Generation-for-Understanding with Structured Action Scripts for Embodied Multimodal Learning

Hange Zhou* **Mingzhe Zhang*** **Hongyu Ge** **Xiaotang Wang**
Enjun Du **Runqing Xu** **Yongqi Zhang†**

The Hong Kong University of Science and Technology (Guangzhou)
{hzhou489, yongqizhang}@connect.hkust-gz.edu.cn

*Equal contribution †Corresponding author

Abstract

Vision-language models (VLMs) are increasingly used as perception and reasoning modules for embodied intelligence, yet they remain brittle when understanding fine-grained physical processes in videos. While existing models often recognize the overall task, they frequently fail to recover the intermediate actions and result states that determine how an embodied interaction actually unfolds. We argue that this limitation stems in part from a supervision bottleneck: conventional video-language data is typically constructed through post-hoc annotation, where captions are written after video collection and often provide only coarse, loosely aligned descriptions of dynamic physical events. We propose **Gen4Understand**, a generation-for-understanding framework that reverses this data flow. Instead of captioning videos after they are observed, Gen4Understand first represents embodied interactions as structured action scripts composed of atomic events and state deltas, then synthesizes videos conditioned on these scripts and real start-end frame anchors. A verifier-calibrator scores each generated candidate using its conditioning script, endpoint anchors, and video content, filtering for semantic consistency, endpoint fidelity, physical plausibility, temporal coherence, and visual quality. The retained atomic clips are further merged into action-, subtask-, and task-level supervision and converted into captioning and question-answering data for VLM adaptation. Experiments across public embodied video splits, distractor-rich hallucination tests, simulated VLA benchmarks, and real-robot tasks show that Gen4Understand consistently improves fine-grained action understanding, result-state recognition, temporal ordering, hallucination resistance, and downstream embodied control. These results suggest that controllable video generation can serve not merely as data augmentation, but as a mechanism for creating calibrated process-level supervision for embodied intelligence. Our code is available at https://github.com/HangerAmber/Gen4Understand.

1 Introduction

Vision-language models (VLMs) have become an important foundation for embodied intelligence, enabling high-level planning, multimodal reasoning, and language-guided interaction in physical environments [1, 2]. Despite this progress, current VLMs still exhibit a critical weakness in embodied video understanding: they can often infer *what* task is being performed, but fail to recover *how* the task physically unfolds. In manipulation scenarios, this process-level understanding requires tracking intermediate actions, object state transitions, temporal order, contact changes, and causal outcomes.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

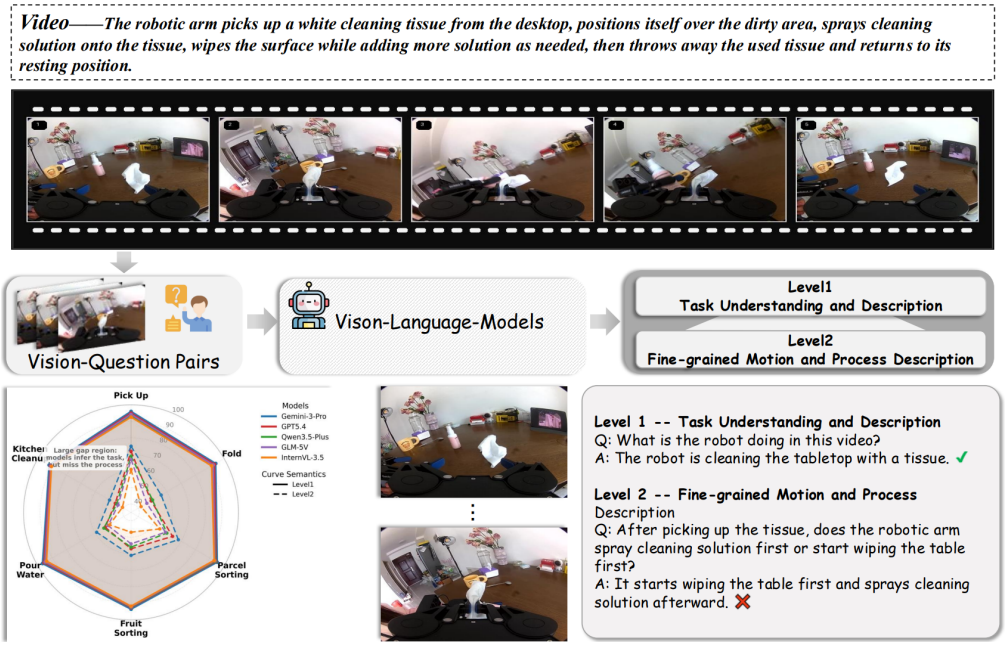

Figure 1: **Motivation: current VLMs often understand the final task but fail to capture the underlying motion process.** Existing VLMs can usually recognize the overall task in an embodied video, but often fail on fine-grained questions about intermediate actions, temporal order, and state transitions. The radar chart shows a large and consistent gap between task-level understanding and motion-process understanding across diverse human/robot manipulation scenarios, motivating our focus on process-level embodied supervision.

These details are essential for embodied agents, where the difference between "placing a brush into a basket" and "taking a brush out to clean a cup" can determine the next action to execute.

A recurring failure mode is that VLMs rely on sparse keyframes and semantic priors rather than the observed motion process. For example, when one gripper holds a cup containing water while another gripper returns a brush to a basket, a model may incorrectly infer that the brush is being used to clean the cup, or that it is about to be taken out for use. Such predictions are plausible under object co-occurrence priors, but unsupported by the actual action sequence. As illustrated in Figure 1, strong VLMs can answer coarse task-level questions with high accuracy, yet remain substantially weaker on fine-grained questions about intermediate action order and state changes.

We argue that this failure is not just a model-side weakness; it is a supervision bottleneck rooted in how embodied video-language data is constructed. Most video-language datasets follow a *post-hoc video-to-caption* pipeline: videos are collected first, and captions are written afterward [3–6]. Post-hoc annotation can provide detailed process supervision, but obtaining consistent event boundaries, state transitions, and their alignment with visual evidence remains demanding. This motivates exploring complementary ways to construct process-level video-language pairs. First, fine-grained post-hoc annotation is expensive and inconsistent: annotators must specify who acts, on which object, through what motion, in what order, and with what resulting state change [7–11]. Second, even detailed captions do not guarantee tight temporal alignment with the visual evidence [12–14]. A caption may mention several actions in a long video, but the model still has to infer which words correspond to which short-horizon motion and state transition. As a result, VLMs can learn to predict plausible task summaries from context without developing robust sensitivity to the physical process itself.

This raises a key question: instead of making post-hoc captions denser, can we create data where the action process is specified before the video exists? We answer this question by proposing **Gen4Understand**, a *generation-for-understanding* framework that reverses the conventional data flow. Our central idea is to first write a structured action script describing the desired physical process, and then synthesize embodied videos conditioned on this script and real start-end frame anchors. In this way, the intended actor, action, object, motion, temporal relation, and state transition are specified at the source of data creation, rather than inferred afterward from an already collected video.

The first component of Gen4Understand is a *structured action-script* representation. Rather than using free-form text, we represent an embodied video as a hierarchy of atomic events explicitly encoding the actor, action, object, state deltas, motion, and temporal relations. This hierarchy provides precise local supervision while maintaining compatibility with realistic long-horizon inputs. The second component is *grounded script-to-video generation*. We condition a generation model on the structured script alongside real start and end frame anchors, which better constrains semantic content and physical state transitions compared to text-only methods. Because generated videos are not automatically reliable, the third component is a *verifier-calibrator* that filters candidates according to semantic consistency, endpoint fidelity, physical plausibility, temporal coherence, and visual quality. The retained high-confidence clips are then hierarchically merged into multi-granularity captioning and QA supervision for VLM adaptation.

We evaluate the framework along three axes: fine-grained embodied video understanding with structural metrics, hallucination resistance on distractor-rich videos, and downstream transfer into a Vision-Language-Action (VLA) system for real-robot manipulation. Experiments show that our calibrated script-generated data improves fine-grained motion understanding, reduces prior-driven hallucination, and yields more reliable execution in tasks requiring progress verification and state-aware decision making.

In summary, our contributions are three-fold:

- We introduce Gen4Understand, an offline framework that constructs process-level video-language supervision from structured action scripts, real endpoint anchors, and verified generated clips.
- We show that the advantage of calibrated generated data persists under matched structured supervision and adaptation budgets. Controlled comparisons distinguish this additional benefit from improvements associated with scripts, QA, and hard negatives.
- We characterize the resulting supervision through visual-evidence probes, independent human fidelity audits, and trajectory-derived alternatives, and evaluate its transfer to simulated and real-robot control.

2 Related Work

2.1 Embodied VLMs and Fine-grained Physical Understanding

VLMs have become an important component of embodied intelligence, enabling language-conditioned perception, planning, affordance reasoning, and robotic control in physical environments. Representative systems such as PaLM-E [1], RT-2 [2], and VoxPoser [15] show that large multimodal models can connect visual observations with high-level action decisions, while recent embodied agents extend these capabilities to sequential decision making and long-horizon interaction [16].

However, embodied reasoning requires more than recognizing the overall task or likely intent. In manipulation videos, models must recover intermediate actions, contact changes, temporal order, and object state transitions. Existing VLMs often remain unreliable under this fine-grained setting, especially in distractor-rich scenes where object co-occurrence and semantic priors can override the observed motion evidence. Recent model-side efforts, including hierarchical video perception and improved spatiotemporal fusion [17, 18], aim to strengthen dynamic visual modeling. Our work is complementary: instead of only modifying architectures, we construct supervision that explicitly specifies atomic embodied events and their resulting state changes.

2.2 Video-Language Supervision for Motion and Embodiment

Large-scale video-language datasets such as MSR-VTT [10], WebVid [19], and InternVid [11] have driven progress in general video understanding, but they mainly emphasize global semantics and web-scale diversity rather than local physical interaction. Egocentric resources such as Ego4D [8] and Ego-Exo4D [7] provide richer action context, yet their annotations are not designed to consistently encode object-centric state transitions, contact relations, or manipulation structure.

Recent benchmarks and annotation frameworks have highlighted this limitation. MotionBench [12] evaluates fine-grained motion reasoning, Caption Anything in Video [14] studies object-centric dense video captioning, and FoundationMotion [13] explores scalable auto-labeling for spatial movement and motion reasoning. These works show the importance of detailed motion supervision, but most

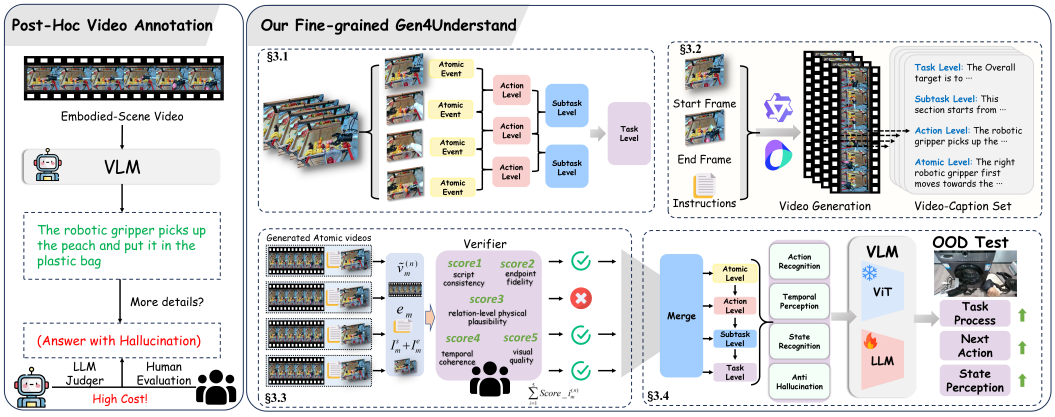

Figure 2: **Overview of Gen4Understand. Left:** Conventional post-hoc annotation captions collected embodied videos, producing coarse descriptions that can miss intermediate actions and state changes. **Right:** Gen4Understand represents processes as structured events (Section 3.1), generates candidate video-script pairs conditioned on scripts and real endpoints (Section 3.2), selects clips using a weighted, multi-criterion verifier (Section 3.3), and constructs hierarchical supervision for student adaptation (Section 3.4). Generation and verification are performed offline.

still follow a post-hoc video-to-text paradigm: videos are collected first, and motion descriptions or questions are produced afterward. Such supervision remains loosely aligned with the short-horizon visual evidence that determines physical interaction. In contrast, we formulate embodied video learning as structured event-and-state supervision, specifying *who* acts, *on what*, *through which motion*, *in what order*, and *with what resulting state change* before constructing the training video.

