TRACECUT: ATOMIC ACTION SEGMENTATION AND LABELING FOR ROBOT MANIPULATION VIDEOS VIA ACTIONABILITY-AWARE HIERARCHY DECODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon tasks and their corresponding planning models represent a prominent yet challenging frontier in robotics research. While collecting diverse long-horizon task videos has become a key focus, fine-grained action segmentation and annotation remain fundamental to constructing high-quality datasets. Although temporal grounding based on vision-language models (VLMs) offers a natural solution, it often struggles inferior accuracy due to the burden of simultaneously handling action perception, temporal localization, and semantic labeling. To address this, we propose TraceCut, a video atomic action segmentation and annotation pipeline based on hierarchical segment decomposition. TraceCut builds a hierarchical action tree for robot videos to enable accurate multi-level segmentation. It achieves precise atomic action splitting and annotation by generating and aggregating segment-level descriptions. Furthermore, we develop a lightweight video captioning model to fulfill the critical requirement for efficient and accurate caption generation within the pipeline. We also introduce RoboActionTrace, a standardized benchmark for evaluating robot video action segmentation and annotation. Experiments demonstrate that both the TraceCut pipeline and the lightweight captioning model achieve competitive performance in atomic action segmentation and annotation. Moreover, data constructed via this pipeline significantly enhances performance in core long-horizon task planning embodied task, demonstrating substantial commercial potential amid the growing demand for automated annotation.

1 INTRODUCTION

The focus of robotic manipulation is shifting toward long-horizon tasks, whose successful execution relies critically on task planning Peng et al. (2026); Jin & Zhang (2026). However, most existing robot manipulation datasets do not provide supervision at this level of granularity. A single video may contain multiple operations—such as grasping, placing, opening, pouring, wiping, folding, and recovery motions, while being annotated with only one high-level instruction. To support downstream policy learning, reward modeling, and VLA-style planning, these long videos should be temporally segmented into subtask-level clips with localized language labels to form an atomic action trace.

Robot video segmentation differs fundamentally from general video segmentation. In general videos, temporal boundaries are often indicated by visually salient shot transitions, such as cuts or scene changes Wake et al. (2025); Ren et al. (2023); Huang et al. (2024); Guo et al. (2025) Robot demonstrations, by contrast, are typically recorded as a single continuous shot without such explicit visual cues. Within this shot, a robot may approach an object, adjust the gripper, briefly pause, grasp, lift, move, release, and retreat. Chen et al. (2026); Macrodata Labs (2026) Segmenting each of these motor phases separately would produce fragments that are too fine-grained to represent meaningful task-level actions, whereas grouping multiple state-changing operations into a single segment would make the result too coarse for planning supervision. The desired unit is therefore neither a low-level motion primitive nor a coarse segment containing multiple operations, but a semantically complete atomic action that induces a meaningful state change and can serve as reusable supervision for downstream models.

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

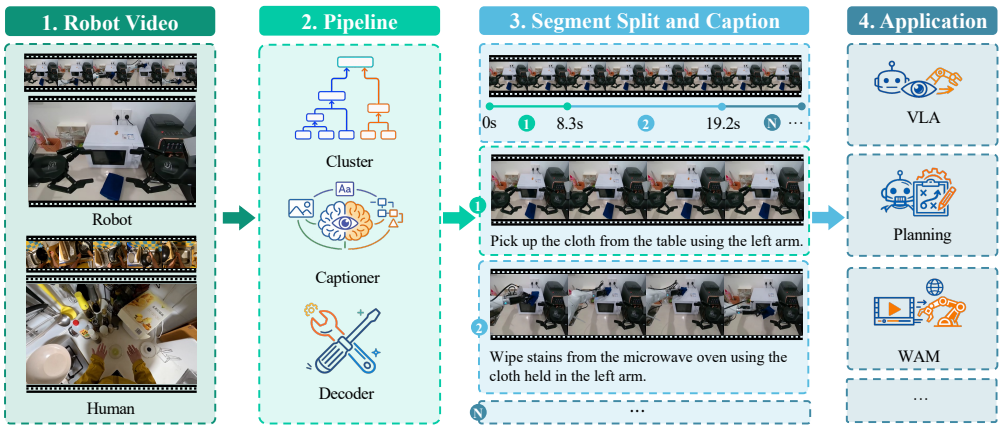


Figure 1: TraceCut precisely segments raw long-horizon robot videos into semantically annotated atomic action clips via hierarchical action tree decomposition and lightweight caption aggregation. The resulting standardized data empowers downstream models such as VLA, WAM, and embodied planner models, significantly enhancing long-horizon task performance.

The most straightforward approach is VLM-based temporal grounding, which prompts a vision-language model to jointly identify subtasks and localize their temporal boundaries. For example, WGO/Refiner converts a video into timestamped contact sheets and prompts Gemini to directly output subtask segments together with their language labels Macrodata Labs (2026). Other methods augment the VLM with additional priors or observations. TREAD first asks the model to generate a sequence of semantic subtasks and then localizes the corresponding robot sub-trajectories Kuramshin et al. (2026), while proprioception-aware prompting supplies joint and end-effector states as complementary cues for segmentation Suzuki et al. (2025). Related approaches may similarly provide planned subtasks or predefined action schemas before localization Tabakov et al. (2026); Feng et al. (2026). Meanwhile, time-aware video LLMs explicitly model temporal information to improve timestamp grounding Ren et al. (2023); Huang et al. (2024); Guo et al. (2025). Despite these enhancements, VLM-based temporal grounding places substantial demands on the model: it must not only understand the video and infer its task structure, but also determine the appropriate action granularity and precisely localize the temporal boundaries of each subtask.

A complementary approach is to postpone the final boundary decision and first construct a high-recall set of candidate segments at multiple temporal scales. Action100M, for example, organizes temporal segments and their captions into a Tree of Captions, allowing evidence from different levels of the hierarchy to be aggregated into open-vocabulary action labels Chen et al. (2026). DenseStep2M follows a similar staged design, separating initial video segmentation, VLM-based step generation, and LLM-based refinement Ge et al. (2026).

Inspired by this line of work, we present TraceCut which represents a robot video as a high-recall hierarchy of captioned segments and defer the selection of the final action trace to a subsequent decoding stage. Direct applying this strategy to robot videos, however, introduces a distinct challenge: overlapping parent, child, and sibling segments may all receive plausible descriptions while differing substantially in their actionability. Fine-grained nodes may capture only motor fragments, whereas coarse nodes may merge several independent state-changing operations. Object-centric systems such as NILS Blank et al. (2024) and VLA metadata pipelines Peng et al. (2026); Wu et al. (2026); Zhang et al. (2026) suggest that object-state changes and robot skill interfaces provide useful cues, but they do not resolve how to select a coherent and nonredundant action trace from an overcomplete segment hierarchy. We therefore focus on the problem of *actionability-aware decoding*: deciding which nodes represent complete state-changing robot actions, filtering out nodes that merely describe low-level motor execution, and resolving redundant parent–child candidates without discarding true atomic operations.

To summarize, we make three contributions:

- We propose TraceCut, a hierarchy-to-trace decoding pipeline that integrates a lightweight Robot Caption VLM and an actionability-aware decoder to achieve precise atomic action segmentation and annotation.
- We introduce RoboActionTrace, a standardized benchmark with a layered evaluation protocol that comprehensively assesses both end-to-end pipelines and segment captioning models.
- Experiments show TraceCut achieves competitive segmentation performance, and data constructed via this pipeline significantly improves embodied long-horizon task planning.

2 Related Work

Long-Horizon Embodied Models and Data Curation. Recent advances in Vision-Language-Action (VLA) models and embodied task planning model have underscored the critical role of large-scale, atomic skill or subtask annotations in supporting model training. Cortex formalizes subtasks into canonical skill primitives with unidirectional temporal constraints to align planning and execution, constructing atomic action data via automated annotation of open-source videos and simulations Peng et al. (2026). LingBot-VLA 2.0 uses an automated VLM pipeline to segment manipulation videos into temporally contiguous subtasks, grouping single interactive actions into interaction-level lists for consistent granularity Wu et al. (2026). AtomicVLA decomposes trajectories into atomic skills using principal axis analysis, motion patterns, and gripper states to support model training Zhang et al. (2026).While these efforts emphasize atomic action annotation, they primarily target downstream utilization rather than foundational data curation. In contrast, TraceCut tackles the core challenge of segmenting and annotating long-horizon task videos, providing essential infrastructure for large-scale embodied model training.

VLM-based Atomic Action Segmentation. Video temporal segmentation has attracted significant VLM-based attention. Methods like T-PIVOT Wake et al. (2025), TimeChat Ren et al. (2023), VTimeLLM Huang et al. (2024), VTG-LLM Guo et al. (2025), Vid2Seq Yang et al. (2023), and PDVC Wang et al. (2021) achieve segmentation through techniques such as iterative visual prompting, explicit temporal tokenization, sequential boundary generation, and parallel event grounding. However, these approaches target salient shots in generic videos and are ill-suited for fine-grained atomic action segmentation in robot manipulation. In robotics, WGO/Refiner is the most relevant work on atomic action segmentation annotation, establishing WGO-Bench for subtask annotation and evaluating VLM-based methods Macrodata Labs (2026). TREAD adopts a two-stage strategy that decomposes high-level instructions into semantic subtasks before VLM localization Kuramshin et al. (2026), while Proprioception-aware Prompting incorporates joint and end-effector states to move beyond RGB-only paradigms Suzuki et al. (2025). Despite these advances, end-to-end VLM methods must simultaneously handle action perception, temporal segmentation, and semantic labeling, imposing heavy burdens on accuracy, latency, and token consumption. In contrast, our method introduces a high-recall pre-slicing module to obtain candidate segments before VLM refinement, significantly improving both pipeline accuracy and efficiency.

Action Trees and Semantic Segmentation. Compared to end-to-end VLM segmentation, a two-stage strategy of fine-grained visual segmentation followed by semantic aggregation is generally more accurate. DenseStep2M segments videos into coherent shots and uses multimodal models to generate structured procedural steps Ge et al. (2026), but targets only generic videos. Action100M employs V-JEPA 2 Assran et al. (2025) to build a hierarchical Tree-of-Captions for sequential open-vocabulary action labeling Chen et al. (2026); while it provides an accurate baseline for human actions, it fails to achieve atomic-level slicing. In contrast, TraceCut is designed for robotic atomic trajectories, precisely matching atomic-action annotations and robustly handling meaningless transitions in robot videos.

3 Method

3.1 Problem Formulation

We propose TraceCut, a framework that decomposes a long-horizon robot manipulation video into an ordered trace of captioned atomic actions. Each atomic action corresponds to one complete task-level state-changing operation, such as picking up an object, placing it at a target, opening

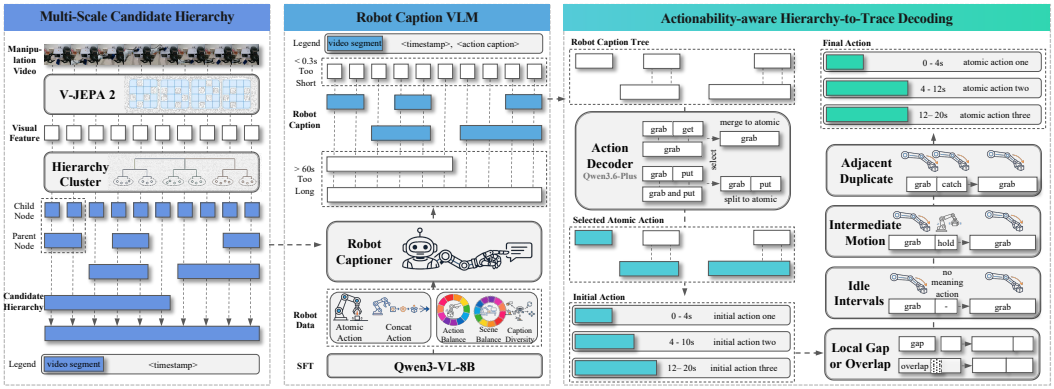


Figure 2: TraceCut hierarchy-to-trace pipeline. Stages 1–2 construct and caption an overcomplete temporal hierarchy. Candidate-guided semantic decoding produces an Initial Action Trace, and deterministic refinement resolves action dependencies and reconciles temporal boundaries. The real local example shows a short release phase consolidated into the preceding placement. Dashed links denote temporal-semantic evidence flow rather than explicit node attribution.

a container, inserting an object, wiping a surface, or pouring material. Formally, the output of TraceCut is an ordered trace

$$\mathcal{T} = \{(s_i, e_i, y_i)\}_{i=1}^{N}, \tag{1}$$

where s_i and e_i denote the start and end timestamps of the i-th atomic action, respectively, and y_i is its corresponding natural-language caption. Unlike an unordered set of temporal segments, a trace explicitly preserves the temporal ordering of actions, which is essential for representing the sequential progression of long-horizon manipulation tasks.

3.2 Pipeline Overview

As illustrated in Figure 2, TraceCut consists of three stages: **Multi-Scale Candidate Hierarchy**, **Robot Caption VLM**, and **Actionability-Aware Hierarchy-to-Trace Decoding**.

In the first stage, TraceCut converts the input video into a **multi-scale hierarchy of temporal candidates** rather than committing immediately to a single segmentation. The candidate nodes retain parent–child and sibling relations, yielding an overcomplete hypothesis space that preserves alternative action granularities for subsequent reasoning.

In the second stage, **Robot Caption VLM**, a robot-specialized model initialized from Qwen3-VL-8B-Instruct, labels each fixed candidate node. For every candidate, the corresponding raw video clip is extracted and processed using duration-adaptive frame sampling. The model then generates either an atomic-action caption or an ordered multi-action caption. These node-level captions are treated as localized visual-language evidence rather than final trace predictions.

In the third stage, **Actionability-Aware Hierarchy-to-Trace Decoding** converts the captioned candidate hierarchy into the final atomic-action trace. Duration-horizon conditioning first clips or removes candidates outside the valid video duration, preventing actions beyond the evaluation horizon from entering the decoded trace. A text-only LLM then aggregates the serialized hierarchy into an initial set of node-grounded atomic actions. Deterministic refinement subsequently resolves actionability, parent-child redundancy, implausible durations, local gaps and overlaps, and low-level motor fragments. Short completion phases following placement-like operations, such as release after placing, hanging, inserting, attaching, or returning an object, are consolidated into the preceding state-changing action.

