted`: at least one material asserted role conflicts with eligible
  ground-truth evidence. A different object, tool, room, state, endpoint,
  polarity, completion claim, or temporally incompatible action is a
  contradiction only when the eligible evidence establishes the conflict.
  Absence alone is never a contradiction.
- `supported_extra`: no eligible reference unit is an appropriate match, but a
  supplied reviewed extra-evidence record explicitly supports the claim.
- `unsupported_extra`: no eligible reference unit is an appropriate match, but a
  supplied reviewed extra-evidence record explicitly refutes or forbids it.
- `unresolved_extra`: no eligible reference unit resolves the claim and no
  reviewed extra-evidence record decides it. This label requests video review and
  is not automatically a hallucination.
- `masked`: the candidate falls wholly inside an explicit ambiguity mask.

Use `partial` rather than `entailed` when a role-bound proposition has a correct
predicate but an unsupported or missing material entity, tool, destination,
state, room, zone, carried object, or navigation endpoint. Use `contradicted`
only when that role is affirmatively inconsistent with the eligible evidence.

Do not downgrade one candidate summary proposition merely because other workflow
stages are absent from the overall submitted summary. Judge the local proposition
against the matching ground-truth summary evidence; missing propositions are
handled by ground-truth coverage.

## Ground-truth coverage policy

Evaluate every required or optional ground-truth value exactly once.

- `covered`: cited candidate values express all material content of the
  ground-truth unit or an explicitly acceptable granularity.
- `partial`: cited candidate values recover the core content but omit or weaken a
  material role, scope, state, endpoint, or required part of a compound unit.
- `missed`: no candidate value expresses the ground-truth unit.
- `masked`: the unit is excluded by an explicit ambiguity mask.

Candidate precision and ground-truth coverage are separate decisions. A concise
candidate claim may be semantically correct yet cover only part of a richer
ground-truth unit. Keep the two directions consistent without forcing their
labels to be identical.

## Role and granularity checks

For a serialized `proposition`, inspect polarity, certainty, and all roles
jointly. Before assigning `entailed`, explicitly verify:

1. predicate or navigation relation;
2. affected entity or carried object;
3. tool versus merely visible or held object;
4. destination and state/result, when asserted;
5. room, zone, and navigation endpoints, when asserted;
6. polarity, certainty, and completion strength.

Treat these as harmless only when the eligible evidence makes the equivalence
clear: singular versus plural, `guest room` versus `hotel room`, `vanity area`
versus `sink area` within the same described location, or a generic object name
for an explicitly compatible subtype. Do not treat different functional tools,
different target objects, different rooms, reversed endpoints, or completed
versus ongoing actions as harmless granularity.

## Contract rules

1. Use only IDs supplied in the input. A candidate may match only IDs listed in
   its `eligible_gold_value_ids`; temporal eligibility is already encoded.
2. `entailed`, `partial`, and `contradicted` require at least one matched eligible
```

```
   ground-truth ID.
3. `supported_extra`, `unsupported_extra`, and `unresolved_extra` require an empty
   match list. Only `unresolved_extra` sets
   `requires_video_adjudication=true`.
4. Forbidden ground-truth values are constraints, not recall targets. Required
   ground truth cannot be masked.
5. Keep candidate matches and ground-truth coverage edges reciprocal. `covered`
   requires entailed support; partial support cannot create full coverage.
6. Do not split, repair, or add roles to a candidate proposition. Do not allocate
   duplicate credit. Deterministic code performs one-to-one capacity-constrained
   matching.
7. Rationales must name the decisive supported, missing, unverified, or
```

```
   conflicting role. Avoid generic statements such as "similar task" or "not in
   gold."

Return only JSON matching the supplied schema. Do not output scores, probabilities,
rankings, or recommendations. Copy the supplied `mapping_version` exactly.
```