2.3 Video Generation as a Data Engine

Recent video generation models, including Sora [20], Seedance [21], and Wan [22], have demonstrated strong text- and image-conditioned video synthesis capabilities. Generated experience has also been used to improve embodied policies and video captioning, as demonstrated by UniSim [23]. Our focus is the construction of explicit process-level video-language supervision: structured scripts specify the intended events and state changes, real endpoints constrain their visual realization, and candidate selection measures consistency before student adaptation. Matched supervision controls evaluate the additional value of this construction pipeline.

This use case imposes stricter requirements than visual realism alone. Generated clips must preserve action semantics, respect start and end states, maintain temporal coherence, and avoid physically implausible transitions. We therefore combine script-conditioned video synthesis with a verifier-calibrator that filters candidates by semantic consistency, endpoint fidelity, physical plausibility, temporal coherence, and visual quality. Thus, generation is not merely used to scale the data, but to create calibrated, process-level supervision that is difficult to obtain from post-hoc annotations alone.

3 Method

We introduce **Gen4Understand**, a generation-for-understanding framework that converts structured action semantics into calibrated embodied video-language supervision. The key idea is to replace post-hoc video captioning with a *script-to-video* pipeline: we first specify the desired physical process as a structured action script, synthesize videos conditioned on this script and real endpoint anchors, filter unreliable generations with a verifier-calibrator, and finally transfer the retained data to VLMs through multi-granularity captioning and QA supervision.

3.1 Structured Action Scripts

Given an embodied video $V = \{I_t\}_{t=1}^{T}$, our goal is to represent the underlying physical process rather than only the final task intent. We describe the process as a sequence of atomic events

$$\mathcal{E} = \{e_m\}_{m=1}^{M}, \qquad e_m = (g_m, u_m, o_m, s_m^-, \mu_m, s_m^+, \rho_m), \tag{1}$$

where g_m is the acting agent, u_m the action verb, o_m the manipulated object, s_m^- and s_m^+ the pre- and post-action states, μ_m a motion descriptor, and ρ_m the temporal relation to neighboring events. Unlike free-form captions, this explicit slot-level structure precisely tracks intermediate state transitions, providing a shared interface for video synthesis, filtering, and evaluation.

Atomic events are hierarchically merged into action-, subtask-, and task-level units. For a unit j at level $l \in \{atom, act, sub, task\}$ covering event indices Ω_j^l, its script is $\mathcal{E}_j^l = Merge(\{e_m : m \in \Omega_j^l\})$. This multi-level structure directly bridges dense local supervision with realistic long-horizon inference contexts.

Each script unit is rendered into a positive caption c_j^{pos} using a deterministic template bank. To improve hallucination resistance, we also render plausible but unsupported actions as hard-negative constraints c_j^{neg}. These negatives are not sampled from the full action vocabulary; instead, they are selected from actions semantically tempting in the current scene but absent from the event sequence, such as `wipe` or `clean` in a brush–cup scene without wiping. The final script caption $c_j = \text{Concat}(c_j^{pos}, c_j^{neg})$ concatenates the positive process description with these negative constraints.

3.2 Endpoint-Grounded Script-to-Video Synthesis

For each atomic event e_m, we extract real start/end frames I_m^s, I_m^e from its source video and render its script $c_m \equiv c_m^{\text{atom}}$. We generate N candidate clips:

$$\tilde{v}_{m,n} = \mathcal{G}_\psi(I_m^s, I_m^e, c_m, \varepsilon_{m,n}), \qquad n = 1, \ldots, N, \tag{2}$$

Here $\mathcal{G}_\psi$ uses Wan2.7 and Seedance 2.0, each supplying two candidates with independent seeds $\varepsilon_{m,n}$ ($N = 4$). Generation and verification are offline; implementation settings are in Appendix E.2.

Endpoint grounding provides a direct constraint on the desired state transition. The start frame anchors the pre-action scene configuration, the end frame anchors the result state, and the structured script specifies the motion that should connect them. Compared with generation from a single initial frame, this design reduces semantic drift and discourages the generator from producing merely plausible but incorrect motions.

3.3 Verifier-Calibrator

Generated candidates are not automatically reliable. A clip may preserve the endpoints but fail to realize the intended action, introduce implausible contact, or contain temporal artifacts. We therefore score each candidate before using it for training. For $z_{m,n} = (\tilde{v}_{m,n}, e_m, I_m^s, I_m^e)$, the verifier-calibrator assigns

$$S_{m,n} = \sum_{r \in \{\text{sem,anc,phys,tmp,qual}\}} w_r S_r(z_{m,n}), \qquad w_r \geq 0, \ \sum_r w_r = 1. \tag{3}$$

The semantic term uses a frozen Qwen2.5-VL-72B event parser. Anchor fidelity combines DINOv2 endpoint features with matched object masks; relational checks use GroundingDINO/SAM2 tracks; temporal coherence combines RAFT flow-warp consistency with trajectory smoothness; and visual quality uses DOVER. The five terms are min–max normalized on 1,000 candidates from 250 training-only events, with normalization parameters frozen for subsequent candidates. The selected weights are $(w_{\text{sem}}, w_{\text{anc}}, w_{\text{phys}}, w_{\text{tmp}}, w_{\text{qual}}) = (0.35, 0.20, 0.20, 0.15, 0.10)$. Weights and the acceptance threshold are calibrated to maximize audited precision subject to event coverage of at least 35%. These scores are filtering proxies for visual and relational consistency, rather than certificates of physical correctness.

For each event, we retain the highest-scoring candidate only if its score exceeds a threshold:

$$n_m^\star = \arg\max_n S_{m,n}, \qquad \tilde{v}_m^\star = \tilde{v}_{m,n_m^\star}, \qquad \mathcal{D}_{\text{atom}} = \{(\tilde{v}_m^\star, e_m, c_m) : S_{m,n_m^\star} \geq \gamma\}. \tag{4}$$

We retain at most one candidate per event: the highest-scoring candidate when its score reaches $\gamma = 0.70$. Approximately 38% of events are retained. In a calibration-disjoint, blinded audit by three annotators, 183/200 accepted clips (91.5%) satisfy the intended event and endpoint state, compared with 24/200 rejected clips (12.0%). Agreement before majority voting is Fleiss' $\kappa = 0.82$. The audit supports the effectiveness of selection while quantifying an 8.5% residual error rate among accepted clips.

Table 1: **Data summary.** We construct structured evaluation splits from public embodied video sources and augment them with event-state annotations required for process-level evaluation. Generated training pairs are produced by Gen4Understand after verifier-calibrator filtering.

Source / Split	Public Source Scale	Structured Split	Event Labels	Primary Use
Ego-Exo4D-Manip	5,035 takes / 1,286h	1,200 clips	4,800	human-object manipulation
EPIC-KITCHENS-Process	700 videos / 89.9K actions	1,000 clips	3,900	egocentric process understanding
DROID-Robot	76K trajectories / 350h	800 clips	3,200	robot manipulation understanding
Public Distractor Split	public videos	600 clips	2,100	hallucination evaluation
Generated Atomic Pairs	generated	20K accepted clips	20K scripts	fine-grained VLM adaptation
Hierarchical Pairs	generated/merged	8K merged clips	multi-level scripts	action/subtask/task adaptation
Real Robot Trials	physical robot	3 families $\times$ 120 trials per model variant	–	downstream validation

3.4 Hierarchical Data Transfer

The validated atomic clips provide precise local supervision, but downstream VLMs are often used on longer videos. We therefore transfer calibrated data at multiple granularities.

For a higher-level unit j at level l, we collect its accepted child events $\Omega^l_{j,acc} = \{m \in \Omega^l_j : S_{m,n^*_m} \geq \gamma\}$. We construct a merged training sample only when the accepted children provide sufficient coverage; in our main experiments, we require all child atomic clips to be accepted (i.e., $\Omega^l_{j,acc} = \Omega^l_j$). The merged video V_j is obtained by concatenating accepted child clips in temporal order while removing duplicated boundary frames. Its language supervision is rendered from the merged script over the same accepted event sequence.

Each video-script pair is then converted into instruction-following examples using a QA template family. For a unit j, we generate tuples (V_j, q, r), where q is an instruction or question and r is the corresponding response. The templates target complementary aspects of embodied understanding, including action recognition, temporal order, state-transition prediction, and distractor-aware negative reasoning. This trains the model not only to describe the video, but also to answer what changed, what did not happen, and which event occurred before or after another.

The target VLM is optimized over all granularity levels:

$$\mathcal{L}(\theta) = - \sum_{\ell \in \{\text{atom,act,sub,task}\}} \alpha_\ell \sum_{(V,q,r) \in \mathcal{D}^\ell_{\text{qa}}} \log p_\theta(r \mid \text{Enc}_{\text{vis}}(V), q)\,, \tag{5}$$

where α_ℓ controls the contribution of each level. Atomic examples teach local manipulation mechanics and state transitions, while action-, subtask-, and task-level examples preserve compatibility with long-horizon embodied inputs. This multi-granularity transfer bridges the gap between fine-grained supervision and realistic VLM deployment.

4 Experiments

We evaluate Gen4Understand through four research questions. **RQ1**: Does script-generated data improve fine-grained embodied video understanding? **RQ2**: Does it reduce hallucination induced by strong object priors? **RQ3**: Do the improved video understanding capabilities transfer to downstream embodied control? **RQ4**: Which design choices are responsible for the gains?

4.1 Experimental Setup

We evaluate human-verified event-state annotations on held-out real videos from Ego-Exo4D, EPIC-KITCHENS-100, and DROID (Table 1). Source recordings are partitioned before script extraction, endpoint selection, or generation; no evaluation recording supplies training scripts, anchors, or candidates. Split construction, annotation agreement, and adaptation settings are detailed in Appendices E.1, E.3, and E.2.

For hallucination evaluation, a 600-clip public-video distractor split pairs tempting object contexts with absent stereotypical actions, annotated as unsupported actions and intents (Appendix E.4).

Downstream control follows the native LIBERO [24] and RoboTwin 2.0 [25] protocols, plus three real-robot task families targeting result states, temporal order, and distractor resistance. The action policy and controller remain fixed; only the perception-language module changes (Appendix E.2).

Table 2: **RQ1: Fine-grained embodied video understanding on public-source diagnostic splits.** FG-Score is the average Slot-F1, State-change Accuracy, Temporal-order Accuracy, and Contact/Containment Accuracy.