3.3 Multi-Scale Candidate Hierarchy

To capture candidate action segments from evolving visual dynamics, Stage 1 first encodes sampled frames using a pretrained V-JEPA 2 video encoder Assran et al. (2025) to obtain robust video

representations. It then performs temporally constrained hierarchical clustering over adjacent frame features, progressively merging visually similar contiguous intervals into a coarse-to-fine candidate hierarchy. Each candidate node in the multi-scale hierarchy is represented as

$$v_j = (s_j, e_j, d_j, p_j, \mathrm{Ch}_j, m_j), \tag{2}$$

where s_j and e_j denote the temporal boundaries of the candidate segment, d_j is its depth in the hierarchy, p_j and Ch_j denote its parent and child nodes, respectively, and m_j stores provenance and hierarchy metadata. Deeper nodes typically correspond to shorter, temporally localized candidates, whereas shallower nodes preserve longer temporal context that may better capture complete action semantics.

Note that hierarchy depth records the order of visual-feature merging rather than a calibrated semantic granularity. A complete atomic action may emerge at different depths, while nodes at the same depth may correspond to a single atomic action, multiple consecutive actions, or only a low-level motor phase. Consequently, the multi-scale hierarchy does not admit a fixed-depth frontier that consistently represents atomic actions.

The resulting hierarchy preserves complementary temporal evidence across multiple scales but is intentionally overcomplete. Different nodes may correspond to the same action with slightly different temporal boundaries, whereas fine-grained nodes often capture only intermediate robot motions such as reaching, gripper adjustment, or release, and coarse nodes may span multiple independent state-changing operations. Therefore, the hierarchy serves as a candidate space rather than a final segmentation, leaving the selection of a compact, temporally consistent, and non-redundant atomic-action trace to the subsequent stages.

3.4 Robot Caption VLM

The candidate hierarchy typically contains hundreds of nodes, making direct annotation with large proprietary VLMs prohibitively expensive. To enable efficient node-wise labeling, we train a robot-specialized **Robot Caption VLM** based on Qwen3-VL-8B-Instruct. Given a candidate node with fixed temporal boundaries, the model predicts only its semantic description and does not modify the temporal interval or perform final trace decoding.

Because nodes in the Multi-Scale Candidate Hierarchy are generated at different temporal scales, a candidate clip may contain either a single atomic action or multiple consecutive atomic actions. The captioner must therefore adapt its output cardinality to the visual content of each clip rather than relying on hierarchy depth or clip duration. Given a candidate clip x_j, Robot Caption VLM produces an ordered open-vocabulary caption

$$C_j = (c_{j,1}, \ldots, c_{j,K_j}), \tag{3}$$

where $c_{j,k}$ describes the k-th task-level operation observed in the clip, including the manipulated object, target, or resulting state when necessary. The action count K_j is inferred from the clip itself: a single-action candidate should yield one atomic-action description, whereas a multi-action candidate should cover all constituent actions in temporal order.

To learn this adaptive labeling behavior, we jointly train the model on atomic-action and multi-action clips. Atomic supervision teaches the model to stop after describing one semantically complete operation instead of unnecessarily decomposing low-level motor phases such as reaching, aligning, grasping, or releasing. Multi-action supervision encourages complete and temporally ordered coverage when a candidate spans several consecutive operations. Together, these two forms of supervision allow the captioner to handle candidate nodes of varying semantic granularity without assuming a fixed correspondence between hierarchy depth and action cardinality.

Robot manipulation datasets often contain repetitive tasks and similar trajectories collected in constrained environments, resulting in homogeneous language annotations. To improve generalization, we augment captions with paraphrases that preserve the operation, manipulated object, target, and resulting state. Multi-action clauses are paraphrased independently while retaining their temporal order. Atomic-action and multi-action training samples are balanced across datasets, tasks, and episodes. We also use duration-adaptive frame sampling under a fixed frame budget: shorter clips are sampled more densely to preserve pre-contact, contact, and post-contact evidence, while longer clips are sampled more sparsely. Sampling depends solely on clip duration.

3.5 ACTIONABILITY-AWARE HIERARCHY-TO-TRACE DECODING

TraceCut aggregates multi-scale captions into a single ordered atomic-action trace. A text-only LLM (Qwen-Plus) combines global context from higher-level nodes with localized descriptions from lower-level nodes to distinguish redundant descriptions from consecutive state changes. Each action is grounded to candidate nodes and their temporal intervals. The initial trace is then refined through four deterministic operations.

Semantic consolidation. The decoder first removes non-action descriptions and consolidates semantically redundant predictions. Multiple candidates describing the same state-changing operation are merged into a single semantic action, while explicit no-action outputs and repeated semantic descriptions are discarded.

Boundary reconciliation. The retained actions are then reconciled along the temporal axis. Parent–child redundancies are resolved by selecting the candidate with the most complete semantic description and the best-supported temporal interval. Adjacent duplicates are merged, local overlaps are eliminated, and short gaps introduced during semantic consolidation are absorbed into neighboring actions.

Actionability filtering. The decoder distinguishes complete task-level operations from low-level motor phases. Candidates describing semantic actions such as grasping, placing, opening, pouring, inserting, or wiping are retained, whereas intermediate motions including reaching, hovering, alignment, retreat, local adjustment, and idle motion are suppressed. Duration is used only as an auxiliary prior, preventing short but semantically complete actions from being removed solely because of their temporal extent.

Completion-phase consolidation. For placement-like operations, including placing, hanging, inserting, attaching, and returning, a short release phase immediately following the state change is merged into the preceding action rather than emitted as a separate trace element. This preserves completion evidence while avoiding unnecessary fragmentation of a single semantic operation.

4 ROBOACTIONTRACE BENCHMARK CONSTRUCTION

Current robotics research lacks a benchmark for long-horizon tasks that jointly provides fine-grained action segmentation and language annotations. To address this gap, we construct the RoboActionTrace benchmark by integrating diverse public datasets (e.g. AgiBotWorld Bu et al. (2025), OpenGalaxea Jiang et al. (2025), RoboCoin Wu et al. (2025), RoboInter-Data Li et al. (2026), RoboMind2.0 Wu et al. (2024); Hou et al. (2025)) and internally collected robot video data. We retain clips lasting at least 20 seconds that contain a minimum of two actions and no unlabeled intervals, and balance data sources, tasks, scenes, durations, and action counts to obtain 100 short- and long-horizon robot videos. All videos are manually annotated with temporally bounded atomic actions and corresponding language descriptions at a fine-grained level, ensuring high accuracy and reliability for long-horizon robot task evaluation.

Recognizing that precise video annotation is critical for both robot data curation and accurate atomic-action segmentation, we further construct a fixed-boundary benchmark from RoboActionTrace to evaluate the Robot Caption VLM independently of the trajectory decomposition stage. This benchmark consists of 500 single-action clips, 500 consecutive multi-action clips, and a small number of no-action clips, all sampled to ensure balanced coverage of data sources, action categories, durations, and temporal positions. To guarantee the reliability of the annotations, every clip is verified to be intact, fully decodable, and strictly aligned with its labeled temporal boundaries.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Training data. We construct the training data for Robot Caption VLM from robot manipulation videos with subtask-level or task-level annotations. For videos with annotated subtask intervals, we extract clips containing either a single subtask or multiple consecutive subtasks and pair them with the corresponding descriptions. Videos with task-level annotations are paired with their associated

task instructions. During data preparation, we exclude samples identified as overlapping with the RoboActionTrace test set.

Evaluation. We evaluate the full TraceCut pipeline on 100 robot manipulation videos from RoboActionTrace and 75 from WGO-Bench Macrodata Labs (2026). We additionally assess the pipeline on the 25 egocentric human-hand videos in WGO-Bench, replacing Robot Caption VLM with Gemini-3.5-Flash while keeping all other pipeline components unchanged. Results are reported separately for each dataset and embodiment. To assess captioning quality independently of segmentation, we compare captioning models on the same single-action and multi-action robot video clips from both benchmarks, as detailed in Section 5.3.

Metrics. Predicted and ground-truth segments are matched one-to-one by temporal IoU within each video. We report four metrics: **Temporal mIoU@.75**, the mean IoU of pairs matched at IoU ≥ 0.75; **Segment F1@.5**, segmentation F1 at IoU ≥ 0.5 regardless of labels; **Label Acc.@.75**, semantic accuracy among pairs matched at IoU ≥ 0.75 with valid judge decisions; and **Semantic E2E F1@.75**, which counts only temporally matched pairs with valid, semantically correct judgments as true positives. Both F1 metrics aggregate counts across videos, and semantic E2E F1 retains all segments in its denominators, including those with invalid judge decisions. You can refer to the detailed formulation of these metrics in the supplementary files.

5.2 Full-pipeline Evaluation on Benchmark Datasets

Evaluation setup. We compare TraceCut with Refiner, TREAD, ROVER, UVD+VIP, and NILS-adapted using common evaluation splits, temporal matching, and aggregation settings within each benchmark (Table 1). Our adapted implementations use Gemini 3.1 Pro Preview for TREAD and ROVER and GPT-5.4 for NILS object proposals. Under default inputs, Refiner, TREAD, and ROVER receive task instructions; the other methods do not. UVD+VIP and NILS-adapted produce boundaries only and are excluded from semantic evaluation. Blind human review covers all 100 RoboActionTrace and 75 WGO75 videos under default inputs, scoring full-trace segmentation and labeling separately on 0–10 scales. Boundary-only methods receive segmentation scores only. These ratings are distinct from fixed-boundary caption evaluation (Section 5.3).

Results under default inputs. TraceCut leads all three automatic metrics on RoboActionTrace, with Semantic E2E F1@.75 of 0.1025 versus 0.0644 for the strongest baseline. On WGO75, its Segment F1@.5 is close to the best result (0.5666 versus 0.5685), but both semantic metrics trail all three instruction-conditioned baselines. Thus, its temporal performance does not imply an equivalent semantic advantage. Human review likewise favors TraceCut's segmentation (5.64/6.65 on RoboActionTrace/WGO75), but Refiner leads labeling (5.71/6.20 versus TraceCut's 4.26/3.61). The automatic and human labeling rankings differ on RoboActionTrace; these evaluations assess different scopes, since Label Accuracy conditions on temporal matches and valid judgments, whereas human review assesses complete traces.

Instruction content and sensitivity. WGO75 task instructions can explicitly name constituent operations rather than only specify high-level goals (Appendix D.4). Connective-form instructions occur in 47/75 WGO75 videos (62.7%), compared with 23/100 (23.0%) on RoboActionTrace (Table 8). This lexical indicator does not measure coverage of the complete annotated action sequence.

We replace the instructions of Refiner, TREAD, and ROVER with empty strings while retaining other pipeline and evaluation settings. Only pipeline inputs change; the semantic judge retains the same task context and criteria. On WGO75, Semantic E2E F1@.75 drops from 0.1593 to 0.0699 for Refiner and from 0.1560 to 0.1033 for TREAD, larger decreases than on RoboActionTrace (0.0552 to 0.0472 and 0.0644 to 0.0514). Segmentation changes are not uniform: Refiner's WGO75 Segment F1@.5 increases from 0.5345 to 0.5551 despite its semantic decline. ROVER generates empty labels after its required root-task input is removed; its zero semantic scores therefore reflect an input-contract failure, not a fully adapted video-only baseline.

Under the evaluated instruction-free settings, TraceCut leads all three automatic metrics on both benchmarks. These results support its use without sample-level task descriptions, but do not isolate the effect of instruction granularity or establish general semantic superiority across different backbones and data distributions.

Method	Task instr.	Default inputs			Without task instructions			Human review	
		Seg. F1 @.5	Label Acc.@.75	Sem. E2E F1 @.75	Seg. F1 @.5	Label Acc.@.75	Sem. E2E F1 @.75	Seg. (default)	Label. (default)
RoboActionTrace: 100 robot manipulation videos									
Refiner	Yes	0.3347	<u>0.4714</u>	0.0552	0.3318	<u>0.3750</u>	0.0472	<u>5.37</u>	**5.71**
TREAD	Yes	<u>0.3746</u>	0.3882	<u>0.0644</u>	<u>0.3419</u>	0.3714	<u>0.0514</u>	4.09	4.21
ROVER	Yes	0.3029	0.3077	0.0344	0.2069	0.0000	0.0000	2.81	2.29
UVD+VIP	No	0.0655	N/A	N/A	0.0655	N/A	N/A	1.66	N/A
NILS-adapted	No	0.0672	N/A	N/A	0.0672	N/A	N/A	2.11	N/A
TraceCut (Ours)	No	**0.4074**	**0.6303**	**0.1025**	**0.4074**	**0.6303**	**0.1025**	**5.64**	<u>4.26</u>
WGO75: 75 robot manipulation videos									
Refiner	Yes	0.5345	0.7377	<u>0.1593</u>	<u>0.5551</u>	0.3800	0.0699	<u>6.40</u>	**6.20**
TREAD	Yes	**0.5685**	**0.7759**	0.1560	0.5240	<u>0.4058</u>	<u>0.1033</u>	5.76	<u>5.68</u>
ROVER	Yes	0.5196	<u>0.7460</u>	**0.1938**	0.3496	0.0000	0.0000	5.05	3.91
UVD+VIP	No	0.3130	N/A	N/A	0.3130	N/A	N/A	5.15	N/A
NILS-adapted	No	0.2173	N/A	N/A	0.2173	N/A	N/A	3.39	N/A
TraceCut (Ours)	No	<u>0.5666</u>	0.4607	0.1399	**0.5666**	**0.4607**	**0.1399**	**6.65**	3.61

Table 1: Full-pipeline results on RoboActionTrace and WGO75. Task instr. indicates instruction use under default inputs; methods not using instructions retain identical automatic scores across input conditions. Automatic scores range from 0 to 1; human scores range from 0 to 10 and cover default inputs only. Higher is better. Bold and underlined values mark the best and second-best results per dataset and column. N/A denotes inapplicable metrics. ROVER's instruction-free semantic zeros result from empty labels.

5.3 Independent Fixed-Boundary Caption VLM Evaluation

Full-pipeline label accuracy is affected by both temporal localization and semantic description. We therefore fix the clip boundaries and compare only the captions produced by VLMs. We use four evaluation sets with 100 clips each. RoboActionTrace (RAT) Atomic100 and WGO Robot-Atomic100 evaluate fixed-boundary atomic labeling, while RAT Concat100 and WGO Robot-MultiAction100 evaluate fixed-boundary multi-action labeling.

