Done, But Not Sure: Disentangling World Completion from Self-Termination in Embodied Agents

Ying Chen[1] **Lihuang Fang**[1] **Rui Jiang**[1] **Mingxu Wang**[1]
Zhifeng Gu[1,2] **Lei Yi**[1] **Jie Chen**[1†]
[1]XPENG Robotics [2]The Hong Kong Polytechnic University
Project Page Code

Abstract

Standard embodied evaluations do not independently score whether an agent correctly commits to task completion at episode closure, a capacity we call *terminal commitment*. Behaviorally distinct failures—never completing the task, completing it but failing to stop, and reporting success without sufficient evidence—collapse into the same benchmark failure. We introduce **VIGIL**, an evaluation framework that makes terminal commitment separately observable and scorable. Under VIGIL's default protocol, agents observe only egocentric RGB, receive no action-success signals, and must end each episode with a semantic report checked deterministically against hidden world state. This yields two separate scores: world-state completion (**W**) and benchmark success (**B**), where **B** additionally requires a correct terminal report. Together with report and trajectory diagnostics, this distinguishes missed execution, continued action after attainment, unsupported commitment, and verified success. Across 20 models on 1,000 frozen episodes, systems with comparable **W** differ by up to 19.7 pp in **B**: one model converts achieved states into correct reports, while another with near-identical execution drifts past the goal without closing. An action-feedback intervention further tests the separation: execution-oriented signals improve **W** broadly, yet its effects on benchmark success and reporting failures vary across models. Interface, history, and memory controls produce heterogeneous changes, supporting separate measurement of execution and reporting under the evaluated contracts.

1 Introduction

Embodied agents must not only take actions to complete a task, but also determine when the task has been completed and commit to that judgment. This is non-trivial when agents operate under partial observability and must infer task progress from limited egocentric observations over time [41, 9], without action feedback or success signals [44, 37]. For example, in a task that asks the agent to turn on a lamp, the agent may successfully turn it on yet continue navigating without submitting a completion report. Under current embodied evaluation, this outcome is often indistinguishable from never having completed the task.

To our knowledge, no existing embodied benchmark cleanly separates task-closure failures from execution failures [29, 6, 17]. Recent work has improved per-skill and capability-level diagnostics [22, 5, 37, 25], yet overall model comparison is still summarized by aggregated task- or episode-level success metrics. These metrics collapse behaviorally distinct cases into the same final outcome: the agent may fail to complete the task, complete it but fail to stop, or declare completion without sufficient evidence—failures with different causes and remedies that current metrics conflate.

†Corresponding author.

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

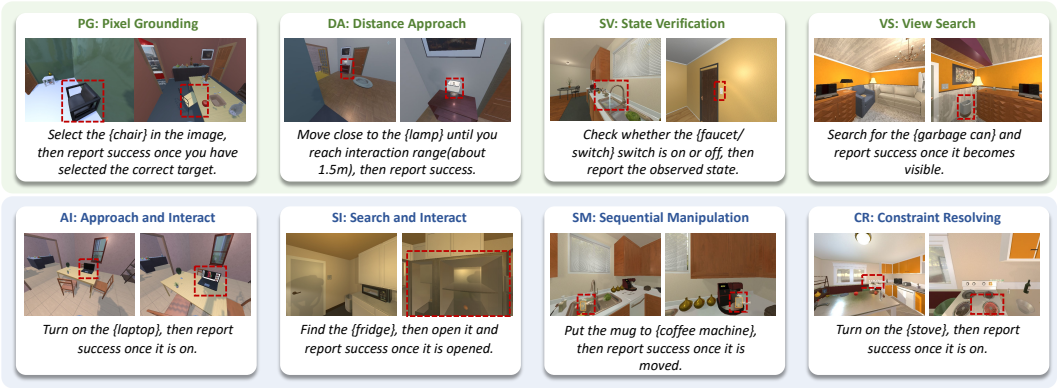


Figure 1: **Controlled evaluation protocol.** VIGIL contains eight task families: a diagnostic tier (PG, DA, SV, VS) that isolates a single bottleneck, and a compositional tier (AI, SI, SM, CR) that combines them in multi-step interaction. All episodes use strict first-person observation and mandatory `report` termination.

We introduce **VIGIL**, an evaluation framework that makes *terminal commitment* separately observable and scorable. The design has three key elements. First, agents observe only egocentric RGB, without privileged state, oracle progress signals, or action-success confirmation. Second, successful closure requires a semantic `report` whose content is checked deterministically against the hidden world state. Third, this yields two separate scores: world-state completion (**W**) and benchmark success (**B**), where **B** additionally requires a correct terminal commitment. The no-feedback baseline controls the evidence available to the agent; action feedback is studied separately as an intervention, since action execution and task completion need not coincide.