Method	FG-Score ↑			Overall Avg
	Ego-Exo4D-Manip	EPIC-KITCHENS-Process	DROID-Robot	
Qwen2.5-VL-3B-Instruct	47.2	45.6	43.8	45.5
+ Real + fine-grained post-hoc captions	51.0 ↑3.8	49.1 ↑3.5	47.1 ↑3.3	49.1 ↑3.6
+ Gen4Understand	55.6 ↑8.4	53.1 ↑7.5	51.0 ↑7.2	53.2 ↑7.7
Qwen2.5-VL-7B-Instruct	52.1	50.3	50.6	51.0
+ Real + fine-grained post-hoc captions	55.5 ↑3.4	53.6 ↑3.3	51.8 ↑1.2	53.6 ↑2.6
+ Gen4Understand	61.9 ↑9.8	59.6 ↑9.3	58.1 ↑7.5	59.9 ↑8.9
Qwen3-VL-4B-Instruct	53.4	51.6	49.9	51.6
+ Real + fine-grained post-hoc captions	56.6 ↑3.2	54.7 ↑3.1	53.0 ↑3.1	54.8 ↑3.2
+ Gen4Understand	60.4 ↑7.0	58.5 ↑6.9	57.1 ↑7.2	58.7 ↑7.1
RoboBrain2.5-4B	56.2	54.0	53.2	54.5
+ Real + fine-grained post-hoc captions	59.1 ↑2.9	56.9 ↑2.9	56.0 ↑2.8	57.3 ↑2.8
+ Gen4Understand	**62.6** ↑6.4	**60.5** ↑6.5	**60.1** ↑6.9	**61.1** ↑6.6

We evaluate Qwen2.5-VL-3B-Instruct, Qwen2.5-VL-7B-Instruct [26], Qwen3-VL-4B-Instruct [27], and RoboBrain2.5-4B [28]. We also include a *Real + fine-grained post-hoc captions* baseline using disjoint real clips annotated after collection with the same event-state schema. We report Slot-F1, State-change Accuracy, Temporal-order Accuracy, Contact/Containment Accuracy, and their average FG-Score for RQ1; False-action Rate (FAR), False-intent Rate (FIR), and Distractor Robustness (DR) for RQ2; and task success, stage completion, state-transition success, and hallucination-induced error for RQ3. Formal metric definitions are provided in Appendix B.

4.2 RQ1: Fine-grained Embodied Video Understanding

We first evaluate whether Gen4Understand improves process-level video understanding. The model answers structured questions derived from event-state annotations, including actor-action-object recognition, result-state prediction, temporal ordering, and contact/containment reasoning. This setting directly tests whether the model understands what physically changes during the interaction, rather than only recognizing the final task.

Table 2 shows improvements across the evaluated backbones and public-source splits, covering human manipulation, egocentric kitchen activities, and robot interactions. To determine whether the advantage persists after matching the structure of supervision, we additionally evaluate controlled real-video baselines.

Matched supervision controls. Table 3 compares Qwen2.5-VL-7B on the same 3,000 held-out real evaluation clips under matched adaptation budgets and training-instance counts. Full Gen4Understand achieves 59.90 FG, compared with 57.20 for the real-video control with matched scripts, QA, and hard negatives. The paired improvement is 2.70 points (95% CI [1.78, 3.66]). This comparison supports the benefit of the complete calibrated generated-data pipeline beyond the evaluated structured real-video control. It does not attribute the entire difference to video synthesis alone. Bootstrap details are given in Appendix F.6.

Trajectory-derived supervision. In a separate DROID-only adaptation setting, Gen4Understand achieves 58.9 FG versus 54.0 for trajectory/state supervision and 55.9 for trajectory plus post-hoc captions under matched budgets. Baseline construction, component results, and paired confidence intervals are provided in Appendix G.2.

Reliance on visual process evidence. Temporal-order accuracy drops from 60.40 to 36.80 when interior frames are shuffled while preserving the endpoints and frame count; endpoint-only inputs achieve 43.60. These results support reliance on intermediate process evidence. Template-generalization and conflicting-script probes are detailed in Appendix G.3.

Table 3: Matched-supervision controls on Qwen2.5-VL-7B. All conditions share 3,000 held-out real evaluation clips. The upper panel reports FG-Score and marginal 95% bootstrap intervals; the lower panel reports paired differences relative to real videos with matched scripts, QA, and hard negatives.

Training supervision	FG ↑	95% CI
Real + post-hoc captions	53.60	[51.8, 55.4]
Real + matched scripts/QA	56.90	[55.1, 58.6]
Real + matched scripts/QA/hard negatives	57.20	[55.4, 58.9]
Calibrated script-to-video	58.70	[57.0, 60.4]
Full Gen4Understand	59.90	[58.2, 61.5]
Paired comparison	**ΔFG**	**95% CI**
Calibrated − strongest matched-real	+1.50	[0.62, 2.41]
Full − strongest matched-real	+2.70	[1.78, 3.66]

Table 4: **RQ2: Hallucination resistance on the public-video distractor split.** Strong closed-source and open-source VLMs both exhibit prior-driven hallucinations. Real post-hoc captions reduce unsupported predictions, while Gen4Understand achieves the strongest hallucination resistance.

Type	Method	DR ↑	FAR ↓	FIR ↓
Closed-source	GPT-4o	76.5	24.2	23.0
	Gemini-2.5-Flash	74.8	26.5	23.9
	GLM-5V	75.6	25.1	23.7
Open-source	Qwen2.5-VL-3B-Instruct	67.6	34.2	30.6
	Qwen2.5-VL-7B-Instruct	71.1	30.5	27.3
	Qwen3-VL-4B-Instruct	72.5	28.6	26.4
	RoboBrain2.5-4B	73.8	27.9	24.5
+ Real post-hoc captions	Qwen2.5-VL-3B-Instruct	73.8 ↑6.2	28.2 ↓6.0	24.2 ↓6.4
	Qwen2.5-VL-7B-Instruct	76.0 ↑4.9	25.1 ↓5.4	22.9 ↓4.4
	Qwen3-VL-4B-Instruct	77.1 ↑4.6	24.1 ↓4.5	21.7 ↓4.7
	RoboBrain2.5-4B	77.8 ↑4.0	23.2 ↓4.7	21.2 ↓3.3
+ Gen4Understand	Qwen2.5-VL-3B-Instruct	81.8 ↑14.2	19.6 ↓14.6	16.8 ↓13.8
	Qwen2.5-VL-7B-Instruct	84.0 ↑12.9	17.5 ↓13.1	14.6 ↓12.7
	Qwen3-VL-4B-Instruct	85.1 ↑12.6	16.5 ↓12.1	13.3 ↓13.1
	RoboBrain2.5-4B	**85.9** ↑12.1	**15.7** ↓12.2	**12.5** ↓12.0

4.3 RQ2: Resistance to Prior-driven Hallucination

We next evaluate whether Gen4Understand reduces hallucination caused by semantic shortcuts. This experiment is not an official task from the source datasets; it is a diagnostic split constructed from public videos to measure a failure mode that existing benchmarks do not isolate. Closed-source models are evaluated zero-shot using the same prompts and parsing protocol; open-source backbones are compared under the base model, Real post-hoc captions, and Gen4Understand.

Table 4 shows that even strong closed-source models exhibit prior-driven hallucinations on this split. Fine-grained post-hoc captions reduce unsupported predictions, but Gen4Understand further lowers FAR and FIR and in several cases surpasses closed-source baselines on DR. These results show that targeted process-level adaptation can substantially reduce unsupported action and intent predictions on this diagnostic split. The zero-shot closed-source models provide reference performance, rather than a controlled comparison of model scale.

4.4 RQ3: Transfer to Downstream Embodied Control

We then evaluate whether improved fine-grained understanding transfers to embodied action. We first test in simulation using native benchmark success rates, then validate representative task families on real robots. To isolate perception-language grounding, we keep the action policy, low-level controller, task instructions, and rollout budget fixed across variants.

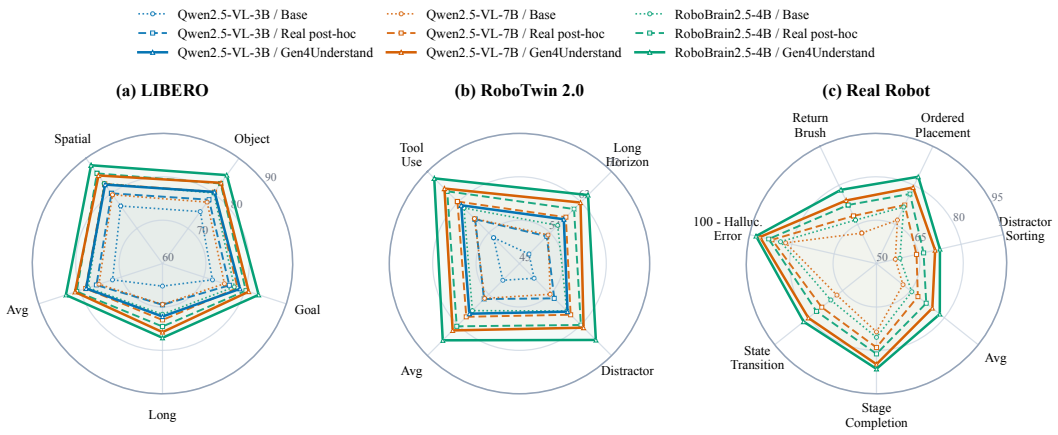

Figure 3: **RQ3: Radar summary of VLA transfer.** Panels (a) and (b) show LIBERO and RoboTwin 2.0 success rates, respectively. Panel (c) shows real-robot transfer across task success, stage completion, state-transition success, and hallucination resistance; for consistent orientation, Hallucination-induced Error is plotted as $100-$Halluc. Error, so larger values are better.

Figure 3(a,b) shows that both adaptations improve VLA success in simulation, with Gen4Understand giving larger gains in RoboTwin 2.0 long-horizon and distractor settings where execution relies more heavily on progress recognition and state-aware interpretation.

Figure 3(c) summarizes real-robot transfer, with detailed numerical results in Table 9. Across both evaluated backbones, Gen4Understand achieves higher task-success point estimates than real post-hoc adaptation on all three task families. Averaged across the three families, Gen4Understand improves task success over real post-hoc adaptation by 6.4 percentage points for Qwen2.5-VL-7B and 6.1 points for RoboBrain2.5-4B.

Each method is evaluated in three configurations with 40 trials each, totaling 120 trials per task family. For Qwen2.5-VL-7B, Gen4Understand succeeds in 89/120 Return Brush, 95/120 Ordered Placement, and 85/120 Distractor-aware Sorting trials. The corresponding success rates are 74.2%, 79.2%, and 70.8%, with 95% Wilson intervals of [65.7, 81.2], [71.1, 85.5], and [62.2, 78.2] percent, respectively. These results support transfer to physical execution in tasks requiring result-state recognition, temporal ordering, and distractor resistance. The intervals quantify uncertainty under the evaluated configuration mixture and do not by themselves establish the significance of pairwise method differences.

4.5 RQ4: Ablation, Scaling, and Failure Analysis

Finally, we analyze which components are responsible for the improvements. Table 7 (Appendix F) summarizes the main ablations on Qwen2.5-VL-7B-Instruct.

The matched controls indicate that structured supervision accounts for part of the improvement, while the full calibrated generated-data pipeline provides an additional benefit under the evaluated controls. These results support the combined pipeline rather than attributing the full gain to video generation alone. The ablations further support the contributions of endpoint grounding, candidate filtering, hierarchical transfer, QA supervision, and distractor-oriented samples under the reported adaptation budgets.