Human4D protocol. Each caption is scored from 0 to 10 on four dimensions: absence of hallucination, action completeness, unambiguous object reference, and spatial-information completeness. The Human4D score is the mean of these four scores, and higher is better. All four evaluation sets have 100 completed reviews. A detailed protocol is in the Appendix.

Comparison with baselines. Table 2 reports Human4D scores on the reviewed subsets. Seed performs best across all four datasets, averaging 7.18. Our model ranks second with 6.31, outperforming its base model, Qwen3-VL-8B-Instruct (5.26), and Qwen3.6-Plus (5.87). It performs competitively on RAT Atomic, WGO Atomic, and RAT Concat, especially on action completeness, object-reference clarity, and spatial-information completeness, but remains weaker on WGO MultiAction. This weakness appears specific to long, low-resolution, multi-action videos. We find that half of the WGO MultiAction subset consists of 320×180 low-resolution clips, on which our score drops to 4.36, compared with 6.73 on the 1280×720 higher-resolution clips.

5.4 Downstream Validation on Planning Benchmarks

To validate our segmentation pipeline, we apply TraceCut to robot manipulation videos from RoboMIND Wu et al. (2024), RealSource RealSource (2025), and RoboChallenge Yakefu et al. (2025) to generate 30K planning supervision samples pairing high-level instructions with temporally localized substeps. Fine-tuning Qwen3-VL-8B-Instruct on this data yields substantial gains over the base model: accuracy improves from 44.2 to 59.6 (+15.4) on EgoPlan-Bench Chen et al. (2024) and 41.9 to 53.9 (+12.0) on EgoPlanBench-2 Qiu et al. (2024).

Model	RAT Atomic	WGO Atomic	RAT MultiAction	WGO MultiAction	Avg.
Seed	**7.08**	**6.30**	**7.75**	**7.58**	**7.18**
Qwen3.6-Plus	6.18	4.53	6.93	<u>5.84</u>	5.87
RynnBrain-8B	4.71	4.94	4.93	4.72	4.83
MiMo-Embodied-7B	5.04	4.02	5.49	4.34	4.72
Qwen3-VL-8B-Instruct	5.00	4.65	5.83	5.54	5.26
Ours (8B)	<u>7.04</u>	<u>5.54</u>	<u>7.11</u>	5.54	<u>6.31</u>

Table 2: Human4D scores on four evaluation sets (0–10; higher is better). Avg. is the unweighted mean across the sets. Bold and underlining indicate the best and second-best scores, respectively.

Training	EgoPlan-1	EgoPlan-2
Original	44.2	41.9
With TraceCut data	**59.6**	**53.9**
Δ	+15.4	+12.0

Table 3: Planning accuracy (%) of Qwen3-VL-8B-Instruct before and after fine-tuning on TraceCut-generated data.

Variant	Seg. F1@.5 ↑
Initial LLM trace	0.3631
+ Semantic consolidation	0.4005
+ Boundary reconciliation	<u>0.4046</u>
+ Full deterministic refinement	**0.4074**

Table 4: Cumulative decoder ablation without ground-truth endpoint alignment.

Since planning is highly sensitive to boundary and label correctness, noisy or misaligned segments would prevent consistent improvement; thus, these gains serve as a strong proxy for segmentation quality. The consistent performance across two benchmarks further confirms that the extracted traces capture task-level state transitions, validating the end-to-end reliability of our pipeline.

5.5 Ablation Studies

We evaluate the hierarchy-to-trace decoder by varying only the enabled refinement operations, while fixing the Stage 1 candidate hierarchy, Stage 2 captions, hierarchy-conditioned LLM outputs, and all other decoder settings.

Cumulative trace refinement. Starting from the initial LLM trace, which already incorporates JSON parsing and adjacent duplicate merging, we progressively enable semantic consolidation, boundary reconciliation, and the remaining deterministic refinement operations. Actionability filtering and completion-phase consolidation are enabled jointly in the final stage, so their individual contributions are not isolated.

Table 4 shows that semantic consolidation yields the largest observed improvement, increasing Segment F1@.5 from 0.3631 to 0.4005 (paired-bootstrap 95% CI for the gain: $[0.0180, 0.0598]$). This stage removes non-action predictions and merges semantically redundant candidates. Boundary reconciliation repairs local gaps and overlaps, raising the score to 0.4046 (+0.0041), although its 95% CI, $[-0.0039, 0.0120]$, includes zero. The final refinement stage further increases the score to 0.4074 (+0.0028; 95% CI: $[0.0010, 0.0048]$). These incremental gains are conditional on the chosen refinement order. Semantic metrics are omitted because pairwise-consistent frozen judge annotations are unavailable for all intermediate variants.

6 Conclusion

We propose TraceCut, a hierarchy-decoding approach for atomic action segmentation and annotation in robot manipulation videos. To overcome the accuracy bottleneck of existing vision-language models in long-horizon task planning, TraceCut leverages a hierarchical action tree and training-free deterministic refinement to produce actionable, temporally coherent traces, supported by a lightweight captioning model. Experiments on our newly constructed RoboActionTrace Benchmark and WGO-Bench show competitive segmentation performance, and data from our pipeline significantly boosts embodied long-horizon planning, validating its practical value amid growing automated annotation demand.

AI USE STATEMENT

Generative AI was used to assist manuscript organization and language polishing.

ETHICS STATEMENT

To the best of our knowledge, this work does not raise any specific ethical concerns.

REPRODUCIBILITY STATEMENT

The proposed method is described in [method section], and the experimental setup and evaluation procedures are detailed in [experiment section and relevant appendices]. We will publicly release the source code to support reproducibility and facilitate future research.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Nils Blank, Moritz Reuss, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Wenzel, Oier Mees, and Rudolf Lioutikov. Scaling robot policy learning via zero-shot labeling with foundation models. *arXiv preprint arXiv:2410.17772*, 2024.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

Delong Chen, Tejaswi Kasarla, Yejin Bang, Mustafa Shukor, Willy Chung, Jade Yu, Allen Bolourchi, Theo Moutakanni, and Pascale Fung. Action100m: A large-scale video action dataset. *arXiv preprint arXiv:2601.10592*, 2026.

Yi Chen, Yuying Ge, Yixiao Ge, Mingyu Ding, Bohao Li, Rui Wang, Ruifeng Xu, Ying Shan, and Xihui Liu. Egoplan-bench: Benchmarking multimodal large language models for human-level planning, 2024. URL https://arxiv.org/abs/2312.06722.

Youhe Feng, Hansen Shi, Haoyang Li, Xinlei Guo, Yang Wang, Chengyang Zhang, Jinkai Zhang, Xiaohan Zhang, Jie Tang, and Jing Zhang. Procvlm: Learning procedure-grounded progress rewards for robotic manipulation. *arXiv preprint arXiv:2605.08774*, 2026.

Mingji Ge, Qirui Chen, Zeqian Li, and Weidi Xie. Densestep2m: A scalable, training-free pipeline for dense instructional video annotation. *arXiv preprint arXiv:2604.26565*, 2026.

Yongxin Guo, Jingyu Liu, Mingda Li, Dingxin Cheng, Xiaoying Tang, Dianbo Sui, Qingbin Liu, Xi Chen, and Kevin Zhao. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3302–3310, 2025.

Chengkai Hou, Kun Wu, Jiaming Liu, Zhengping Che, Di Wu, Fei Liao, Guangrun Li, Jingyang He, Qiuxuan Feng, Zhao Jin, et al. Robomind 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence. *arXiv preprint arXiv:2512.24653*, 2025.

Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14271–14280, 2024.

Tao Jiang, Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Jianning Cui, Xiao Liu, Shuiqi Cheng, Jiyang Gao, Huazhe Xu, and Hang Zhao. Galaxea open-world dataset and g0 dual-system vla model. *arXiv preprint arXiv:2509.00576*, 2025.

Ruofan Jin and Zaixi Zhang. Agentic-vla: Efficient online adaptation for vision-language-action models. *CoRR*, abs/2605.22896, 2026. doi: 10.48550/ARXIV.2605.22896. URL https://doi.org/10.48550/arXiv.2605.22896.

Artur Kuramshin, Özgür Aslan, Cyrus Neary, and Glen Berseth. Task robustness via re-labelling vision-action robot data. *arXiv preprint arXiv:2606.10918*, 2026.

Hao Li, Zi-han Ding, Shuai Yang, Yilun Chen, Yang Tian, Xiaolin Hu, Tai Wang, Dahua Lin, Feng Zhao, Si Liu, et al. Robointer: A holistic intermediate representation suite towards robotic manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 113314–113381, 2026.

Macrodata Labs. Segmenting robot video into actionable subtasks, 2026. URL https://macrodata.co/blog/annotating-robot-video-subtasks.

Jiaqi Peng, Xiqian Yu, Delin Feng, Yuqiang Yang, Wenzhe Cai, Jing Xiong, Ganlin Yang, Jinliang Zheng, Jiafei Cao, Xueyuan Wei, et al. Cortex: A bidirectionally aligned embodied agent framework for long-horizon manipulation. *arXiv preprint arXiv:2607.05377*, 2026.

Lu Qiu, Yi Chen, Yuying Ge, Yixiao Ge, Ying Shan, and Xihui Liu. Egoplan-bench2: A benchmark for multimodal large language model planning in real-world scenarios, 2024. URL https://arxiv.org/abs/2412.04447.

RealSource. Realsource world: A large-scale real-world dual-arm manipulation dataset. https://huggingface.co/datasets/RealSourceData/RealSource-World, 2025.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. *CoRR*, abs/2312.02051, 2023. doi: 10.48550/ARXIV.2312.02051. URL https://doi.org/10.48550/arXiv.2312.02051.

Philip Schroeder, Ondrej Biza, Thomas Weng, Hongyin Luo, and Jim Glass. Rover: Recursive reasoning over videos with vision-language models for embodied tasks. *Advances in Neural Information Processing Systems*, 38:140623–140669, 2026.

Kanata Suzuki, Shota Shimizu, and Tetsuya Ogata. Proprioception enhances vision language model in generating captions and subtask segmentations for robot task. *arXiv preprint arXiv:2512.20876*, 2025.

Stefan Tabakov, Asen Popov, Dimitar Dimitrov, S Ensiye Kiyamousavi, Boris Kraychev, and Vladimir Hristov. Atomic action slicing: Planner-aligned options for generalist vla agents. In *Proceedings of the 41st ACM/SIGAPP Symposium on Applied Computing*, pp. 780–782, 2026.

Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Open-vocabulary action localization with iterative visual prompting. *IEEE Access*, 2025.

Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6847–6857, 2021.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yumeng Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation, 2024. URL https://arxiv.org/abs/2412.13877.

Shihan Wu, Xuecheng Liu, Shaoxuan Xie, Pengwei Wang, Xinghang Li, Bowen Yang, Zhe Li, Kai Zhu, Hongyu Wu, Yiheng Liu, et al. Robocoin: An open-sourced bimanual robotic data collection for integrated manipulation. *arXiv preprint arXiv:2511.17441*, 2025.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, et al. From foundation to application: Improving vla models in practice. *arXiv preprint arXiv:2607.06403*, 2026.

Adina Yakefu, Bin Xie, Chongyang Xu, Enwen Zhang, Erjin Zhou, Fan Jia, Haitao Yang, Haoqiang Fan, Haowei Zhang, Hongyang Peng, Jing Tan, Junwen Huang, Kai Liu, Kaixin Liu, Kefan Gu, Qinglun Zhang, Ruitao Zhang, Saike Huang, Shen Cheng, Shuaicheng Liu, Tiancai Wang, Tiezhen Wang, Wei Sun, Wenbin Tang, Yajun Wei, Yang Chen, Youqiang Gui, Yucheng Zhao, Yunchao Ma, Yunfei Wei, Yunhuan Yang, Yutong Guo, Ze Chen, Zhengyuan Du, Ziheng Zhang, Ziming Liu, and Ziwei Yan. Robochallenge: Large-scale real-robot evaluation of embodied policies, 2025. URL https://arxiv.org/abs/2510.17950.

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.

Likui Zhang, Tao Tang, Zhihao Zhan, Xiuwei Chen, Zisheng Chen, Jianhua Han, Jiangtong Zhu, Pei Xu, Hang Xu, Hefeng Wu, et al. Atomicvla: Unlocking the potential of atomic skill learning in robots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20743–20754, 2026.

A Appendix

This supplement provides additional implementation details, dataset analyses, and experimental results that complement the main paper. It preserves the task definition, benchmark identities, evaluation protocols, and headline conclusions reported in the submitted manuscript.

The supplement serves four complementary purposes. First, it provides implementation details for the three core modules of TraceCut, enabling full reproducibility of the proposed pipeline. Second, it analyzes the full Video-to-Trace evaluation by jointly examining the characteristics of RoboActionTrace and WGO Bench Robot75 together with their benchmark-specific evaluation behavior, providing additional interpretation of the results reported in the main paper. Third, it presents an in-depth evaluation of Robot Caption VLM under fixed temporal boundaries, including its training-data distribution, checkpoint comparison, and fixed-boundary captioning performance, thereby isolating caption quality from temporal decomposition. Finally, it provides additional analyses of cross-embodiment generalization, candidate coverage, decoder components, and representative failure modes.

B Reproducibility and Implementation Details

B.1 Multi-Scale Candidate Hierarchy

The Multi-Scale Candidate Hierarchy is the first stage of TraceCut, which converts an input video into an overcomplete hierarchy of temporally ordered candidate segments instead of committing to a single segmentation. As described in the main paper, the hierarchy preserves parent–child and sibling relations, allowing multiple plausible action granularities to coexist for subsequent captioning and hierarchy-to-trace decoding.

The hierarchy is constructed by first encoding temporally sampled video frames using a pretrained V-JEPA 2 video encoder Assran et al. (2025). Hierarchical agglomerative clustering is then performed under a strict temporal-connectivity constraint, so that every merge combines two adjacent temporal intervals while preserving the temporal ordering of the original video.

Each candidate node stores its temporal interval, hierarchy depth, parent and child indices, together with construction metadata. Hierarchy depth records only the order of feature-space merging and is not interpreted as semantic granularity. Consequently, candidate nodes at the same depth may correspond to a complete atomic action, multiple consecutive actions, or only a motor phase. Rather than selecting a fixed hierarchy level, all candidate nodes are retained as candidate evidence for the subsequent Robot Caption VLM and Actionability-aware Hierarchy-to-Trace Decoding stages.

Configuration	Value
Base model	Qwen3-VL-8B-Instruct
Fine-tuning	Full-parameter SFT, 1 epoch
Train / validation samples	2,000,000 / 20,000
Atomic / multi-action ratio	1:3
Per-device batch / accumulation	2 / 1
LLM / aligner learning rate	1×10^{-6}
Vision-encoder learning rate	5×10^{-7}
Warmup / weight decay	0.03 / 0.1
Maximum gradient norm	0.5
Precision and parallelism	BF16, FlashAttention, ZeRO-3
Input configuration	32,768 tokens; at most 240 frames

Table 5: Training configuration of the Robot Caption VLM used in the submitted experiments.

B.2 Robot Caption VLM

Robot Caption VLM is the second stage of TraceCut. As described in the main paper, it predicts semantic descriptions for fixed candidate nodes without modifying their temporal boundaries. The generated captions serve as localized visual-language evidence for the subsequent Actionability-aware Hierarchy-to-Trace Decoding stage rather than directly producing the final atomic-action trace.

The submitted Robot Caption VLM is initialized from Qwen3-VL-8B-Instruct through a previously trained robot-caption checkpoint. Before the final training stage, both the optimizer and learning-rate scheduler are reinitialized, followed by one epoch of full-parameter supervised fine-tuning.

The model is trained on a corpus of two million training samples. An additional validation set containing 20,000 samples is held out exclusively for monitoring training progress, selecting checkpoints, and early quality assessment; it is never used for parameter optimization. Unless otherwise specified, all experiments reported in the main paper use the training configuration summarized in Table 5.

During training, fixed-boundary atomic-action clips and concatenated multi-action clips are sampled with a fixed ratio of 1:3. This ratio is the empirical configuration adopted throughout this work rather than a claim of universal optimality.

B.3 Actionability-aware Hierarchy-to-Trace Decoding

Actionability-aware Hierarchy-to-Trace Decoding is the final stage of TraceCut. As described in the main paper, it converts the captioned candidate hierarchy into the final atomic-action trace through one text-LLM aggregation step followed by deterministic refinement. This section focuses on the implementation details of the decoding process.

Hierarchy serialization and text-LLM aggregation. The captioned hierarchy is serialized into an indented text representation before being passed to the text LLM. Each serialized candidate node contains a stable node identifier, temporal interval, and caption, while indentation encodes the parent–child and sibling relations within the hierarchy. Nodes marked as skipped, pruned, erroneous, multi-action placeholders, or *no meaningful action* are omitted from serialization, although their valid descendants remain recursively accessible.

The root caption provides global task context but is not emitted as an atomic action unless the entire video consists of a single repeated operation. Qwen-Plus is invoked once per video over the complete serialized hierarchy. The prompt explicitly constrains the language model to the serialized candidate

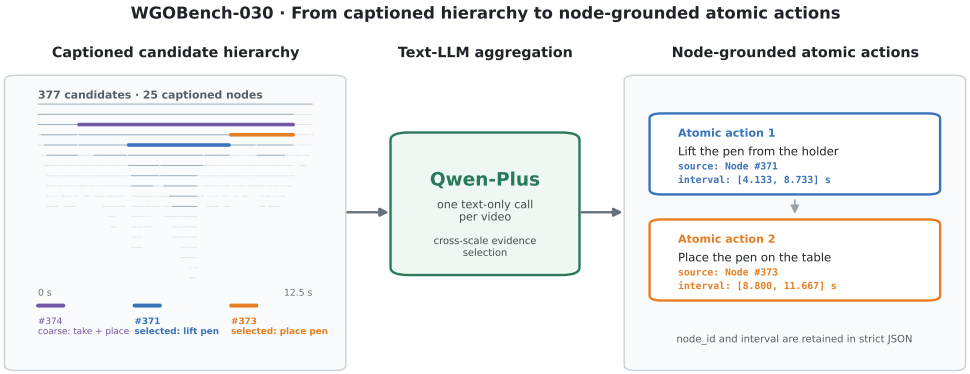


Figure 3: From a captioned hierarchy to node-grounded atomic actions, illustrated with real WGOBench030 candidates. The left panel shows the complete 377-node hierarchy and highlights candidate evidence at multiple scales. Qwen-Plus aggregates the serialized hierarchy once per video and selects two temporally ordered atomic actions. Each output action retains its source node identifier and directly inherits the corresponding candidate interval; the implementation serializes these fields as strict JSON rather than freely regressing temporal boundaries.

nodes: it must not introduce actions or manipulated objects that are unsupported by the candidate hierarchy, and every predicted action must retain the identifier and temporal interval of its originating candidate node. Temporal boundaries are therefore inherited directly from candidate nodes rather than being freely regressed by the language model. The output is a strict JSON-formatted, node-grounded initial trace. Figure 3 illustrates this grounding contract.

Deterministic refinement. The node-grounded initial trace is subsequently refined through a sequence of deterministic operations. These operations implement the semantic consolidation, boundary reconciliation, actionability filtering, and completion-phase consolidation stages described in the main paper. They resolve semantic redundancy, repair local temporal inconsistencies, suppress non-actionable motor fragments, and consolidate short completion phases following placement-like actions.

The refinement rules operate only on the predicted trace and hierarchy metadata without revisiting the original video. They filter, merge, and normalize predicted actions while preserving the semantic action set generated by the language model.

C Datasets and Evaluation Protocol

This section provides additional details of the datasets, evaluation protocols, and human-review procedures used throughout the paper. It complements the experimental setup described in the main paper and focuses on reproducibility rather than experimental interpretation.

C.1 Robot Caption VLM Training Data

As described in the main paper, Robot Caption VLM is trained using three types of training samples. Fixed-boundary atomic-action clips supervise semantic granularity, fixed-boundary multi-action clips supervise action coverage and temporal ordering, and video–task pairs provide additional task-level semantic diversity when dense subtask annotations are unavailable.

Atomic-action clips contain one complete state-changing operation, whereas multi-action clips contain multiple temporally ordered operations within a fixed temporal interval. Video–task pairs expand the diversity of tasks, scenes, embodiments, and language without requiring dense temporal annotations. If present in the frozen training set, no-action clips are treated as an independent training sample type rather than being merged into atomic-action supervision.

Order	Implementation rule	Trace effect
Validity and no-action handling	Remove missing or non-positive intervals and meaningless spans shorter than 0.5s. Remove other no-action spans, extending an adjacent retained action when the no-action endpoint is within 6s	Preserve contextual time without emitting idle or transition descriptions as atomic actions
Consecutive consolidation	Merge adjacent semantically similar descriptions; the strict documented setting disables cross-verb low-level grouping and does not replace labels with ancestor captions	Reduce duplicate actions without treating the hierarchy as boundary supervision
Collapse guard	If a source trace has at least two actions but consolidation leaves fewer than two, restore the source trace before later refinement	Prevent local rules from collapsing a multi-action trace into a single action
Local boundary reconciliation	Use a gap limit of 3s for $T < 30$s, 7s for $30 \leq T < 60$s, 8s for $60 \leq T < 120$s, and 15s otherwise. Close gaps up to the limit, bisect gaps up to twice the limit, and trim overlaps; discard a trimmed interval no longer than 0.05s	Produce ordered, non-overlapping intervals after local consolidation
Motor-fragment suppression	Remove a segment no longer than 2.6s when its leading verb belongs to the low-actionability motor set and its description contains no task verb; grasp, grab, and lift are excluded from that set	Suppress short execution phases while protecting brief object-acquisition actions
Completion-phase consolidation	If a suppressed release follows attach, drop, hang, insert, place, put, or return, extend the previous action through the release endpoint	Treat release as completion evidence for the preceding task action; the implementation does not test object identity
Final gap closure	Extend the preceding retained action across a positive gap of at most 3.0s for videos shorter than 30s, or 7.5s otherwise	Remove short holes introduced by fragment suppression while leaving larger gaps explicit

Table 6: Implementation details for the documented deterministic refinement variant. The operations consume the predicted trace and hierarchy metadata, not raw video. Main text abstracts them as action dependency resolution and temporal boundary reconciliation.

The submitted Robot Caption VLM is trained on a corpus of two million training samples together with a held-out validation set of 20,000 samples used exclusively for monitoring training progress and checkpoint selection. During training, fixed-boundary atomic-action clips and concatenated multi-action clips are sampled with a fixed ratio of 1:3. This sampling strategy is adopted for the submitted model to expose the captioner to both semantically complete single operations and temporally ordered multi-action intervals, and should be regarded as an implementation choice rather than a claim of universal optimality.

For multi-action supervision, temporally consecutive atomic annotations are concatenated only within the same episode and never across unlabeled intervals or explicit no-action regions. The resulting captions always preserve the original temporal ordering of the constituent actions.

Meaning-preserving caption diversification. Robot manipulation datasets are typically collected in constrained environments, resulting in repetitive tasks, similar motion trajectories, and highly homogeneous language annotations. To improve linguistic diversity while preserving semantic consistency, captions are diversified without changing the operation, manipulated object, target, movement direction, or resulting state. Multi-action clauses are diversified independently before restoring their original temporal order. The diversification procedure never introduces unseen objects, changes semantic granularity, or adds actions absent from the source annotation.

C.2 RoboActionTrace Full-Video Benchmark

This section provides additional construction details of the RoboActionTrace full-video benchmark introduced in the main paper. The benchmark contains 100 robot-manipulation videos collected from five public datasets. Every video lasts at least 20 seconds, contains at least two positive atomic actions, and provides complete temporal annotations without unlabeled intervals.

Video selection balances data sources, task categories, scenes, video duration, action cardinality, frame rate, spatial resolution, and annotation provenance. Benchmark construction does not use model predictions, evaluation scores, or downstream task performance.

The resulting benchmark contains 596 manually annotated atomic actions spanning 85 manipulation tasks across 100 videos. Every selected video passes timeline verification, video decoding, and manual quality inspection before inclusion in the benchmark.

Source	Videos	Atomic actions
AgiBotWorld	21	168
OpenGalaxea	22	133
RoboCoin	21	57
RoboInter-Data	26	167
RoboMind2.0	10	71
Total	100	596

Table 7: Source composition of the RoboActionTrace full-video benchmark.

This 100-video set is used for full Video-to-Trace evaluation. The fixed-boundary caption sets associated with the RoboActionTrace benchmark family are separate evaluation sets and are described in Section C.4. In particular, they should not be interpreted as clips cropped from the 100 full-video benchmark videos.

C.3 WGO Full-Video Evaluation Splits

WGO Robot75 contains 75 robot-manipulation videos and is used for full Video-to-Trace evaluation. Together with the RoboActionTrace full-video benchmark, it forms the two robot-domain full-video evaluation sets used in the pipeline comparison.

WGO Human25 contains 25 egocentric human-hand videos and is evaluated independently as a cross-embodiment generalization setting. Robot75 and Human25 are never merged into a single WGO score, and neither split is pooled with RoboActionTrace.

The fixed-boundary WGO caption sets are described separately in Section C.4. They evaluate caption generation under predefined temporal intervals rather than complete Video-to-Trace generation.

C.4 Fixed-Boundary Caption Evaluation Sets

Fixed-boundary evaluation addresses a different question from full Video-to-Trace evaluation. Each model is provided with the same predefined video interval, thereby isolating caption quality, action coverage, and output cardinality from temporal segmentation and boundary-selection errors.

We evaluate four robot captioning sets, grouped by benchmark family: RoboActionTrace Atomic100 and RoboActionTrace Concat100, together with WGO Robot-Atomic100 and WGO Robot-MultiAction100.

RoboActionTrace Atomic100 contains 100 clips, each covering exactly one positive atomic action. RoboActionTrace Concat100 contains 100 clips, each covering two or more temporally consecutive atomic actions. These two sets are constructed as separate caption-evaluation sets within the RoboActionTrace benchmark family rather than by cropping clips from the 100-video full-video benchmark. Multi-action clips are constructed only from consecutive annotations within the same episode and never cross an unlabeled interval or an explicit no-action region.

WGO Robot-Atomic100 and WGO Robot-MultiAction100 contain 100 fixed-boundary clips each and are sampled from WGO-Bench for independent caption evaluation. The atomic set contains one positive atomic action per interval, whereas the multi-action set contains multiple temporally consecutive actions. These sets are evaluated separately from the full-video WGO Robot75 split; they do not measure complete Video-to-Trace generation.

Within each benchmark family, clip selection balances the available data sources, action families, clip durations, and temporal positions within the parent video. Every selected clip passes materialization and full-decode validation to ensure that the complete annotated interval is available and decodable. Human4D results are reported separately for all four evaluation sets; scores are not pooled across benchmark families or between atomic and multi-action settings.

C.5 Automatic Metrics and Evaluation Protocol

All automatic full-pipeline evaluations use the same deterministic one-to-one temporal matching implementation, such that each predicted segment and each ground-truth segment can participate in at most one matched pair. Segment-level true positives, false positives, and false negatives are accumulated over the complete evaluation set before precision, recall, and F1 are computed, corresponding to micro aggregation.

All automatic full-pipeline comparisons use `endpoint_mode=raw`, `matching_mode=legacy_wgo_greedy_desc`, and micro aggregation. The evaluator uses the start and end times produced by each complete pipeline. The same temporal matching implementation, IoU thresholds, and aggregation rules are applied to all methods within each evaluation set.

Temporal mIoU@.75. Let $M_{0.75}$ denote the set of one-to-one matched prediction–ground-truth pairs whose temporal IoU is at least 0.75. Temporal mIoU@.75 is defined as

$$\text{mIoU@.75} = \frac{1}{|M_{0.75}|} \sum_{(p,g)\in M_{0.75}} \text{IoU}(p, g). \tag{4}$$

The score is set to zero when $M_{0.75}$ is empty. This metric measures boundary precision among successfully matched actions, but does not penalize unmatched predictions or ground-truth actions.

Segment F1@.5. A one-to-one matched pair with temporal IoU of at least 0.5 is counted as a temporal true positive. Unmatched predictions and unmatched ground-truth actions are counted as false positives and false negatives, respectively:

$$P_{0.5} = \frac{TP_{0.5}}{N_{\text{pred}}}, \qquad R_{0.5} = \frac{TP_{0.5}}{N_{\text{gt}}}, \tag{5}$$

$$\text{Segment F1@.5} = \frac{2P_{0.5}R_{0.5}}{P_{0.5} + R_{0.5}}. \tag{6}$$

This metric jointly penalizes missed, fragmented, and spurious action segments.