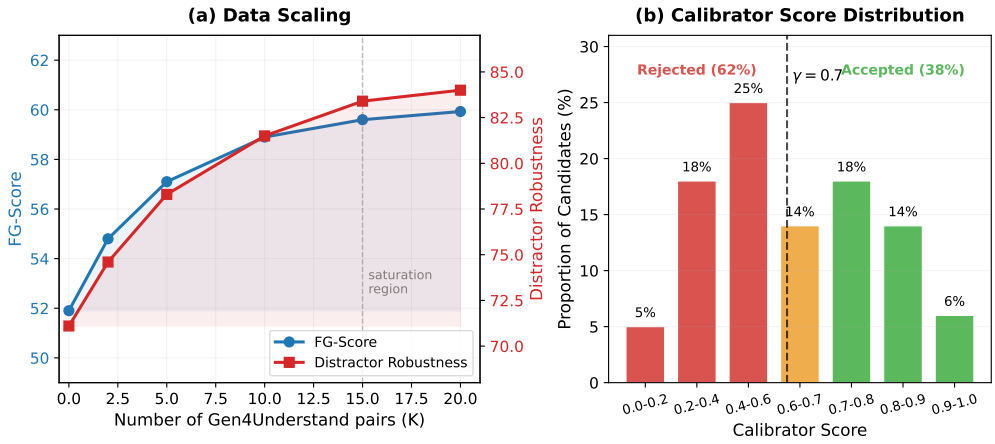

Figure 4: **Scaling and calibration analysis. Left**: FG-Score and Distractor Robustness as a function of the number of Gen4Understand training pairs. **Right**: verifier-calibrator score distribution over generated candidates.

Figure 4 analyzes scaling and calibration quality. Performance improves steadily up to ∼15K pairs and then begins to saturate. At $\gamma = 0.70$, approximately 38% of eligible events retain a clip, selected as the highest-scoring of four candidates. Calibrated generated data outperforms unfiltered generated data under the reported matched adaptation budget. In a source-balanced human audit, 274/300 accepted clips (91.3%) satisfy all four script-fidelity criteria. Appendix D.5 reports the audit protocol and the domain- and category-dependent variation in retention and fidelity.

Figure 5 decomposes the remaining errors on the distractor split. After adaptation, hallucinated-action and hallucinated-intent errors decrease substantially, while remaining errors are increasingly dominated by perceptual ambiguity, such as occlusion, subtle contact changes, and visually similar objects. Additional qualitative examples are provided in Appendix H.

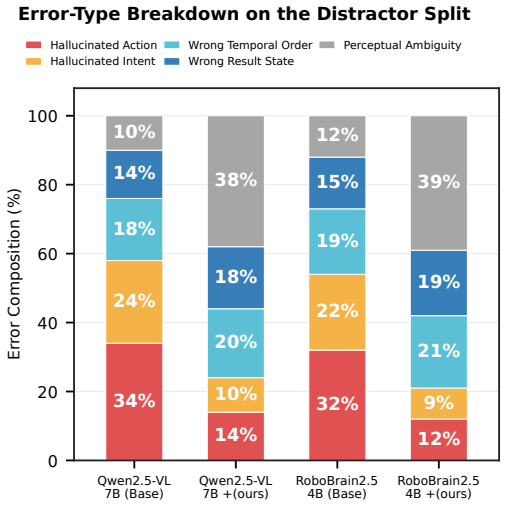

Figure 5: **Error-type breakdown on the distractor split.** Gen4Understand reduces prior-driven hallucination errors, while residual errors are increasingly caused by perceptual ambiguity.

5 Conclusion

We introduced Gen4Understand, a generation-for-understanding framework that addresses the process-level supervision bottleneck in embodied video learning. Instead of relying solely on post-hoc captions, Gen4Understand represents embodied interactions as structured atomic events and state deltas, synthesizes endpoint-grounded clips from these scripts, filters candidates with a verifier-calibrator, and transfers the retained data into hierarchical captioning and QA supervision. This calibrated supervision teaches VLMs not only what task is likely being performed, but also what intermediate actions occur, what states change, and what plausible actions do not happen. Experiments show consistent gains in fine-grained action understanding, result-state and temporal reasoning, hallucination resistance, and downstream VLA control. More broadly, our results suggest that controllable video generation is not merely a data-scaling tool, but a mechanism for supplying structured physical supervision to embodied intelligence systems, motivating future work on state-aware video generation and embodied control.

Acknowledgments and Disclosure of Funding

Funding. This work was supported by the Guangdong Basic and Applied Basic Research Foundation (Project No. 2025A1515010304), the Guangdong Provincial Project (Project No. 2024QN11X088), and the Guangzhou Science and Technology Planning Project (Project No. 2025A03J4491). This work was also partially supported by the HKUST(GZ) Kunpeng&Ascend Center of Cultivation, which provided access to Ascend computing resources used for Wan-based video generation.

Competing Interests. The authors declare no competing interests.

References

[1] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[2] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[3] Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, et al. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

[4] Yifei Wang, Zhenkai Li, Tianwen Qian, Huanran Zheng, Zheng Wang, Yuqian Fu, and Xiaoling Wang. Streameqa: Towards streaming video understanding for embodied scenarios. *arXiv preprint arXiv:2512.04451*, 2025.

[5] Bowen Yang, Zishuo Li, Yang Sun, Changtao Miao, Yifan Yang, Man Luo, Xiaotong Yan, Feng Jiang, Jinchuan Shi, Yankai Fu, et al. Aoe: Always-on egocentric human video collection for embodied ai. *arXiv preprint arXiv:2602.23893*, 2026.

[6] Shuyi Zhang, Xiaoshuai Hao, Yingbo Tang, Lingfeng Zhang, Pengwei Wang, Zhongyuan Wang, Hongxuan Ma, and Shanghang Zhang. Video-cot: A comprehensive dataset for spatiotemporal understanding of videos based on chain-of-thought. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 12745–12752, 2025.

[7] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024.

[8] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022.

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022.

[10] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[11] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023.

[12] Wenyi Hong, Yean Cheng, Zhuoyi Yang, Weihan Wang, Lefan Wang, Xiaotao Gu, Shiyu Huang, Yuxiao Dong, and Jie Tang. Motionbench: Benchmarking and improving fine-grained video motion understanding for vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8450–8460, 2025.

[13] Yulu Gan, Ligeng Zhu, Dandan Shan, Baifeng Shi, Hongxu Yin, Boris Ivanovic, Song Han, Trevor Darrell, Jitendra Malik, Marco Pavone, et al. Foundationmotion: Auto-labeling and reasoning about spatial movement in videos. *arXiv preprint arXiv:2512.10927*, 2025.

[14] Yolo Yunlong Tang, Jing Bi, Chao Huang, Susan Liang, Daiki Shimada, Hang Hua, Yunzhong Xiao, Yizhi Song, Pinxin Liu, Mingqian Feng, et al. Caption anything in video: Fine-grained object-centric captioning

via spatiotemporal multimodal prompting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 41697–41699, 2026.

[15] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

[16] Xinxin Zhao, Wenzhe Cai, Likun Tang, and Teng Wang. Imaginenav: Prompting vision-language models as embodied navigator through scene imagination. *arXiv preprint arXiv:2410.09874*, 2024.

[17] Tanveer Hannan, Md Mohaiminul Islam, Jindong Gu, Thomas Seidl, and Gedas Bertasius. Revisionllm: Recursive vision-language model for temporal grounding in hour-long videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19012–19022, 2025.

[18] Jinhui Yi, Syed Talal Wasim, Yanan Luo, Muzammal Naseer, and Juergen Gall. Video-panda: Parameter-efficient alignment for encoder-free video-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24119–24128, 2025.

[19] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021.

[20] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.

[21] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

[22] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[23] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1(2):6, 2023.

[24] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[25] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[26] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

[27] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025.

[28] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1724–1734, 2025.

[29] Dayong Liu, Chao Xu, Weihong Chen, Suyu Zhang, Juncheng Wang, Jiankang Deng, Baigui Sun, and Yang Liu. Beyond description: Cognitively benchmarking fine-grained action for embodied agents. *arXiv preprint arXiv:2511.18685*, 2025.

[30] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[31] Yun Li, Yiming Zhang, Tao Lin, XiangRui Liu, Wenxiao Cai, Zheng Liu, and Bo Zhao. Sti-bench: Are mllms ready for precise spatial-temporal world understanding? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5622–5632, 2025.

[32] Ziang Zhang, Zehan Wang, Guanghao Zhang, Weilong Dai, Yan Xia, Ziang Yan, Minjie Hong, and Zhou Zhao. Dsi-bench: A benchmark for dynamic spatial intelligence. *arXiv preprint arXiv:2510.18873*, 2025.

[33] Songhao Han, Wei Huang, Hairong Shi, Le Zhuo, Xiu Su, Shifeng Zhang, Xu Zhou, Xiaojuan Qi, Yue Liao, and Si Liu. Videoespresso: A large-scale chain-of-thought dataset for fine-grained video reasoning via core frame selection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26181–26191, 2025.

[34] Zeqian Li, Qirui Chen, Tengda Han, Ya Zhang, Yanfeng Wang, and Weidi Xie. Multi-sentence grounding for long-term instructional video. In *European Conference on Computer Vision*, pages 200–216. Springer, 2024.

[35] Kaiting Liu and Hazel Doughty. Let's split up: Zero-shot classifier edits for fine-grained video understanding. *arXiv preprint arXiv:2602.16545*, 2026.

[36] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

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

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

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

[40] Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

[41] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[42] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[43] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 9156–9172. Curran Associates, Inc., 2023.

[44] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[45] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *arXiv preprint arXiv:2311.01455*, 2023.

[46] Hai Zhang, Siqi Liang, Li Chen, Yuxian Li, Yukuan Xu, Yichao Zhong, Fu Zhang, and Hongyang Li. Sparse video generation propels real-world beyond-the-view vision-language navigation. *arXiv preprint arXiv:2602.05827*, 2026.

[47] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[48] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts, 2023.

[49] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[50] 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.

[51] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[52] Enjun Du, Siyi Liu, Zirong Chen, Xinyu Zuo, Jinwen Luo, Ruiwen Tao, Lisheng Duan, Haijin Liang, Jin Ma, Junfu Pu, et al. Evirank: Structured relevance evidence for multimodal image re-ranking. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, pages 7640–7666, 2026.

[53] Siyi Liu, Xiaorong Zhu, Enjun Du, Xinyu Zuo, Lisheng Duan, Haijin Liang, Jin Ma, Junfu Pu, and Yongqi Zhang. Repair: Turning retrieval failures into counterfactual hard pairs. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, pages 43774–43800, 2026.

[54] Haodong Wu, Jiahao Zhang, Lijie Hu, and Yongqi Zhang. From instance selection to fixed-pool data recipe search for supervised fine-tuning. *arXiv preprint arXiv:2605.12944*, 2026.

[55] Bairui Zhang, Weixuan Liu, Defan Xue, and Yongqi Zhang. Tokencliff: Benchmarking output-token budget sensitivity in multi-turn llm agents. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, pages 9415–9427, 2026.

A Appendix Overview

This appendix provides additional details omitted from the main paper due to space constraints. Appendix B gives formal definitions of the evaluation metrics. Appendix C describes the structured action-script vocabulary, language rendering, and hard-negative construction. Appendix D provides implementation details for the verifier-calibrator. Appendix E describes dataset construction and evaluation protocols. Appendix F reports ablation studies. Appendix J discusses limitations and failure cases.