Label Accuracy@.75. Semantic label accuracy is evaluated only on temporal matches with IoU of at least 0.75 and a valid semantic-judge decision:

$$\text{Label Acc.@.75} = \frac{N_{\text{label match}}}{N_{\text{valid judge}}}. \tag{7}$$

Missing or erroneous judge outputs are excluded from the denominator and are reported separately. In the default-input WGO75 comparison in Table 1, Refiner and ROVER have one and four missing semantic-judge pairs, respectively. This metric measures conditional semantic accuracy after strict temporal alignment, but does not penalize unmatched segments.

Semantic E2E F1@.75. A semantic true positive must simultaneously satisfy three conditions: temporal IoU of at least 0.75, a valid judge decision, and a positive semantic match. Semantic precision, recall, and end-to-end F1 are defined as

$$P_{\text{sem}} = \frac{TP_{\text{sem}}}{N_{\text{pred}}}, \qquad R_{\text{sem}} = \frac{TP_{\text{sem}}}{N_{\text{gt}}}, \tag{8}$$

$$\text{Semantic E2E F1@.75} = \frac{2P_{\text{sem}}R_{\text{sem}}}{P_{\text{sem}} + R_{\text{sem}}}. \tag{9}$$

Missing or erroneous judge outputs never count as semantic true positives. Boundary-only methods are evaluated using temporal metrics, while their semantic fields are reported as N/A.

C.6 Semantic Judge and Human Review

Semantic judge protocol. After temporal matching is fixed, the semantic judge receives the sample-level task instruction when available, the ground-truth action label, and the predicted action label. It returns a binary semantic-match decision.

A match requires the prediction to describe the same operation event with the correct core verb and manipulated object. Source, target, movement direction, or resulting state must also be correct when these attributes are necessary to distinguish the event. Minor wording and synonym differences are ignored. Wrong objects, reversed source–target relations, missing action-critical directions, overly vague descriptions, and hallucinated key objects are treated as mismatches.

Fixed-boundary Human4D review. Each anonymized caption receives four scores from 0 to 10: absence of hallucination, action completeness, unambiguous object reference, and spatial-information completeness. Human4D is the arithmetic mean of these four dimensions. Model identities are hidden during review, while the frozen assignment-to-model mapping preserves the traceability of all reported scores.

Human4D evaluates overall caption quality but does not provide an adjudicated ordered action list or per-action semantic slots. It therefore cannot be converted into action-level precision, recall, ordering accuracy, or action-cardinality accuracy without additional structured annotation.

Full Video-to-Trace blind review. Each complete pipeline is evaluated on all 100 videos in the RoboActionTrace Benchmark. Reviewers independently score segmentation and labeling quality on separate 0–10 scales. Boundary-only methods receive segmentation scores but no labeling or overall score. The over-fine rate records the proportion of videos judged to contain excessively fragmented segmentation or descriptions that are finer than the target atomic-action granularity.

C.7 Why Human Review Is Necessary for Robot Videos

Robot manipulation actions typically unfold gradually through approach, contact, grasp, lift, stabilization, transport, placement, and release. The temporal extent of a semantically complete operation may therefore admit more than one reasonable boundary choice. For example, a pick-up action may reasonably begin at late approach or at initial contact, while still covering the same complete object-state change.

A frozen ground-truth interval provides a consistent reference for large-scale automatic evaluation, but it represents one annotation convention rather than the only semantically defensible interval. IoU-based metrics consequently measure agreement with the frozen temporal annotation, whereas video-grounded human review assesses whether the predicted segmentation captures a complete, coherent, and appropriately granular manipulation process.

Automatic and human evaluation therefore measure complementary properties. Automatic metrics provide reproducible temporal and semantic comparisons at scale, while human review captures reasonable boundary variation, semantic completeness, and fragmentation that may not be fully reflected by a single IoU threshold.

D Benchmark Analysis and Full-Pipeline Evaluation

This section provides additional analyses of the benchmark characteristics and the full Video-to-Trace evaluation presented in the main paper. We first compare RoboActionTrace and WGO Robot75 to highlight the differences in temporal horizon, action distribution, visual quality, and task instructions. We then use these benchmark characteristics together with the official input contracts of the compared methods to interpret the observed temporal and semantic performance of different Video-to-Trace pipelines.

D.1 Benchmark Characteristics

As introduced in the main paper, RoboActionTrace and WGO Robot75 evaluate the same Video-to-Trace task but differ substantially in temporal horizon, action cardinality, action density, image

Benchmark	Videos	Avg. duration	Avg. actions	Actions/min	Resolution distribution	Task instruction
RoboActionTrace	100	86.6 s	5.96	4.13	26 at 320×180; 38 at 640×480; 36 at 1280×720	Mostly task-level; connective form in 23/100
WGO Robot75	75	25.9 s	3.64	8.43	50 at 320×180; 25 at 1280×720	More operation-explicit; connective form in 47/75

Table 8: Characteristics of the two robot benchmarks. Connective-form instructions provide a lexical proxy for explicit operation structure rather than a direct measure of semantic coverage.

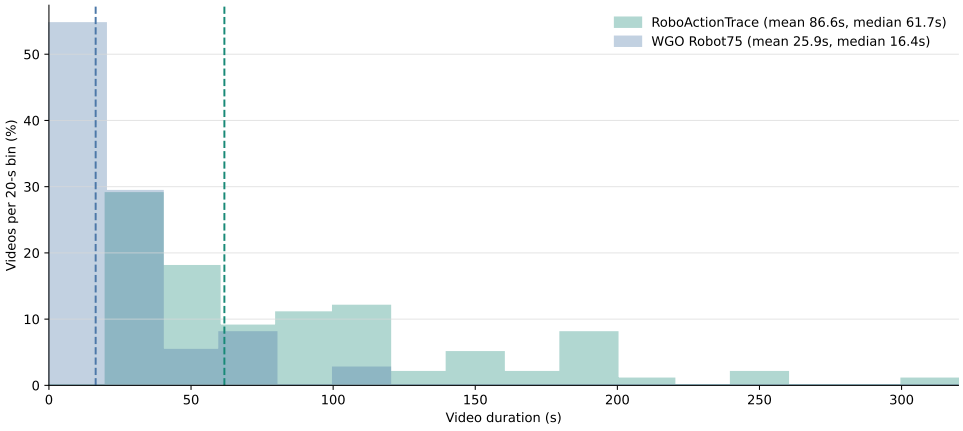


Figure 4: Video-duration distributions. RoboActionTrace averages 86.6 s with a median of 61.7 s, whereas WGO Robot75 averages 25.9 s with a median of 16.4 s.

resolution, and task-instruction characteristics. These factors influence different aspects of temporal localization and semantic prediction and therefore provide the context needed to interpret the full-pipeline results reported in the main paper.

RoboActionTrace contains 100 robot-manipulation videos with 596 manually annotated atomic actions, averaging 86.6 seconds and 5.96 actions per video. In comparison, WGO Robot75 contains 75 robot-manipulation videos with 273 annotated actions, averaging 25.9 seconds and 3.64 actions per video.

Compared with WGO Robot75, RoboActionTrace requires substantially longer temporal reasoning together with recovery of more atomic actions per video. It therefore increases the difficulty of long-range search, action enumeration, temporal ordering, and suppression of non-action fragments. WGO Robot75, in contrast, contains shorter demonstrations with fewer target actions, reducing the overall temporal search space. However, its actions are more densely distributed in time, making local endpoint placement under strict IoU thresholds comparatively more sensitive. These differences alter the sources of evaluation difficulty rather than making either benchmark uniformly easier.

The two benchmarks also differ visually and linguistically. WGO Robot75 contains a substantially larger proportion of low-resolution videos while more frequently providing task instructions that explicitly lexicalize constituent operations. These properties influence semantic prediction differently and are analyzed in Section D.4.

The following sections interpret the observed full-pipeline behavior in light of these benchmark characteristics.

Figure 6 jointly visualizes temporal horizon and action cardinality at the video level. Unlike the two marginal distributions above, it reveals the combined search space encountered during full Video-to-Trace generation.

Taken together, these benchmark statistics indicate that RoboActionTrace primarily stresses long-horizon search, action enumeration, temporal ordering, and non-action suppression, whereas WGO Robot75 reduces the global search space while remaining sensitive to local boundary placement

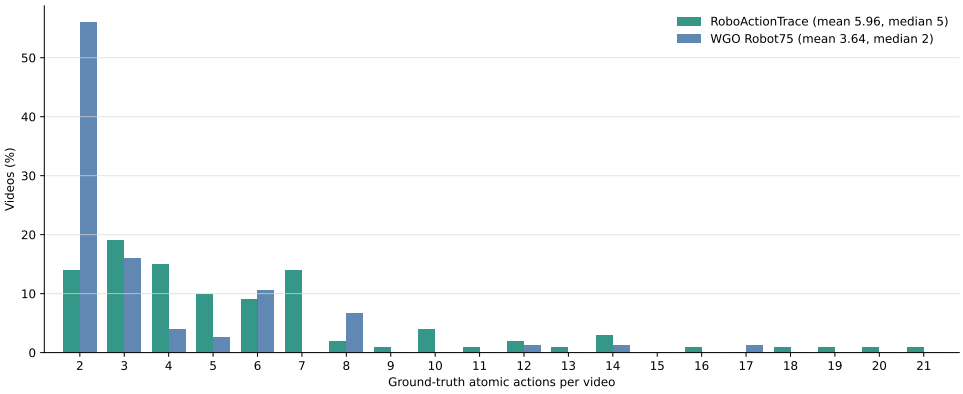


Figure 5: Ground-truth atomic-action counts per video. RoboActionTrace has a mean/median of 5.96/5 actions, compared with 3.64/2 for WGO Robot75.

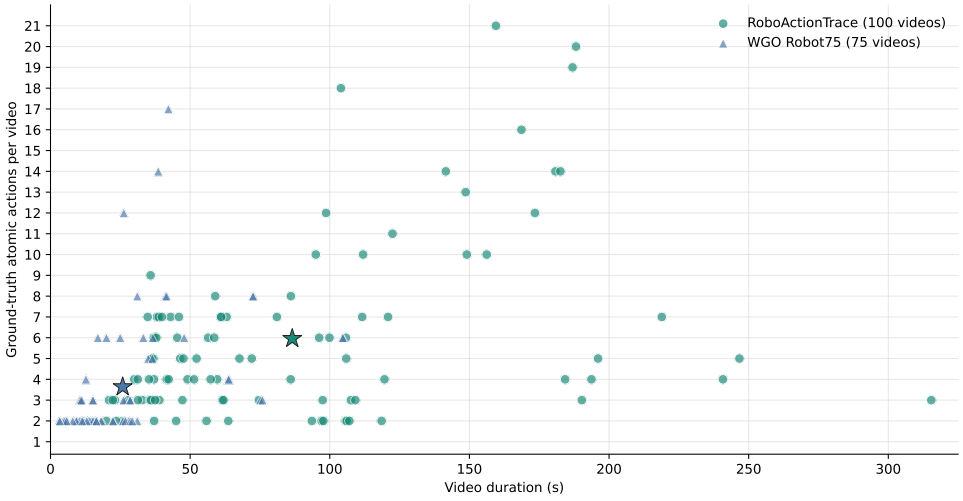


Figure 6: Joint distribution of video duration and ground-truth atomic-action count. Each point represents one video, and stars denote the benchmark means. RoboActionTrace simultaneously spans longer temporal horizons and higher action cardinality, whereas WGO Robot75 concentrates around shorter demonstrations with fewer target actions despite several action-dense cases.

and semantic ambiguity. The following sections explain how these characteristics contribute to the temporal and semantic behavior of different Video-to-Trace pipelines.

D.2 Official Method Settings

The compared methods are evaluated under the settings specified in their original formulations rather than a unified input protocol. As summarized in Table 9, they differ in visual input, temporal sampling, access to sample-level task instructions, available temporal context, and authority to determine temporal boundaries. Consequently, the reported results compare complete annotation pipelines under their official settings instead of a strictly same-input evaluation.

These differences should be taken into account when interpreting the benchmark results, particularly when comparing methods that receive sample-level task instructions with those operating solely from RGB video. For instruction-conditioned pipelines, the semantic prior available to the model depends on the granularity of the provided task description. Representative examples illustrating this effect are included in the accompanying media supplement.

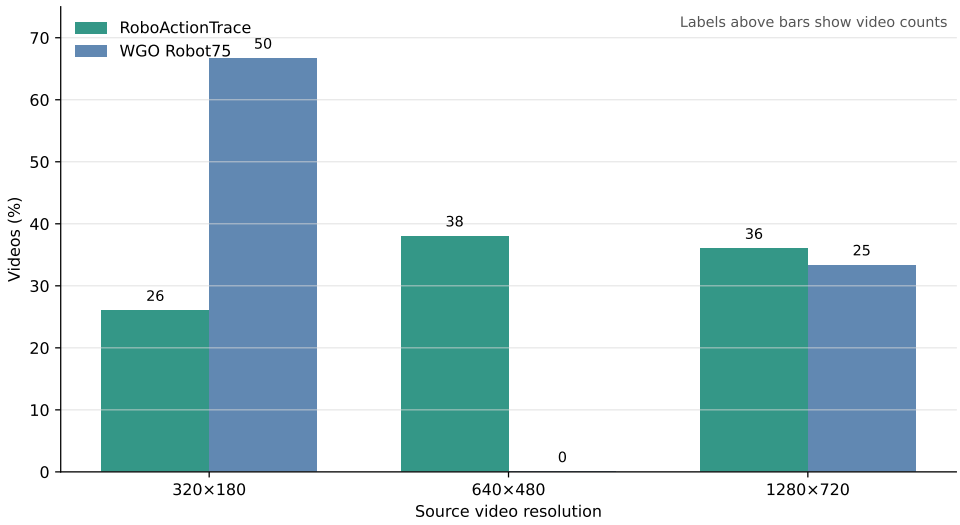


Figure 7: Resolution distributions of the two robot benchmarks. WGO Robot75 contains 50 videos at 320×180 and 25 at 1280×720, whereas RoboActionTrace contains 26 videos at 320×180, 38 at 640×480, and 36 at 1280×720. Their relationship to semantic prediction is analyzed in Section D.4.

Method	Visual input	Temporal sampling	Instruction & temporal context	Boundary authority
TraceCut	Stage-1 candidate clips (1–30 s) followed by text-only hierarchy aggregation	Stage 1 samples every fourth decoded frame ($F/4$); Robot Caption VLM uses native, 8, or 4 FPS with at most 240 frames	No sample-level task instruction, timestamp overlay, or whole-video visual context	Reuses candidate intervals; no free continuous-time regression
Refiner Macrodata Labs (2026)	Whole-video contact sheets generated from 2-Hz sampled frames	Fixed 2 Hz; 224-pixel tiles	Sample-level task instruction, timestamp badges, and whole-video context	Directly predicts complete action traces from timestamped contact sheets
TREAD Kuramshin et al. (2026)	First frame together with the complete target video and few-shot demonstrations	Native provider-side preprocessing (effective temporal sampling unknown)	Sample-level task instruction, full-video context, and granularity examples	Directly predicts full-video temporal intervals
ROVER Schroeder et al. (2026)	Thirty uniformly sampled frames followed by recursive local reasoning	Sparse recursive sampling at source-dependent resolution	Sample-level task instruction and recursive progress state	Maps recursive frame indices to continuous timestamps

Table 9: Official method settings used in the benchmark comparison. Provider-side preprocessing is reported as unknown when it cannot be observed.

Full-pipeline results are reported in Table 1 of the main paper. The following analyses discuss their relationship to benchmark characteristics and method inputs.

D.3 Temporal Difficulty across Benchmarks

The benchmark characteristics summarized in Section D show that RoboActionTrace contains substantially longer videos and more atomic actions per episode than WGO Robot75. Longer temporal horizons increase the opportunity for missed actions, duplicate predictions, incorrect action ordering, and accumulation of local boundary errors during Video-to-Trace generation.

Consistent with these benchmark characteristics, all evaluated methods obtain higher Segment F1@.5 on WGO Robot75 than on RoboActionTrace under the same evaluation protocol. This consistent pattern indicates that the shorter temporal horizon and lower action cardinality of WGO Robot75 reduce the difficulty of long-range search and action enumeration.

Nevertheless, WGO Robot75 remains challenging under strict temporal metrics because its actions are more densely distributed in time. Small boundary deviations between adjacent actions can there-

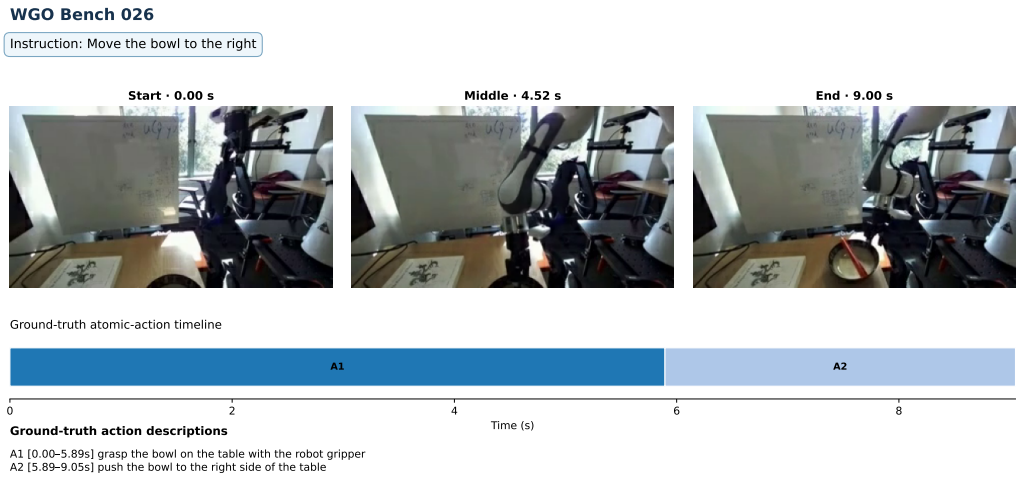


Figure 8: WGO026: a goal-level instruction expands into a two-action ground-truth trace.

fore produce substantial changes in temporal IoU, even when the recovered action sequence is qualitatively correct.

Taken together, these observations suggest that temporal performance is primarily governed by the temporal search space defined by benchmark duration, action cardinality, and action density. The benchmark-specific behavior of individual pipelines is discussed in Section E.

D.4 Semantic Difficulty: Resolution and Instruction Priors

Unlike temporal localization, semantic prediction depends not only on temporal horizon but also on visual quality, task priors, ontology differences, and the official input contracts of the compared methods.

WGO Robot75 combines two competing characteristics. On one hand, a large proportion of its videos are available only at low spatial resolution, making object identity, target localization, and subtle state changes more difficult to recover visually. On the other hand, the official input contracts of Refiner Macrodata Labs (2026), TREAD Kuramshin et al. (2026), and ROVER Schroeder et al. (2026) provide both the input video and the corresponding sample-level task instruction. In many Robot75 examples, these instructions explicitly lexicalize one or more constituent operations, supplying additional semantic priors beyond the visual observations alone.

Figures 8–11 illustrate representative instruction-to-trace relationships. Some instructions specify only a high-level task that expands into multiple atomic actions, whereas others explicitly mention constituent pick-and-place operations or partially disclose the target trace. These examples demonstrate that task instructions often provide useful semantic priors while remaining distinct from complete atomic-action annotations.

The influence of instruction granularity is further illustrated in the accompanying media supplement. Representative long-horizon examples show that instruction-conditioned pipelines become noticeably more prone to missed or merged action boundaries when only concise task-level descriptions are available, compared with instructions that explicitly describe constituent operations. Although these examples are qualitative rather than a controlled ablation, they further suggest that the semantic prior provided by task instructions can materially affect temporal decomposition.

Taken together, these observations suggest that semantic performance on WGO Robot75 should be interpreted jointly with benchmark characteristics and official method input contracts rather than visual evidence alone. Low-resolution perception, task priors provided by sample-level instructions,

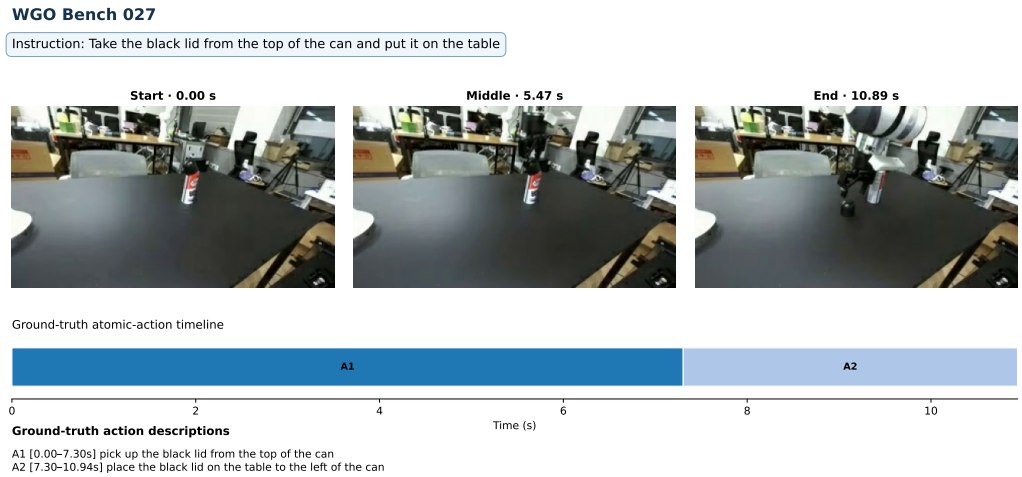


Figure 9: WGO027: the instruction explicitly lexicalizes the take and put operations appearing in the annotated trace.

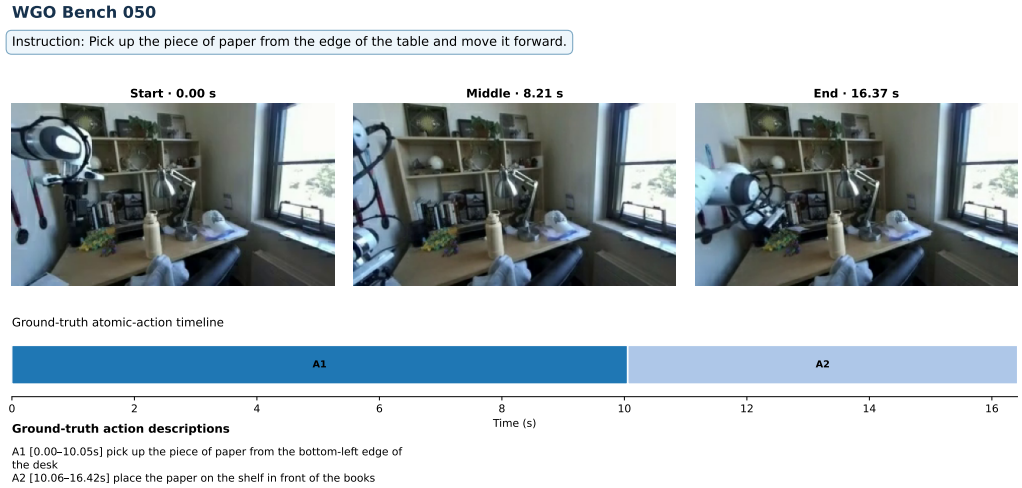


Figure 10: WGO050: the instruction partially specifies the trace, while the annotation resolves a more specific placement target.

caption granularity, ontology differences, and domain shift all contribute to the observed semantic behavior across methods.

The current evidence remains correlational rather than causal because benchmark characteristics, visual resolution, ontology, task instructions, and model input contracts vary simultaneously. The instruction-removal experiment in Section 5.2 measures the sensitivity of the evaluated pipelines to task descriptions. It does not separately identify the contribution of operation-explicit wording, nor isolate the effects of visual resolution, annotation conventions, or domain shift.

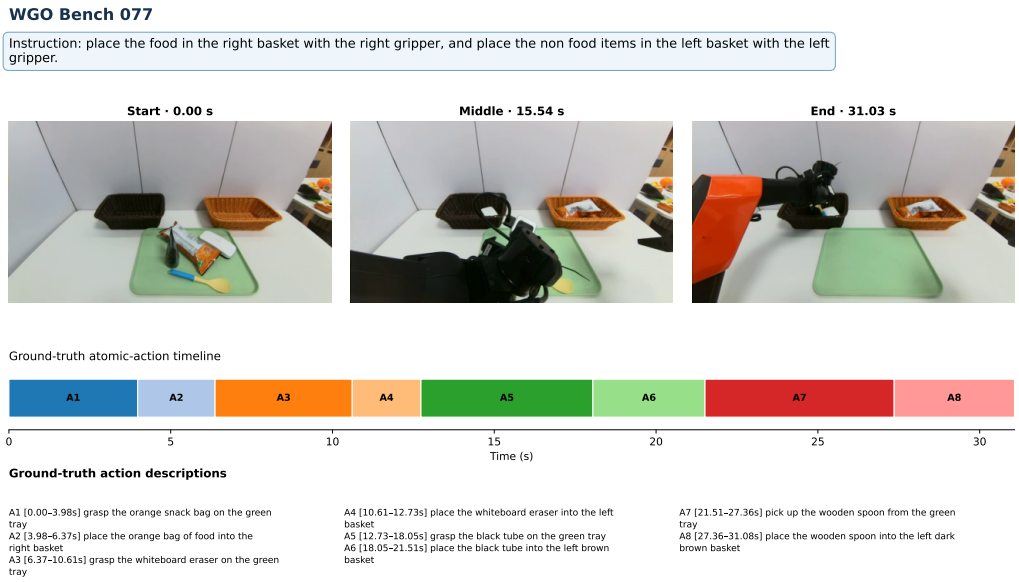


Figure 11: WGO077: one category-level sorting instruction corresponds to multiple object-specific grasp-and-place actions.

D.5 Pipeline-specific Interpretation

The benchmark results should be interpreted together with the official method settings summarized in Table 9. Different pipelines receive different forms of visual and textual information and therefore solve slightly different variants of the Video-to-Trace problem.

Instruction-conditioned end-to-end pipelines. Refiner Macrodata Labs (2026), TREAD Kuramshin et al. (2026), and ROVER Schroeder et al. (2026) receive both the input video and the corresponding sample-level task instruction as part of their official settings. These methods therefore perform instruction-conditioned Video-to-Trace generation rather than purely visual action discovery. As discussed in Section D.4, many WGO Robot75 instructions explicitly lexicalize one or more constituent operations, providing additional semantic priors beyond the visual observations. These priors help reduce semantic ambiguity, although the three methods adopt different reasoning strategies for temporal localization.

TraceCut. Unlike the instruction-conditioned pipelines, TraceCut operates exclusively on RGB video and does not assume the availability of sample-level task instructions. This design allows the same pipeline to be applied to datasets in which task descriptions are unavailable, incomplete, or unreliable. Consequently, semantic prediction depends entirely on visual observations together with the Multi-Scale Candidate Hierarchy, Robot Caption VLM, and Actionability-aware Hierarchy-to-Trace Decoding. The benchmark results therefore reflect the general Video-to-Trace setting targeted by TraceCut rather than an instruction-conditioned variant of the task.

Candidate view	Candidates/GT	Best IoU	Recall@.3	Recall@.5	Recall@.7
Full captionable hierarchy (1–30s)	11.68×	0.7383	0.9883	0.9050	0.6269
Atomic band (2–12s)	6.35×	0.6845	0.9625	0.8118	0.5228
Coarse band (8–30s)	1.47×	0.5091	0.7343	0.4773	0.2871
Dual-level band (2–24s)	7.05×	0.7273	0.9750	0.8835	0.6133
Budgeted action frontier	1.35×	0.5095	0.8190	0.4829	0.2190

Table 10: Candidate-pool analysis. Flat budgeted clustering loses too many true actions. The useful structure is a high-recall atomic–coarse candidate view followed by actionability-aware decoding.

Trace source	Time IoU ↑	Object R ↑	Verb R ↑	Halluc.↓	Count ratio
Adaptive-gap final trace	<u>0.4234</u>	0.6567	<u>0.7019</u>	0.4100	1.1551
Naive atomic frontier	0.2850	<u>0.6963</u>	0.4792	0.5820	1.9221
Naive coarse frontier	0.3031	0.5451	0.3537	<u>0.4036</u>	0.5531
Oracle over captioned hierarchy	**0.7383**	**0.9593**	**0.9563**	**0.0000**	**0.9958**

Table 11: Hierarchy oracle analysis. The captioned segment hierarchy contains high-quality candidates, but naive frontiers fail by over-fragmenting or under-segmenting. The central problem is selecting an actionable trace from redundant nodes.