Broader research context. General-purpose vision-language and reasoning models complement benchmarks and supervision methods for fine-grained action, spatiotemporal understanding, and instructional-video grounding [29–39]. Related directions connect robotic datasets, affordance prediction, multimodal prompting, and generalist policies with video generation, interactive environments, and predictive video representations [40–51]. Broader methodological context includes structured multimodal relevance evidence, counterfactual hard pairs, fixed-pool instruction-data recipes, and output-token-budget sensitivity [52–55].

B Evaluation Metrics

B.1 Structural Correctness Metrics

To evaluate fine-grained embodied video understanding, we measure whether a model correctly recovers the structured event sequence

$$\mathcal{E} = \{e_m\}_{m=1}^{M}, \qquad e_m = (g_m, u_m, o_m, s_m^-, \mu_m, s_m^+, \rho_m),$$

where the slots denote actor, action, manipulated object, source state, motion descriptor, result state, and temporal relation.

Slot-level Accuracy and F1. For each predicted event $\hat{e}_m$, we compare its fields against the ground-truth event e_m. We report per-slot accuracy and F1 over actor, action, object, result state, and temporal relation. Overall Slot-F1 is computed by aggregating predictions across all events and slots.

Exact Event Match. An event is counted as correct only if all key slots are correct:

$$\mathrm{EM}(e_m, \hat{e}_m) = \mathbb{I}[g_m = \hat{g}_m \wedge u_m = \hat{u}_m \wedge o_m = \hat{o}_m \wedge s_m^+ = \hat{s}_m^+ \wedge \rho_m = \hat{\rho}_m].$$

We report exact match at both event level and full-sequence level.

Result-State Accuracy. Because the resulting object state is critical for embodied reasoning, we separately evaluate post-action state prediction:

$$\mathrm{Acc}_{\mathrm{state}} = \frac{1}{M} \sum_{m=1}^{M} \mathbb{I}[s_m^+ = \hat{s}_m^+].$$

This metric measures whether the model understands what physically changed after each manipulation.

Temporal Order Accuracy. For temporal reasoning, we compare the relative order of event pairs. Let $\pi(i, j) = 1$ if event e_i occurs before e_j, and 0 otherwise. We compute

$$\mathrm{Acc}_{\mathrm{temp}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \mathbb{I}[\pi(i, j) = \hat{\pi}(i, j)],$$

where $\mathcal{P}$ is the set of valid event pairs. We additionally report Kendall's τ when the event sequence length is sufficient.

Contact and Containment Accuracy. We evaluate relation-level physical understanding by checking whether the model correctly predicts relations such as `held-by`, `touching`, `in`, `on`, and `inside`. These relations are evaluated at the source and result states when available.

FG-Score. For summary reporting, we define

$$\mathrm{FG} = \frac{1}{4}\left(\mathrm{SlotF1} + \mathrm{Acc}_{\mathrm{state}} + \mathrm{Acc}_{\mathrm{temp}} + \mathrm{Acc}_{\mathrm{rel}}\right),$$

where $\mathrm{Acc}_{\mathrm{rel}}$ denotes Contact/Containment Accuracy.

B.2 Hallucination-Resistance Metrics

We evaluate hallucination resistance on distractor videos containing strong prior-inducing object combinations, where the actual motion does not realize the stereotypical action associated with those objects.

False-Action Rate. Let $\mathcal{A}^-(V)$ denote the set of unsupported actions for a distractor video V, such as `wipe`, `clean`, `brush`, or `take-out`. Given the predicted structured caption $\hat{\mathcal{E}}$, we define

$$\mathrm{FAR} = \frac{1}{|\mathcal{D}_{\mathrm{dis}}|} \sum_{V \in \mathcal{D}_{\mathrm{dis}}} \mathbb{I}\left[\exists \hat{u} \in \hat{\mathcal{E}} \text{ such that } \hat{u} \in \mathcal{A}^-(V)\right].$$

False-Intent Rate. Let $\mathcal{G}^-(V)$ denote unsupported intents for V, such as "prepare to clean the cup." We define

$$\mathrm{FIR} = \frac{1}{|\mathcal{D}_{\mathrm{dis}}|} \sum_{V \in \mathcal{D}_{\mathrm{dis}}} \mathbb{I}\left[\exists \hat{g} \in \hat{\mathcal{G}}(V) \text{ such that } \hat{g} \in \mathcal{G}^-(V)\right].$$

Intents are extracted either from structured QA responses or from free-form outputs using a fixed intent taxonomy.

Distractor Robustness. We combine the two hallucination metrics as

$$\mathrm{DR} = 1 - \frac{\mathrm{FAR} + \mathrm{FIR}}{2}.$$

B.3 Downstream Transfer Metrics

For VLA transfer, we report four metrics.

Task Success Rate. A trial is successful if the final task goal is completed according to the task specification.

Stage Completion Rate. Each long-horizon task is decomposed into intermediate stages. Stage Completion measures the fraction of required stages completed in the correct order.

State-Transition Success. This metric measures whether the intended object-state change occurs, such as an object ending `inside` a container, a tool being returned to a target region, or a support/contact relation being correctly established.

Hallucination-Induced Error Rate. We count an error as hallucination-induced if the robot executes an action caused by unsupported perception or prior-driven interpretation, such as wiping when no wiping is required, taking an object out when it should be returned, or interacting with a distractor object.

C Structured Action-Script Details

C.1 Slot Vocabulary

Our structured action script represents each atomic event as

$$e_m = (g_m, u_m, o_m, s_m^-, \mu_m, s_m^+, \rho_m).$$

The slot vocabulary is intentionally compact to improve annotation consistency and automatic evaluation.

Actor. Actors include human hands and robotic effectors, e.g., `left hand`, `right hand`, `left gripper`, and `right gripper`.

Action. The action vocabulary includes atomic manipulation verbs such as `reach`, `grasp`, `lift`, `move`, `place`, `insert`, `release`, `wipe`, `open`, `close`, `pour`, and `push`. We use a compact vocabulary to reduce ambiguity between semantically similar verbs.

Object. The object slot specifies the manipulated entity, such as `brush`, `cup`, `basket`, `block`, `towel`, or `container`.

Source and Result States. States encode relative location, contact relation, containment, support, and visible attributes. Examples include `held-by(left gripper)`, `on(table)`, `inside(basket)`, `near(cup)`, and `open(drawer)`.

Motion Descriptor. The motion descriptor captures directional or trajectory-level information, e.g., `left-to-right`, `upward`, `toward basket`, `away from cup`, or `downward into container`.

Temporal Relation. Temporal relations include `before`, `after`, and `while`. These relations are used for sequential and concurrent manipulation events.

C.2 Natural-Language Rendering

Each structured event or merged event group is rendered into natural language using a deterministic template bank. For an atomic event, a typical template is:

Actor g_m performs action u_m on object o_m, moving it according to μ_m, resulting in state s_m^+.

For higher-level units, we merge consecutive events into a temporally coherent description. For example, an event sequence can be rendered as:

> The right gripper moves the brush toward the basket and releases it, resulting in the brush being inside the basket.

C.3 Hard-Negative Construction

To improve hallucination resistance, we augment positive captions with hard-negative constraints. Instead of treating all absent actions as negatives, we select plausible but unsupported actions based on scene context. For example, if a video contains a brush, cup, and water but the event sequence only shows returning the brush to a basket, `wipe`, `clean`, and `brush` are selected as hard negatives.

The negative statement is rendered as:

> No wiping or cleaning action occurs in this segment; the brush is returned to the basket.

This design discourages the model from using object co-occurrence priors to infer unsupported actions.

C.4 QA Template Examples

We convert structured scripts into QA-style supervision. The templates cover four categories:

Action Recognition.

> Q: What action does the right gripper perform on the brush?
> A: It moves the brush toward the basket and releases it.

State-Transition Prediction.

> Q: Where is the brush after the action?
> A: The brush is inside the basket.

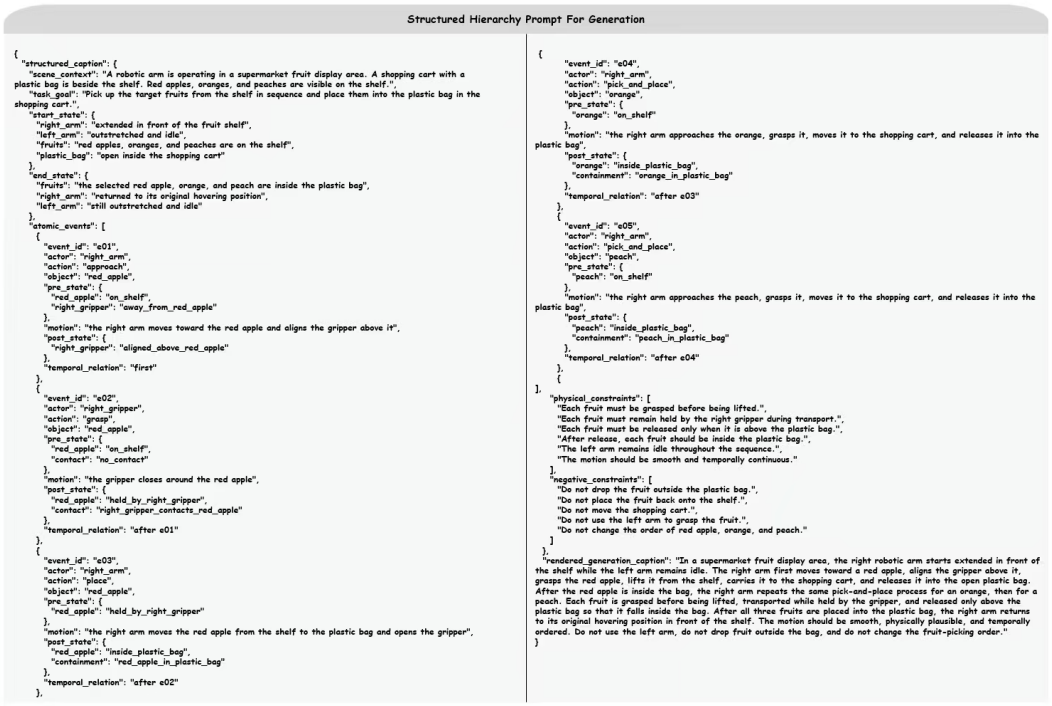

Figure 6: **Structured hierarchy prompt for video generation.** This example demonstrates how an embodied task is decomposed into explicit atomic events and constraints, which are then rendered into a dense natural language caption to guide video synthesis.

Temporal Order.

Q: Does the robot spray cleaning solution before or after wiping begins?
A: It sprays cleaning solution before wiping begins.

Negative/Distractor Reasoning.

Q: Does the robot use the brush to clean the cup?
A: No. The brush is returned to the basket, and no cleaning action occurs.

C.5 Complete Structured Prompt Example for Video Generation

Figure 6 presents a complete structured prompt, demonstrating how atomic events, state transitions, and constraints are integrated into a final caption to condition video generation.

D Verifier-Calibrator Details

The verifier-calibrator assigns a confidence score to each generated candidate clip:

$$S_m^{(n)} = \lambda_{\text{sem}} S_{\text{sem}} + \lambda_{\text{anc}} S_{\text{anc}} + \lambda_{\text{phys}} S_{\text{phys}} + \lambda_{\text{tmp}} S_{\text{tmp}} + \lambda_{\text{qual}} S_{\text{qual}}.$$

The weights are non-negative and normalized to sum to one.