D.6 Human Review and Automatic Metrics

Automatic metrics and human review measure complementary properties of Video-to-Trace generation. Automatic metrics quantify agreement with frozen temporal annotations and semantic labels, whereas human reviewers evaluate whether the predicted trace forms a complete, coherent, and interpretable description of the manipulation process.

This distinction is particularly important for robot manipulation, where semantically complete actions often evolve gradually through approach, contact, grasp, transport, placement, and release. Multiple temporal intervals may therefore correspond to equally reasonable interpretations of the same operation. Consequently, automatic and human rankings need not coincide even when both evaluations are internally consistent.

E Additional Pipeline Analysis

This section analyzes the two pipeline components surrounding Robot Caption VLM: the Multi-Scale Candidate Hierarchy and Actionability-aware Hierarchy-to-Trace Decoding. Unless otherwise stated, dataset construction, candidate features, Robot Caption VLM outputs, and evaluation protocols are held fixed so that each experiment isolates the component under study.

E.1 Candidate Recall and Hierarchy Oracle

This auxiliary RoboActionTrace experiment analyzes the temporal support provided by the Multi-Scale Candidate Hierarchy before semantic decoding. It is a mechanism-level diagnostic of Stage 1 candidate generation rather than a benchmark result. Candidate recall measures whether at least one candidate node overlaps each ground-truth action above a given IoU threshold; it does not evaluate whether the downstream decoder can successfully identify that node.

The candidate recall results show that the full hierarchy and the dual-level view retain substantially more high-IoU candidates than compact fixed frontiers. This supports the central design choice of postponing semantic selection until node captions and cross-scale hierarchy context become available. Compact hierarchy views reduce the candidate budget but may irreversibly discard temporally accurate intervals before semantic reasoning begins.

The hierarchy oracle further estimates the upper bound of Stage 1 by selecting, for each ground-truth action, the candidate with the highest temporal overlap. Because this selection requires ground-truth annotations, it is not deployable. Instead, it separates two sources of error: insufficient temporal support in the candidate hierarchy and imperfect candidate selection by the subsequent caption-and-

Method	Segmentation	Labeling	Over-fine rate
TraceCut	**5.44**	**4.92**	**0%**
Refiner	5.04	4.52	8%
TREAD	3.00	3.64	20%
ROVER	1.44	2.40	**0%**
UVD+VIP	1.48	N/A	96%
NILS-adapted	1.44	N/A	88%

Table 12: Human evaluation on WGO Human25. Segmentation and labeling are rated independently on a 0–10 scale. Boundary-only methods do not predict semantic labels and therefore receive N/A for labeling.

decoding stages. Consequently, a large oracle gap indicates remaining decoder headroom rather than an attainable system result.

E.2 Decoder Ablation

We evaluate the deterministic refinement stages on RoboActionTrace while holding the Stage 1 hierarchy, Stage 2 captions, hierarchy-conditioned text-LLM outputs, and decoder parameters fixed. The initial trace already includes the common strict-JSON parser and adjacent-duplicate normalization shared by every variant; it is therefore not a raw language-model output.

The cumulative decoder results and confidence intervals are reported in Table 4 and the corresponding Ablation Studies section of the main paper, using the evaluation protocol specified in Section C.5.

The gains are cumulative rather than additive because later refinement stages operate on traces already modified by earlier stages. Semantic metrics are not reported for intermediate variants because frozen, pairwise-consistent semantic-judge decisions are unavailable for every altered matching configuration.

F Cross-Embodiment Evaluation on WGO Human25

WGO Human25 is evaluated independently as a cross-embodiment setting. Compared with the robot benchmarks, it differs in embodiment, camera viewpoint, motion characteristics, and action distribution. Consequently, Human25 results are reported separately and are never pooled with RoboActionTrace or WGO Robot75.

F.1 Evaluation Setup

Human25 contains 25 egocentric human-hand videos. Since Robot Caption VLM is trained exclusively on robot-manipulation data, Stage 2 is replaced with Gemini-3.5-Flash for this cross-embodiment evaluation. The remaining stages, including the Multi-Scale Candidate Hierarchy and Actionability-aware Hierarchy-to-Trace Decoding, are kept unchanged.

Refiner, TREAD, and ROVER retain their official settings. UVD+VIP and NILS-adapted remain boundary-only methods and therefore receive N/A for semantic labeling metrics. Human review is performed independently on Human25 and is never pooled with the robot-domain benchmarks.

F.2 Human25 Results

Table 12 summarizes the human evaluation on WGO Human25. Using Gemini-3.5-Flash as the Stage 2 caption model, TraceCut achieves the highest segmentation score (5.44) and the highest labeling score (4.92) among all evaluated methods. Compared with the strongest baseline, Refiner, segmentation improves from 5.04 to 5.44, while labeling improves from 4.52 to 4.92. TraceCut also produces the most appropriate action granularity. Its over-fine caption rate is 0%, compared with 8% for Refiner, 20% for TREAD, 96% for UVD+VIP, and 88% for NILS-adapted.

These results suggest that the proposed Multi-Scale Candidate Hierarchy together with Actionability-aware Hierarchy-to-Trace Decoding transfers effectively from robot-manipulation



(a) Placement-like operation (b) Completion-phase release (c) Protected short press

Figure 12: Representative hierarchy-to-trace cases. The first example shows selection of a semantically complete placement interval. The second illustrates completion-phase consolidation, where a short release is absorbed into the preceding placement-like operation. The third shows preservation of a brief but task-relevant state-changing press.

videos to egocentric human-hand videos. Although the Stage 2 caption model is replaced with a general-purpose VLM, the overall pipeline remains competitive under substantial embodiment and viewpoint changes.

G Qualitative Analysis and Limitations

G.1 Representative Qualitative Cases

Figure 12 illustrates the intended behavior of Actionability-aware Hierarchy-to-Trace Decoding. The decoder favors complete state-changing operations, consolidates short non-independent completion phases, and avoids suppressing brief but meaningful actions solely because of their duration.

G.2 Representative Failure Mode

The dominant failure mode of the current pipeline originates from incorrect node-level captions produced by Robot Caption VLM. When an incorrect action description is assigned to a candidate node, the subsequent hierarchy-to-trace decoder does not revisit the raw video and therefore has no additional visual information with which to recover the correct semantics. During text-LLM aggregation, the incorrectly captioned node may be retained as an independent atomic action. Deterministic refinement can merge or suppress structurally redundant predictions, but it cannot reliably correct an incorrect object, target, direction, or action label. Consequently, node-level caption errors can propagate directly to the final action trace.

G.3 Limitations

The current pipeline depends on the quality and coverage of the Multi-Scale Candidate Hierarchy and Robot Caption VLM. An action that is not adequately represented by any candidate interval, or is consistently assigned an incorrect caption, cannot be reliably recovered by the text-only aggregation and deterministic refinement stages.

The benchmark comparisons also involve simultaneous differences in video resolution, task instructions, temporal interfaces, and method inputs. The corresponding analyses should therefore be interpreted as descriptive rather than fully causal unless these factors are isolated through controlled experiments.

H Qualitative Caption Comparisons

Figures 14-15 show two examples from each of the four fixed-boundary evaluation subsets. For every video, the left panel contains eight frames sampled uniformly over time, and the right panel places the ground-truth reference first, followed by the model captions and their Human4D scores. The score is the mean of the four human-rated dimensions: hallucination, missing action, object-

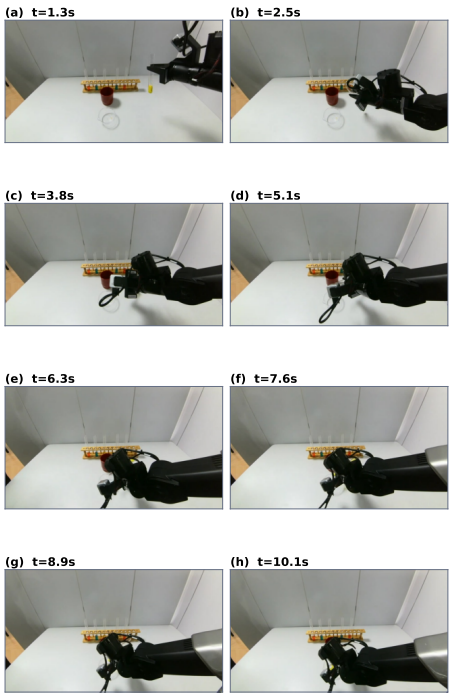
reference ambiguity, and missing spatial information. Blue and orange rows denote Ours(8B) and Ours(27B), respectively.

WGO Robot-Atomic100. The WGO atomic cases emphasize precise action and destination recognition. In Figure 13, both of our models distinguish pouring from merely placing the test tube and identify the receiving beaker. In Figure 14, they correctly describe placing the blue cup into the left basket. Both models receive 10.0 on both examples.

WGO Robot-MultiAction100. In Figure 15, both of our models preserve the full pick-up–pour–return sequence, whereas several baselines describe only the central pouring action. Figure 16 contains a longer sorting sequence. Ours(8B) retains the ordering, object distinctions, and destination baskets, while Ours(27B) remains action-complete but uses somewhat less precise object names. This example also highlights why uniformly sampled frames are useful for multi-action clips: different sub-actions occur at widely separated times.

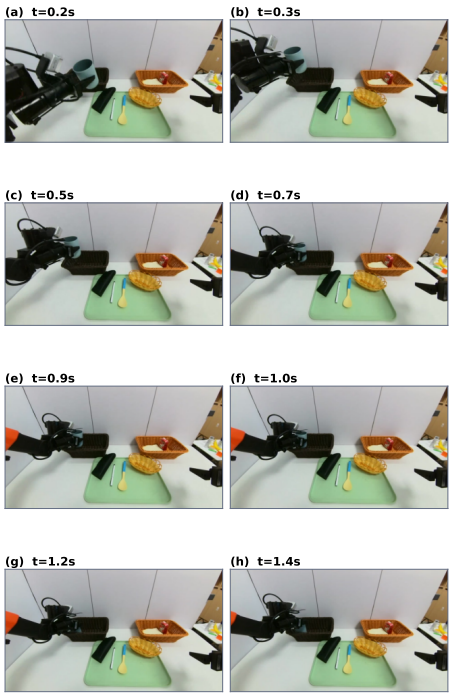
RoboActionTrace Atomic100. The two atomic examples test whether a caption identifies a short and visually subtle state change. Both of our models correctly ground the acting arm and the microwave-door closing action in Figure 17. In Figure 18, they capture the release event rather than describing the static scene or the preceding clothes-handling action; Ours(8B) scores 10.0 and Ours(27B) scores 9.5.

RoboActionTrace Concat100. Figures 19 and 20 require captions to preserve multiple object interactions and their temporal order. Our models cover all three phases of the teapot manipulation—pick up, pour, and set down—and retain both objects and the bag destination in the takeout example. This action completeness accounts for their perfect Human4D scores on both examples.



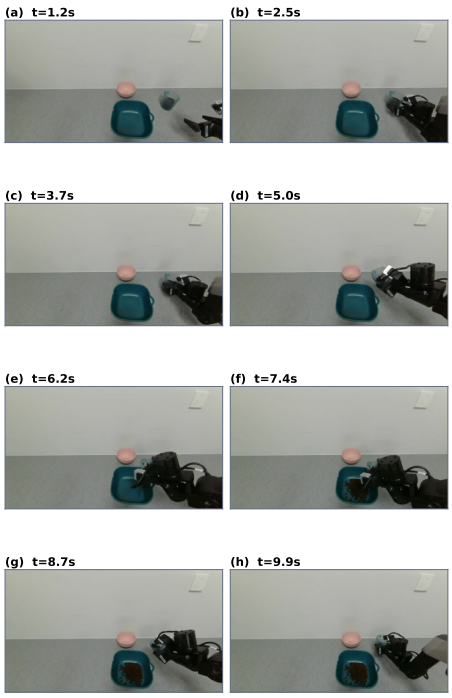
Ground truth (Reference)	pour the yellow pigment from the test tube into the beaker
Seed-2.0-Pro 10.0	Pick up the test tube with yellow liquid from the test tube rack; Pour the yellow liquid from the test tube into the transparent beaker.
Qwen3.6-Plus 10.0	Pour the yellow liquid from the test tube into the beaker.
Qwen3-VL-8B-Instruct 8.0	Pour liquid from the test tube into the beaker.
RynnBrain-8B 0.0	Place the test tube in the beaker
MiMo-Embodied-7B 8.0	Pour the yellow object into the beaker.
Ours(8B) 10.0	Pour the yellow solution from the test tube into the beaker using your right hand.
Ours(27B) 10.0	The right arm pours the yellow solution from the test tube into the beaker.

Figure 13: WGO Robot-Atomic100 example WGO 061: pour yellow pigment from a test tube into a beaker. Ours(8B) and Ours(27B) both receive 10.0.



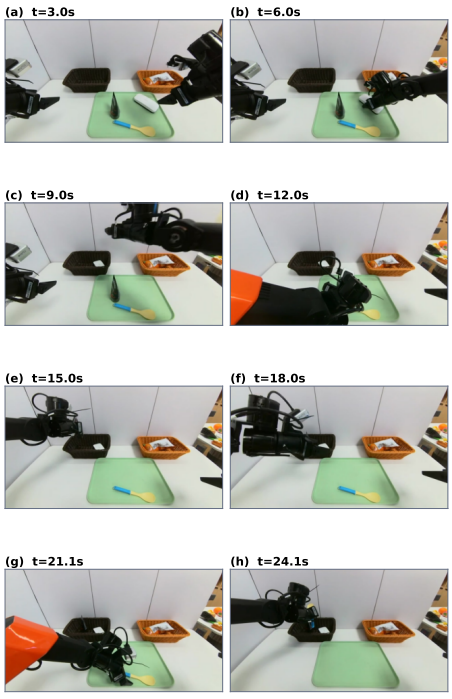
Ground truth (Reference)	drop the blue cup into the left brown basket
Seed-2.0-Pro 10.0	Place the light blue cup into the dark brown wicker basket.
Qwen3.6-Plus 10.0	Place the blue cup into the black basket.
Qwen3-VL-8B-Instruct 10.0	Place the blue cup into the brown basket.
RynnBrain-8B 9.5	Place the cup in the basket
MiMo-Embodied-7B 10.0	Place the blue cup into the dark brown basket.
Ours(8B) 10.0	The left arm places the blue cup into the left basket.
Ours(27B) 10.0	The left gripper places the cup into the dark basket.

Figure 14: WGO Robot-Atomic100 example WGO 058: place a blue cup into the left basket. Ours(8B) and Ours(27B) both receive 10.0.



Ground truth (Reference)	Step 1: pick up the glass with beans Step 2: pour the brown solids from the cup into the blue tray Step 3: place the glass next to the pink bowl
Seed-2.0-Pro 10.0	Pick up the transparent cup from the floor; Pour the dark pellets from the cup into the teal tray; Place the transparent cup on the floor.
Qwen3.6-Plus 8.5	Pour the contents of the glass cup into the teal tray.
Qwen3-VL-8B-Instruct 8.0	Pick up the measuring cup; Pour the kibble into the blue bowl.
RynnBrain-8B 8.0	Pour the kibble from the measuring cup into the blue container.
MiMo-Embodied-7B 8.5	Pour the glass contents into the blue tray
Ours(8B) 10.0	The right arm picks up the blue cup on the right side of the table.; The right arm pours the food from the blue cup into the blue tray on the left side of the table.; The right arm places the blue cup back on the right side of the table.
Ours(27B) 10.0	The right arm picks up the rightmost cup.; The right arm pours the contents of the rightmost cup into the middle blue tray.; The right arm places the rightmost cup back on the table.

Figure 15: WGO Robot-MultiAction100 example WGO 088: pick up a cup, pour its contents, and return it. Ours(8B) and Ours(27B) both receive 10.0.



Ground truth (Reference)	Step 1: place the orange bag of food into the right basket Step 2: grasp the whiteboard eraser on the green tray Step 3: place the whiteboard eraser into the left basket Step 4: grasp the black tube on the green tray Step 5: place the black tube into the left brown basket Step 6: pick up the wooden spoon from the green tray Step 7: place the wooden spoon into the left dark brown basket
Seed-2.0-Pro 9.0	Place the orange packet from the green tray into the orange wicker basket; Place the white rectangular object from the green tray into the dark brown wicker basket; Place the dark tube from the green tray into the dark brown wicker basket; Place the blue-handled wooden spoon from the green tray into the dark brown wicker basket.
Qwen3.6-Plus 9.0	Place the orange packet into the orange basket; Place the white object into the brown basket; Place the green bottle into the brown basket; Place the spoon into the brown basket.
Qwen3-VL-8B-Instruct 10.0	Place the orange packet into the orange basket; Pick up the white object from the green tray; Place the white object into the brown basket; Pick up the black object from the green tray; Place the black object into the brown basket; Pick up the blue and yellow spoon from the green tray; Place the blue and yellow spoon into the brown basket.
RynnBrain-8B 8.5	Place the packet into the basket; Place the white object into the basket; Place the brush into the basket
MiMo-Embodied-7B 0.0	Place the white object into the brown basket; Place the blue and yellow spatula into the black basket.
Ours(8B) 10.0	The right hand places the orange-packaged snack into the right-side basket.; Use the right hand to lift the white object and then place it into the left basket.; The left hand picks up the dark black item and places it in the left basket.; The left hand picks up the blue item and places it in the left basket.
Ours(27B) 9.5	Place the bread into the right basket.; Grasp the white box and move it into the left basket.; Pick up the wash and care set and place it in the left basket.; Lift the spoon and set it inside the left basket.

Figure 16: WGO Robot-MultiAction100 example WGO 055: sort food and non-food objects into two baskets. Ours(8B) scores 10.0 and Ours(27B) scores 9.5.

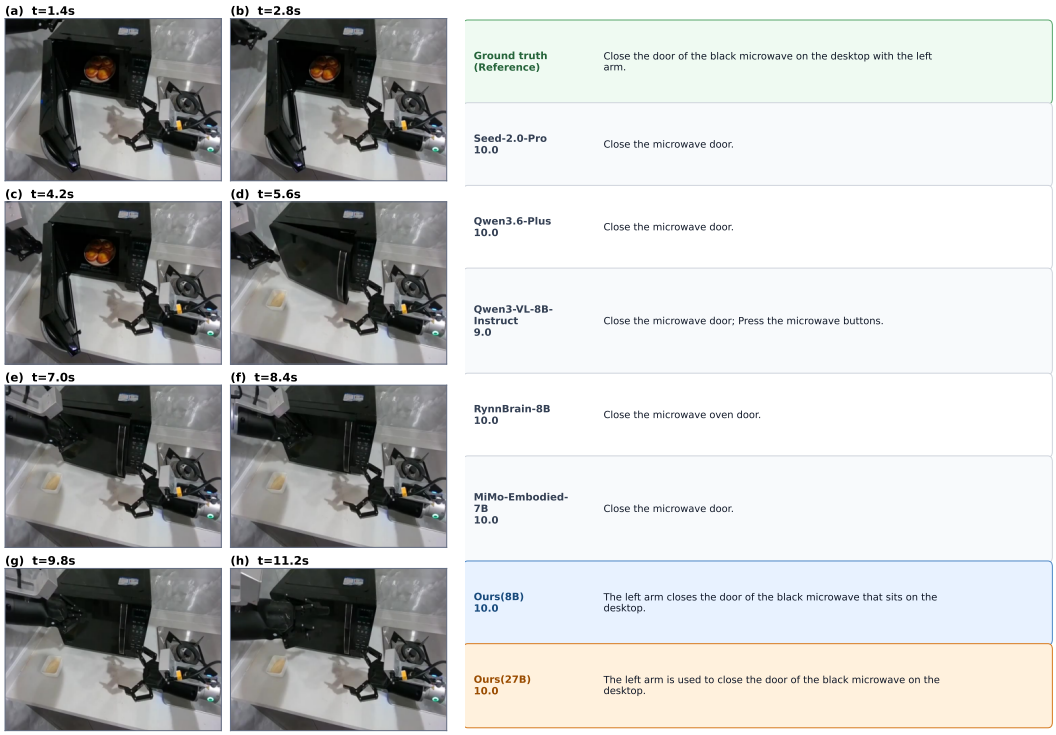


Figure 17: RoboActionTrace Atomic100 example RAT 001: close the microwave door with the left arm. Ours(8B) and Ours(27B) both receive 10.0.

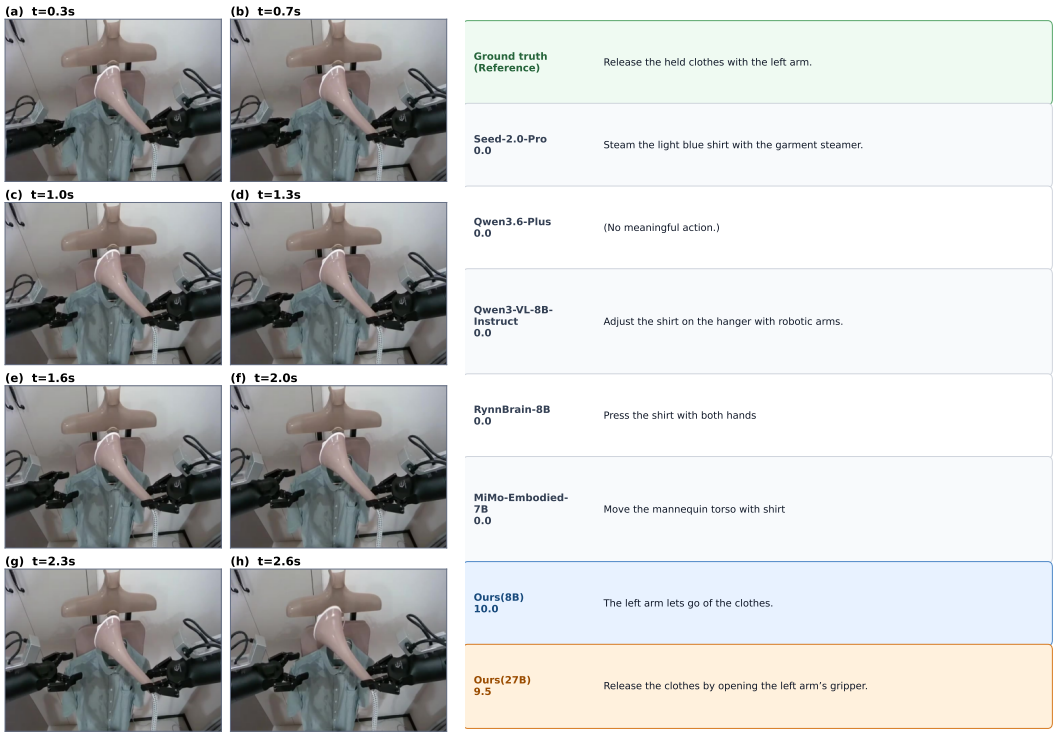


Figure 18: RoboActionTrace Atomic100 example RAT 004: release the held clothes with the left arm. Ours(8B) scores 10.0 and Ours(27B) scores 9.5.

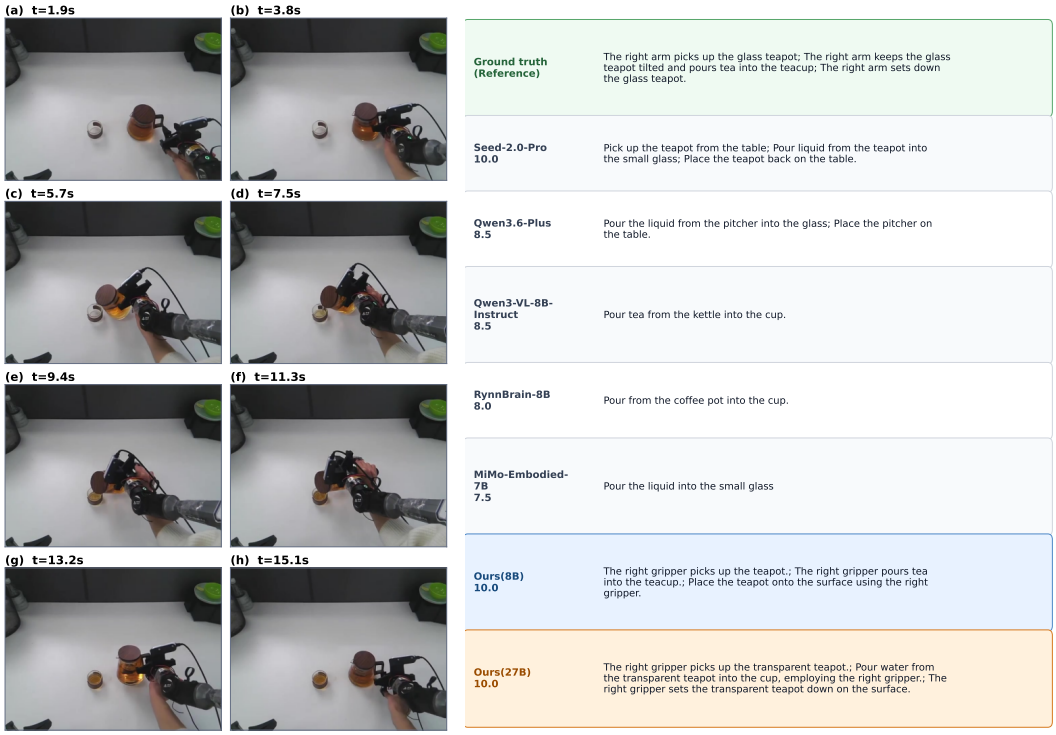


Figure 19: RoboActionTrace Concat100 example RAT 023: pick up a teapot, pour tea, and set the teapot down. Ours(8B) and Ours(27B) both receive 10.0.

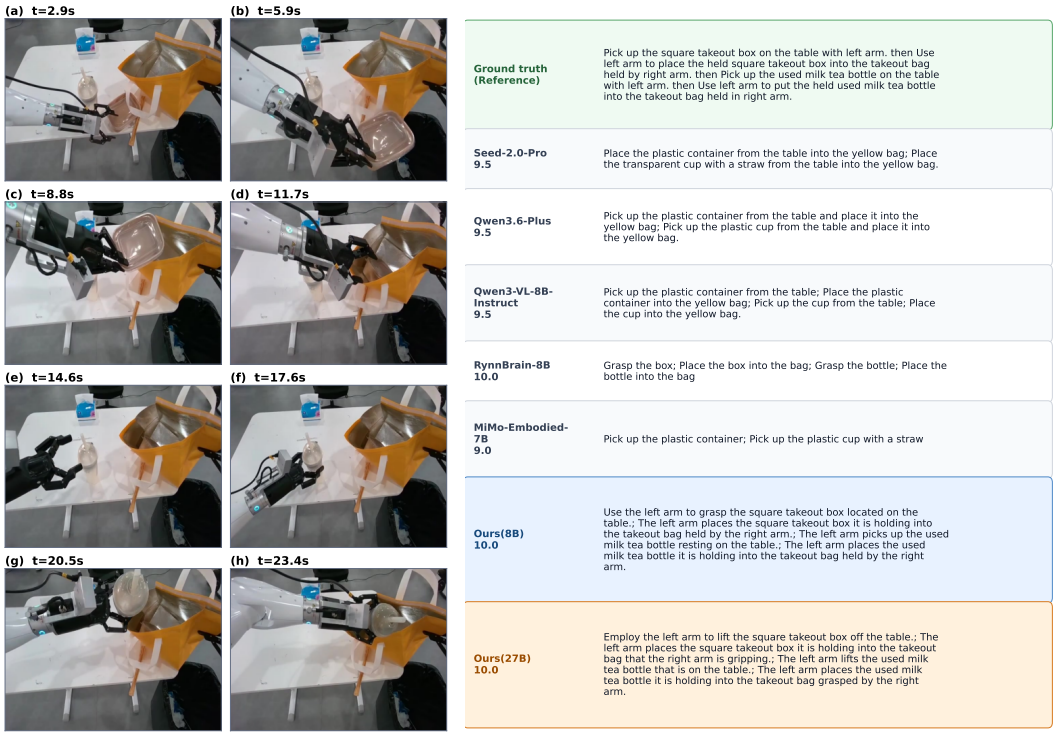


Figure 20: RoboActionTrace Concat100 example RAT 002: place a takeout box and a bottle into a bag. Ours(8B) and Ours(27B) both receive 10.0.