D.1 Semantic Consistency

We instantiate the event parser Φ with frozen Qwen2.5-VL-72B and use greedy decoding to extract the realized event slots from each generated clip. Semantic consistency is the slot-level F1 agreement between these predictions and the conditioning event. The parser is used for candidate scoring; its generated language is not used as student training supervision.

$$S_{\text{sem}} = \text{SlotF1}(\Phi(\tilde{v}_m^{(n)}), e_m).$$

This term checks whether the generated clip realizes the intended actor, action, object, temporal relation, and result state.

D.2 Anchor Fidelity

Anchor fidelity measures whether the generated video preserves the real boundary frames:

$$S_{\text{anc}} = \frac{1}{2}\left[\text{sim}(\text{First}(\tilde{v}_m^{(n)}), I_m^s) + \text{sim}(\text{Last}(\tilde{v}_m^{(n)}), I_m^e)\right].$$

Anchor fidelity uses DINOv2 ViT-L/14 endpoint cosine similarity and mask IoU from Hungarian-matched GroundingDINO/SAM2 object tracks. Detection and tracking use GroundingDINO Swin-T OGC and SAM2 Hiera-Large. A candidate is rejected if target tracks are absent from more than 20% of its interior frames. For each event, we generate four candidates and retain the highest-scoring candidate that passes this track-presence check only if its overall verifier score satisfies $S_m^{(n)} \geq 0.70$; otherwise, the event is discarded.

D.3 Physical Plausibility

Let $\mathcal{R}(\cdot)$ denote predicted physical relations, including contact, support, containment, and relative location. We compute

$$S_{\text{phys}} = \text{RelAcc}\left(\mathcal{R}(\tilde{v}_m^{(n)}), \mathcal{R}(e_m)\right).$$

Pre- and post-states are evaluated over the first and last eight frames at 16 fps. A relation must hold in at least six of eight frames. With distances normalized by the frame diagonal, contact requires mask-boundary distance ≤ 0.02. Support requires vertical gap ≤ 0.03 and horizontal overlap, divided by the smaller object width, ≥ 0.30. Containment requires the object centroid to lie inside the container mask and a contained fraction ≥ 0.70. Release requires pre-window contact, post-window non-contact, and a separation increase ≥ 0.03. Directional relations use a signed centroid margin ≥ 0.05 of the relevant axis. The score is the fraction of applicable scripted checks satisfied. These checks assess projected visual relations, rather than complete three-dimensional physical dynamics.

D.4 Temporal Coherence and Visual Quality

Temporal coherence combines RAFT-Large photometric flow-warp consistency with target-trajectory smoothness:

$$S_{\text{tmp}} = 0.7S_{\text{flow}} + 0.3S_{\text{smooth}},$$

where

$$S_{\text{flow}} = \exp(-E_{\text{warp}}/0.05), \qquad S_{\text{smooth}} = \exp\left(-\frac{\text{mean}_t \, \|p_{t+1} - 2p_t + p_{t-1}\|_2}{0.015}\right).$$

Here E_{warp} is the mean RGB photometric warp error with RGB values in $[0, 1]$, and p_t is the normalized target centroid. Technical visual quality is obtained from DOVER using the official `DOVER.pth` checkpoint.

S_{qual} provides a technical-quality signal for candidate selection. Lower technical quality reduces the aggregate verifier score. Apart from the track-presence rejection rule, acceptance is determined by the aggregate threshold; no additional per-component thresholds are used.

D.5 Threshold Selection and Human Audits

Calibration uses 1,000 candidates from 250 training-only events. Each component is min–max normalized on this set; the resulting extrema are frozen, and subsequent scores are clipped to $[0, 1]$. We search component weights over $\{0.10, 0.15, \ldots, 0.40\}$, constrained to sum to one, and thresholds over $\{0.60, 0.65, 0.70, 0.75, 0.80\}$. Selection maximizes audited precision subject to event coverage of at least 35%.

The selected weights are $(0.35, 0.20, 0.20, 0.15, 0.10)$ for semantic, anchor, physical, temporal, and quality scores, respectively. An event is retained only when its highest-scoring candidate reaches $\gamma = 0.70$, yielding approximately 38% event-level retention. At most one of the four candidates is retained for each event. Appendix F studies the effect of removing the calibrator and ablating its components.

Stratified fidelity audit. We conduct a blinded human audit of 300 accepted clips, with 100 clips from each source domain. Three annotators assess atomic-event realization, temporal order, state delta, and motion descriptor, with each criterion aggregated by majority vote. Full script adherence requires satisfying all four criteria.

Full script adherence is observed in 94/100 clips from DROID, 91/100 from Ego-Exo4D, and 89/100 from EPIC-KITCHENS. Across the 300 clips, 290/300 realize the atomic event, 285/300 preserve temporal order, 280/300 realize the state delta, and 278/300 satisfy the motion descriptor. Overall, 274/300 clips (91.3%) satisfy all four criteria. This percentage describes the source-balanced audit sample, rather than a distribution-weighted estimate for the full retained dataset. We report this audit separately from the accepted/rejected audit and do not pool their sample counts.

Variation across domains and action categories. Event-level retention is the fraction of eligible events whose highest-scoring candidate among four reaches $\gamma = 0.7$. Retention is 44% for DROID, 37% for Ego-Exo4D, and 33% for EPIC-KITCHENS.

Across action categories, event-level retention decreases from 46% for rigid pick-and-place to 24% for cutting/deformable actions, while audited validity among accepted clips decreases from 95% to 84%. Harder categories therefore exhibit both lower coverage and lower retained fidelity. Filtering mitigates, but does not eliminate, the dependence on generation quality.

D.6 Cost of data construction.

Let M be the number of atomic events, N the number of generated candidates per event, and L the number of frames per generated clip. The dominant cost is candidate generation and verification, which scales as $O(MN(C_{\text{gen}}(L) + C_{\text{ver}}(L)))$. Hierarchical merging is linear in the number of retained clips and is negligible compared with generation. With streaming verification, only accepted clips need to be stored, yielding storage proportional to the acceptance rate. This matches the empirical scaling behavior analyzed in Section 4.5.

E Dataset and Evaluation Protocol Details

E.1 Public Fine-grained Evaluation Splits

We evaluate fine-grained video understanding on public-source benchmarks that emphasize physical interaction and state transitions. Ego-Exo4D provides skilled human activities and object-centric manipulation, offering rich context for fine-grained action reasoning. EPIC-KITCHENS-100 offers large-scale egocentric videos of complex kitchen activities, allowing us to evaluate process-level understanding and intermediate state changes. DROID contributes a diverse collection of real-world robot manipulation trajectories, which we use to assess the model's ability to recover precise physical interactions in robotic scenarios. Since none of these datasets provides all state-transition annotations required by our structured evaluation, we derive structured targets from released annotations and public metadata, then manually verify the held-out evaluation subset.

Split construction. Source recording groups are partitioned before script extraction, endpoint selection, or video generation. No evaluation recording contributes a training script, anchor, or generated candidate. Scene and object instances are separated across partitions, while categories and verbs may overlap. Event-tuple deduplication and cross-partition endpoint screening further reduce overlap.

E.2 Training and Evaluation Protocol

VLM adaptation. We adapt each backbone VLM on the Gen4Understand dataset using supervised fine-tuning (SFT). Following the architecture illustrated in Figure 2, we keep the visual encoder (ViT) *frozen* and fine-tune only the language model (LLM) component and the vision-language projector. This design preserves the pretrained visual representations while allowing the language decoder to internalize process-level supervision from our generated data.

For all Qwen-family models (Qwen2.5-VL-3B-Instruct, Qwen2.5-VL-7B-Instruct, Qwen3-VL-4B-Instruct), we use LoRA applied to all linear layers of the LLM with rank r=64 and scaling factor

α=128. For RoboBrain2.5-4B, we also use LoRA with the same rank and scaling configuration. Table 5 summarizes the training hyperparameters.

Table 5: **Training hyperparameters for VLM adaptation.**

Hyperparameter	Value
Visual encoder	Frozen
LLM tuning method	LoRA (all-linear)
LoRA rank / alpha	64 / 128
Optimizer	AdamW
Learning rate	1e-4
LR schedule	Cosine decay with linear warmup
Warmup ratio	0.03
Total epochs	3
Per-device batch size	4
Gradient accumulation steps	4
Effective batch size	16 (per node)
Max sequence length	4096 tokens
Video frame sampling	16 frames per clip (uniform)
Max video resolution	Native resolution, max 1280 px
Precision	BF16 mixed precision
Weight decay	0.01
Gradient clipping	1.0
Multi-granularity loss weights (Eq. 5)	
α_{atom}	1.0
α_{act}	0.5
α_{sub}	0.3
α_{task}	0.2

Training data composition. As summarized in Table 1 (main text), the adaptation dataset consists of 20K accepted atomic video-script pairs and 8K hierarchically merged clips. Each video-script pair is converted into QA-format instruction tuples via our template family (Section 3.4). This yields approximately 56K training instances in total (each atomic clip produces $\sim$2 QA pairs covering action recognition and state-transition questions; each merged clip produces $\sim$1.5 QA pairs covering temporal ordering, task-level captioning, and distractor-aware negative reasoning). During training, atomic and hierarchical instances are sampled into mixed batches according to the loss weights above.

Generation settings. Each event receives four independently seeded candidates, two each from Wan2.7 and Seedance 2.0 in first-and-last-frame conditioning mode. Candidates are five-second, 720p videos, resampled to 16 fps for verification; student adaptation uses 16 uniformly sampled frames per clip. Generation and verification are offline data-construction stages and are not required at student inference time.

Real-video baselines and matched controls. The real-video baselines use human event-state annotations on evaluation-disjoint real clips. Within each backbone, compared adaptation conditions share the adaptation hyperparameters and total training-instance count.

Table 3 distinguishes post-hoc caption supervision from controls with matched scripts and QA, and with additional hard negatives. The controls use a shared deterministic renderer for the supervision types enabled in each condition. The strongest matched-real control also matches the generated training set's event-category distribution. We use this control to assess the additional benefit of the calibrated generated-data pipeline.

VLA downstream evaluation. When evaluating transfer to downstream embodied control (RQ3), we integrate the adapted VLM as the perception-language module of a VLA system. The low-level action policy, controller parameters, task instructions, and rollout budget are kept identical across all VLM variants. Specifically:

Annotation target	κ	Exact / boundary (%)
Event boundary	–	86.8
Actor slot	0.91	94.1
Action slot	0.84	89.3
Object slot	0.88	92.0
Result state	0.81	86.7
Temporal relation	0.86	90.4
Contact/containment	0.78	84.1
Unsupported action	0.83	88.5
Unsupported intent	0.80	86.9
Slot average	0.84	89.0

Table 6: **Annotation agreement for structured evaluation labels.** Agreement is measured before third-annotator adjudication. Exact agreement is reported in percent. For event boundaries, we report the percentage of boundaries matched within 0.5 seconds, so Cohen's κ is not applicable.

- **LIBERO**: The benchmark contains 130 tasks overall, including LIBERO-100. We report success rates for the Spatial, Object, Goal, and Long suites, averaged over 20 rollouts per task.
- **RoboTwin 2.0**: We evaluate 50 dual-arm manipulation tasks using the benchmark environments and task-success criteria, with 20 rollouts per task. We report results for the Tool-use, Long-horizon, and Distractor task groups used in our evaluation.
- **Real robot**: Each model variant, defined by a backbone and its adaptation condition, is evaluated on three task families: result-state-sensitive, temporal-order-sensitive, and distractor-rich tasks. For each task family, we evaluate three configurations with 40 trials each, totaling 120 trials per family and 360 trials per model variant. Configurations vary object identity and pose, initial gripper state, receptacle position, distractors, and action order. A trial is successful only if the final symbolic state and all task constraints are satisfied within the time limit. Two observers verify synchronized recordings and resolve disagreements by reviewing those recordings. For each model variant and task family, task success rate is computed as the total number of successful trials across the three configurations divided by 120. We quantify uncertainty in these proportions using two-sided 95% Wilson intervals with $n = 120$. Stage completion measures the fraction of required stages completed in the correct order. State-transition success assesses whether the intended object-state change occurs. Hallucination-induced errors are identified when an unsupported action or intent prediction in the VLM's intermediate perception output leads to an incorrect downstream action.

Closed-source model evaluation. Closed-source models (GPT-4o, Gemini-2.5-Flash, GLM-5V) in Table 4 are evaluated zero-shot using the same structured QA prompts and output parsing protocol. No fine-tuning is applied to these models; they serve as reference points for the hallucination resistance that strong commercial VLMs achieve out of the box.

Evaluation consistency. All evaluations use greedy decoding (temperature $= 0$) to ensure reproducibility. For structured QA evaluation (RQ1, RQ2), model outputs are parsed into slot-level predictions using deterministic regex-based extraction aligned with our event schema. Parsing failures are counted as incorrect predictions.

E.3 Annotation Protocol and Agreement

Each evaluation clip is independently annotated by two trained annotators using the event-state schema in Appendix C. Annotators first mark atomic event boundaries and then fill the actor, action, object, result-state, temporal-relation, and contact/containment fields. For the distractor split, they additionally mark unsupported actions and intents. Table 6 reports pre-adjudication agreement; all disagreements are resolved by a third annotator before constructing the final evaluation labels.

The κ column measures chance-corrected agreement on categorical slots, while the last column reports direct exact-match agreement for slots and boundary-match accuracy for event segmentation. Agreement is highest for visually explicit fields such as actor and object, and lower for result state and contact/containment because these labels depend on subtle physical relations and short-horizon state changes. The slot-average agreement indicates that the structured labels are reliable before

Variant	FG-Score ↑	State-change Acc. ↑	Distractor Robustness ↑	Real-robot Avg (%) ↑
Full Gen4Understand	**59.93**	**59.10**	**84.00**	**74.73**
Start-frame only	57.20	55.80	81.60	71.20
w/o Verifier-Calibrator	55.20	52.90	79.40	69.10
Semantic-only Calibrator	56.40	54.10	80.30	70.00
Semantic + Anchor	57.50	55.60	81.20	71.10
Semantic + Anchor + Physical	58.80	57.90	82.90	72.80
Atomic-only Training	58.10	58.20	82.70	70.90
Caption-only Transfer	57.00	55.40	81.90	70.80
w/o Distractor-oriented Samples	59.20	58.70	77.80	72.60

Table 7: **Ablation studies.** Each component contributes to the final performance. Endpoint grounding improves result-state recognition, the verifier-calibrator improves data precision, hierarchical transfer improves long-horizon and VLA performance, QA-oriented transfer improves process-level reasoning, and distractor-oriented samples are essential for hallucination resistance.

adjudication, while the third-annotator pass removes the remaining disagreements used in final evaluation.

E.4 Distractor Split Construction

The distractor split is constructed from public-source videos containing strong prior-inducing object combinations. Examples include `brush-cup-water`, `towel-table`, and `toothbrush-toothpaste`. A sample is included only when the stereotypical action suggested by object co-occurrence does not occur in the video. Each distractor sample is annotated with unsupported actions $\mathcal{A}^{-}(V)$ and unsupported intents $\mathcal{G}^{-}(V)$ for hallucination evaluation.

E.5 VLA Evaluation Tasks

We evaluate downstream transfer on three real-robot task families.

Result-State-Sensitive Tasks. These tasks require the robot to achieve a precise final object relation, such as placing a brush inside a basket rather than beside it.

Temporal-Order-Sensitive Tasks. These tasks require completing subtasks in a specified order, such as opening a container before placing an object inside.

Distractor-Rich Tasks. These tasks include semantically tempting objects that should not be acted upon. For example, a brush and cup may appear in the scene although the intended task is to return the brush rather than clean the cup.

F Ablation Studies

We ablate the main components of Gen4Understand. Unless otherwise specified, all variants use the same base VLM and matched adaptation budgets. We report representative metrics covering fine-grained understanding, hallucination resistance, and downstream transfer.

F.1 Start-frame Only vs. Start-End Anchors

The Start-frame only variant conditions generation on the first frame and structured script, but removes the end-frame anchor. Performance drops most strongly on State-change Accuracy, confirming that the end frame provides important supervision for the intended result state.

F.2 Effect of the Verifier-Calibrator

Removing the verifier-calibrator substantially degrades all metrics. This shows that raw generated clips contain enough semantic drift and physical inconsistency to harm downstream training.

Component-level ablations further show that semantic consistency alone is insufficient; anchor fidelity and physical plausibility are both important for high-quality synthetic supervision.

F.3 Atomic-only vs. Hierarchical Transfer

Atomic-only training improves short-horizon understanding but underperforms on long-horizon videos and VLA transfer. This supports our design choice of merging validated atomic clips into action-, subtask-, and task-level supervision.

F.4 Caption-only vs. QA-oriented Transfer

Caption-only transfer removes structured QA templates and trains only on caption generation. This variant performs worse on state-change and temporal reasoning, suggesting that QA supervision forces the model to attend explicitly to process-level details.

F.5 Effect of Distractor-oriented Samples

Removing distractor-oriented samples leads to a large drop in Distractor Robustness while only mildly affecting FG-Score. This confirms that hallucination resistance is driven by targeted supervision over semantically tempting but unsupported actions.

F.6 Matched Real-Video Controls

Table 3 evaluates whether the benefit of generated supervision persists after matching the structure of real-video supervision. All conditions use Qwen2.5-VL-7B, the same held-out evaluation clips, and matched adaptation settings and training-instance counts.

Adding matched scripts and QA increases FG-Score from 53.60 to 56.90, and adding hard negatives increases it to 57.20. Calibrated script-to-video supervision reaches 58.70, exceeding the strongest matched-real control by 1.50 points (paired 95% CI [0.62, 2.41]). Full Gen4Understand reaches 59.90, with a paired difference of 2.70 points [1.78, 3.66] relative to the same control.

Bootstrap confidence intervals. These intervals use 10,000 source-split-stratified clip-bootstrap replicates, pairing the same evaluation clips across conditions.

These comparisons support an additional benefit from the calibrated generated-data pipeline after matching structured supervision. They do not attribute the entire improvement over the post-hoc baseline to video generation alone.

G Additional Results

G.1 Detailed Numerical Results for VLA Experiments

In this section, we provide the detailed numerical results(Table 8,Table 9) corresponding to the radar charts in Figure 3. These metrics offer a precise quantification of the performance gains achieved by **Gen4Understand** across simulation benchmarks and real-world robotic tasks.

Uncertainty in downstream control. For simulation, Avg is the unweighted macro-average over the reported benchmark suites or task categories. We use 10,000 category-stratified hierarchical bootstrap replicates, resampling tasks within each category and binary rollouts within each selected task, and recomputing the macro-average. For real-robot task success, we report two-sided 95% Wilson intervals from the pooled success counts with $n = 120$ per method and task family. These intervals describe uncertainty under the evaluated settings and do not by themselves establish the significance of pairwise method differences.

G.2 Comparison with Trajectory-Derived Supervision

This experiment examines whether generated video-language supervision provides additional benefits over supervision derived from robot trajectories and states. We use Qwen2.5-VL-7B and evaluate

Table 8: **RQ3: Simulated VLA transfer on LIBERO and RoboTwin 2.0.** We use the native success-rate protocols of both benchmarks. Both Real post-hoc captions and Gen4Understand improve downstream manipulation performance without changing the controller or policy architecture.

Method	LIBERO Success Rate ↑					RoboTwin 2.0 Success Rate ↑			
	Spatial	Object	Goal	Long	Avg	Tool-use	Long-horizon	Distractor	Avg
Qwen2.5-VL-3B-Instruct	76.5	74.8	72.1	65.2	72.1	52.6	47.8	49.3	49.9
+ Real post-hoc captions	80.0 ↑3.6	78.2 ↑3.4	76.2 ↑4.1	69.5 ↑4.3	76.0 ↑3.9	58.0 ↑5.4	53.0 ↑5.2	55.2 ↑5.9	55.4 ↑5.5
+ Gen4Understand	82.5 ↑6.1	80.4 ↑5.6	78.8 ↑6.7	72.2 ↑7.0	78.5 ↑6.4	62.1 ↑9.5	58.0 ↑10.2	59.1 ↑9.8	59.7 ↑9.8
Qwen2.5-VL-7B-Instruct	79.6	77.5	75.0	69.4	75.4	58.2	53.4	54.0	55.2
+ Real post-hoc captions	82.7 ↑3.1	80.4 ↑2.9	78.2 ↑3.2	73.0 ↑3.6	78.6 ↑3.2	63.2 ↑5.0	58.6 ↑5.2	60.0 ↑6.0	60.6 ↑5.4
+ Gen4Understand	85.1 ↑5.5	83.0 ↑5.5	80.9 ↑5.9	75.8 ↑6.4	81.2 ↑5.8	67.0 ↑8.8	62.9 ↑9.5	63.8 ↑9.8	64.6 ↑9.4
RoboBrain2.5-4B	82.8	80.1	77.4	71.7	78.0	61.3	56.2	58.9	58.8
+ Real post-hoc captions	85.7 ↑2.9	82.8 ↑2.7	80.2 ↑2.8	74.5 ↑2.8	80.8 ↑2.8	66.2 ↑4.9	61.0 ↑4.8	63.0 ↑4.1	63.4 ↑4.6
+ Gen4Understand	**88.0** ↑5.2	**85.2** ↑5.1	**83.3** ↑5.9	**77.1** ↑5.4	**83.4** ↑5.4	**70.0** ↑8.7	**65.1** ↑8.9	**67.4** ↑8.5	**67.5** ↑8.7

Table 9: **RQ3: Real-robot transfer.** We evaluate three task families designed to stress result-state recognition, temporal ordering, and distractor resistance. For each model variant, each task family contains three configurations with 40 trials each. Each task-success percentage is computed from the number of successful trials pooled across these configurations, with a denominator of 120. Avg is the unweighted mean of the three family success rates, equivalently the total number of successful trials divided by 360. Task-success averages and gains are computed from integer trial counts before rounding. Arrows denote absolute percentage-point changes from the corresponding backbone's base model. The additional metrics are described in Appendix B.3.

Method	Task Success ↑			Avg ↑	Stage Completion ↑	State-transition Success ↑	Halluc. Error ↓
	Return Brush	Ordered Placement	Distractor-aware Sorting				
Qwen2.5-VL-7B-Instruct	61.7	66.7	56.7	61.7	73.6	67.5	17.8
+ Real post-hoc captions	68.3 ↑6.7	72.5 ↑5.8	64.2 ↑7.5	68.3 ↑6.7	79.0 ↑5.4	74.2 ↑6.7	13.0 ↓4.8
+ Gen4Understand	74.2 ↑12.5	79.2 ↑12.5	70.8 ↑14.2	74.7 ↑13.1	84.8 ↑11.2	80.1 ↑12.6	8.7 ↓9.1
RoboBrain2.5-4B	66.7	71.7	58.3	65.6	75.5	70.2	16.0
+ Real post-hoc captions	72.5 ↑5.8	76.7 ↑5.0	66.7 ↑8.3	71.9 ↑6.4	81.2 ↑5.7	76.5 ↑6.3	11.8 ↓4.2
+ Gen4Understand	**78.3** ↑11.7	**83.3** ↑11.7	**72.5** ↑14.2	**78.1** ↑12.5	**86.4** ↑10.9	**82.2** ↑12.0	**7.3** ↓8.7

on the 800 real clips in the DROID-Robot split. All adapted conditions use the same DROID-only training-instance budget and adaptation hyperparameters.

Trajectory baseline construction. We construct the trajectory/state baseline from synchronized robot proprioception, including end-effector poses, joint states, and gripper aperture. Trajectories are segmented into motion phases at gripper events and velocity sign changes. Each phase is assigned a motion primitive and rendered into captions and QA pairs using the same event-state schema and template families as Gen4Understand. Object identity is obtained from DROID's language instructions. We also evaluate a combined condition using trajectory-derived supervision and real post-hoc captions under the same total instance budget.

Results and interpretation. Table 12 shows that trajectory-only supervision improves temporal-order accuracy over post-hoc captions (57.2 versus 54.9), while their combination further improves it to 58.0. Gen4Understand achieves the highest score on all four components among the evaluated conditions and reaches 58.9 FG. It exceeds trajectory-only supervision by 4.9 points (paired 95% bootstrap CI [3.5, 6.3]) and the combined condition by 3.0 points ([1.7, 4.3]).

These results support the utility of generated video-language supervision beyond the evaluated trajectory-textualization approach. They do not establish that dense video generation is universally necessary: this comparison does not evaluate direct latent-world-model supervision or alternative trajectory-to-VLM alignment architectures.

G.3 Visual-Process Evidence and Question Generalization

We separately perturb visual evidence and question formulation to examine what supports the adapted model's predictions.

Table 10: Simulation averages with 95% bootstrap confidence intervals, corresponding to Table 8.

Method	LIBERO Avg. [95% CI]	RoboTwin Avg. [95% CI]
Qwen2.5-VL-3B	72.1 [68.8, 75.3]	49.9 [42.9, 57.0]
+ Real post-hoc	76.0 [72.8, 79.0]	55.4 [48.6, 61.9]
+ Gen4Understand	78.5 [75.6, 81.3]	59.7 [52.7, 66.8]
Qwen2.5-VL-7B	75.4 [71.8, 78.8]	55.2 [48.9, 61.6]
+ Real post-hoc	78.6 [75.6, 81.4]	60.6 [53.5, 67.3]
+ Gen4Understand	81.2 [77.9, 84.4]	64.6 [58.1, 71.0]
RoboBrain2.5-4B	78.0 [74.6, 81.3]	58.8 [52.3, 65.4]
+ Real post-hoc	80.8 [77.6, 83.8]	63.4 [56.0, 70.5]
+ Gen4Understand	83.4 [80.3, 86.3]	67.5 [61.4, 73.4]

Table 11: Real-robot task-success percentages with two-sided 95% Wilson intervals. Each cell uses 120 trials pooled across three configurations, corresponding to Table 9.

Method	Return Brush	Ordered Placement	Distractor-aware Sorting
Qwen2.5-VL-7B	61.7 [52.7, 69.9]	66.7 [57.8, 74.5]	56.7 [47.7, 65.2]
+ Real post-hoc	68.3 [59.6, 76.0]	72.5 [63.9, 79.7]	64.2 [55.3, 72.2]
+ Gen4Understand	74.2 [65.7, 81.2]	79.2 [71.1, 85.5]	70.8 [62.2, 78.2]
RoboBrain2.5-4B	66.7 [57.8, 74.5]	71.7 [63.0, 79.0]	58.3 [49.4, 66.8]
+ Real post-hoc	72.5 [63.9, 79.7]	76.7 [68.3, 83.3]	66.7 [57.8, 74.5]
+ Gen4Understand	78.3 [70.1, 84.8]	83.3 [75.7, 88.9]	72.5 [63.9, 79.7]

Visual interventions. Temporal-order accuracy is 60.40 with ordered video, but falls to 43.60 with endpoint-only inputs and to 36.80 when interior frames are shuffled while preserving the endpoints and frame count. The decreases of 16.80 and 23.60 points indicate that intermediate frames and their order contribute substantially to temporal judgments.

Question generalization. Question changes produce smaller differences under the reported protocol. FG-Score is 59.93 with seen templates, 58.60 with unseen compositional template families, and 57.40 with independently paraphrased free-form questions.

Conflicting-script probe. In a further 600-video conflicting-script probe, 472 responses follow visual evidence, 85 follow the supplied script, and 43 are mixed or unsupported. A response follows the video only when every conflicting slot matches visual ground truth, and follows the script only when every conflicting slot matches that script. Together, these results support the use of visual process evidence and transfer across question formulations.

H Additional Qualitative Examples

We include additional qualitative examples illustrating three common failure modes of baseline VLMs: wrong result-state prediction, incorrect temporal ordering, and distractor-induced hallucination(Figure 7). For each example, we compare the baseline output, the Gen4Understand-adapted output, and the structured ground-truth event sequence.

I Broader Impact

Gen4Understand may contribute to more reliable embodied AI by improving VLMs' ability to recognize intermediate actions, object state transitions, temporal order, and physically grounded

Supervision	Slot-F1	State	Temporal	Contact/Contain.	FG
Base	49.8	48.2	50.6	53.8	50.6
Trajectory/state	51.4	50.3	57.2	57.1	54.0
Real post-hoc	53.6	52.8	54.9	57.5	54.7
Trajectory + post-hoc	54.0	53.4	58.0	58.2	55.9
Gen4Understand	**57.8**	**57.0**	**60.1**	**60.7**	**58.9**

Table 12: DROID-only adaptation comparisons on Qwen2.5-VL-7B, evaluated on the same 800 real DROID clips used in Table 2. Base denotes the shared unadapted checkpoint. Adapted conditions in this table use DROID-only training data, whereas Table 2 reports adaptation in the main multi-source setting. Adapted conditions use the same training-instance budget and hyperparameters. All scores are on a 0–100 scale, with higher values better. FG is the average of Slot-F1, state-change accuracy, temporal-order accuracy, and contact/containment accuracy.

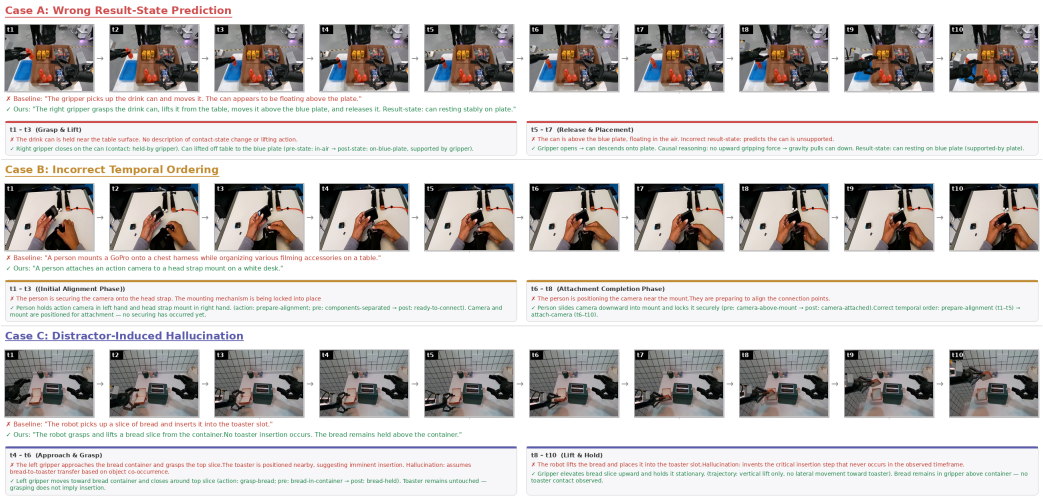

Figure 7: **Qualitative comparison of process-level video understanding.** We illustrate three common failure modes of baseline VLMs: (A) wrong result-state prediction, (B) incorrect temporal ordering, and (C) distractor-induced hallucination. Compared to the baseline, Gen4Understand correctly grounds physical states, tracks temporal progress, and resists semantic distractors.

outcomes in manipulation videos. Such capabilities are useful for assistive robotics, household agents, industrial automation, and safety monitoring, where understanding how an action unfolds is often as important as recognizing the final task. However, using generated videos as supervision also introduces risks: synthetic clips may contain physical artifacts, biased scenarios, or misleading transitions that could be inherited by downstream models. We mitigate these risks through verifier-calibrated filtering based on semantic consistency, endpoint fidelity, physical plausibility, temporal coherence, and visual quality. Nevertheless, models trained with synthetic embodied data should be carefully evaluated before deployment in safety-critical settings, and the generation and filtering pipeline should be transparently documented.

J Limitations

Despite consistent gains, several limitations remain. First, endpoint grounding improves controllability but does not fully eliminate imperfect intermediate dynamics in generated clips, especially for highly articulated motion and subtle contact changes. Second, the verifier-calibrator removes many low-quality samples but cannot guarantee perfect physical validity. Third, hallucination is reduced rather than eliminated; under severe occlusion, visually ambiguous hand-object interactions, or

extremely familiar object configurations, the model may still revert to prior-driven reasoning. Finally, downstream gains are strongest when task difficulty is dominated by perception, progress verification, or temporal disambiguation; tasks dominated by low-level motor control benefit less from improved VLM understanding. Our matched controls reduce differences in supervision structure and student adaptation budget, but generated and real data can still differ in sample selection, visual difficulty, and annotation provenance. All student backbones adapted in our experiments are Qwen-related, so improvements across these backbones do not rule out model-family bias in parser-based selection. Although parser-generated language is not used as student supervision, filtering may favor clips that the parser's model family finds easier to interpret.

The DROID trajectory comparison evaluates state and trajectory signals converted into caption and QA supervision; direct latent-world-model supervision remains unevaluated. Offline data construction also incurs generation, verification, and annotation costs that are separate from the matched student adaptation budget. The present comparisons therefore do not establish superiority over direct latent supervision or greater end-to-end cost efficiency.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix J discusses generation artifacts, verifier and parser bias, the scope of the controls, and end-to-end construction costs.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed methods in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code will be released after accepted.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Yes, we discuss it in the experiment part.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Table 3 and Appendix F.6 report marginal and paired 95% bootstrap confidence intervals. Section 4.4 and Appendix E.2 report and define 95% Wilson intervals for real-robot success proportions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix D.6 analyzes construction complexity, but hardware specifications, peak memory, per-run runtimes, and total compute are not reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Appendices I and J discuss risks, mitigation, and limitations; Appendix E describes data separation and annotation procedures.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix I discusses potential benefits, risks from generated supervision, mitigation through filtering, and caution in safety-critical deployment.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not deem that this paper poses such risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: Existing datasets and model providers are credited, but asset-specific licenses and terms of use are not enumerated in the manuscript.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The Gen4Understand dataset construction details are provided in Appendix E. Data and prompts will be released upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Section 3.3 and Appendix D.1 describe the frozen Qwen2.5-VL-72B event parser used for candidate scoring. Its generated language is not used as student training supervision.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.