This target differs from prior work on self-assessment and uncertainty in language agents [16, 28, 13]. Rather than inferring latent confidence or internal belief [16], we evaluate whether the agent expresses the correct task-state judgment at episode closure through a terminal report that can be checked against the hidden world state. For state-verification tasks, where the agent must report `open/closed` or `on/off`, the correct terminal act is a categorical state judgment, not a binary decision to halt. Consequently, a stop-only protocol cannot represent this class of failure, because the error lies in report content rather than in termination timing.

VIGIL evaluates this dimension over eight task families that probe the conditions under which terminal judgment becomes difficult (Figure 1), spanning target visibility, distance, state uncertainty, temporal dependency, and physical constraint. A diagnostic tier (short budgets, single bottlenecks) isolates closure failures when execution is tractable, while a compositional tier (longer budgets, chained prerequisites) reveals when execution floors mask closure errors. We additionally use an action-feedback intervention, modeled on proprioceptive signals available to physical robots, to examine how execution feedback changes world completion and terminal reporting.

Findings. Across 20 multimodal systems spanning open-weight and closed-source frontier models (10 anchor models in the main text; full panel in Appendix F), we find that execution and terminal commitment are empirically separable:

- **Structured closure-failure profiles.** Models exhibit distinct terminal behaviors, including premature false commitments (**FR**), chronic no-report exhaustion (**NR**), and selective reporting. These profiles are obscured by aggregate success; prompt controls quantify their sensitivity to the reporting interface.
- **Execution floors mask closure failures.** On longer compositional tasks, execution often fails before terminal judgment can be meaningfully evaluated, compressing the observable gap and masking closure errors that the diagnostic tier isolates directly.
- **Execution feedback is not a universal fix.** A limited action-feedback intervention reduces execution traps for many models, but changes in world completion and benchmark success differ across configurations.

Table 1: Representative embodied evaluation settings and whether they make terminal task-state judgment externally scorable. VIGIL makes agent-side terminal commitment independently scorable under a no-feedback native-control contract.

Benchmark	Task Level	Multi-modal	Fine-Grained	Decoupled	Active Termination	No Feedback
ALFRED [29]	High	✓	✗	✗	Stop	✗
ALFWorld (text-only) [30]	High	✗	✗	✗	✗	✗
VLMbench [42]	Low	✓	✗	✗	✗	✗
BEHAVIOR-1K [21]	High	✓	✗	✗	✗	✗
LoTa-Bench [6]	High	✗	✗	✗	Stop	✗
GOAT-Bench [17]	Low	✓	✗	✗	Stop	✗
Embodied Agent Interface [22]	High	✗	✓	✗	✗	✗
EmbodiedBench [37]	High&Low	✓	✓	✗	Stop	✗
EmbodiedEval [5]	High	✓	✗	✗	✗	✗
NativeEmbodied [25]	High&Low	✓	✓	Task	✗	✗
VIGIL (Ours)	High&Low	✓	✓	Task+Report	Report	✓

Together, these results identify terminal commitment as a distinct failure dimension: it is empirically separable from execution, produces consistent step-level patterns, and is not uniformly repaired by the tested feedback intervention. VIGIL enables behavioral diagnosis of terminal reporting under a common prompted interface; it does not identify an interface-independent cognitive capacity. Resources: project website and code repository.

2 Related Work

VIGIL intersects three lines of work: embodied evaluation, self-assessment and confidence-related control, and belief-oriented embodied reasoning. Its scientific target is distinct: whether the agent can correctly judge and report its achieved task state at episode closure, scored independently of execution success. Table 1 summarizes how existing settings compare along this axis.

Column definitions: *Fine-Grained*—per-skill diagnostics beyond aggregate success. *Active Termination*—"Stop": bare stop action absorbed into task success; "Report": semantic terminal judgment independently scored; ✗: evaluator-side termination only. *Decoupled*—"Task": per-skill decomposition without independent report scoring; "Task+Report": adds deterministically scored terminal commitment. *No Feedback*—no external confirmation of task progress, goal completion, or action success.

Embodied Evaluation Protocols. Household instruction-following benchmarks define long-horizon tasks evaluated by terminal success [29, 30, 24, 18]. Subsequent work extends the setting across simulation fidelity [21], LLM-based planning [6], navigation [17], and compositional manipulation [42]. Recent suites add per-skill diagnostics [22, 37, 5, 25], improving attribution across perception, navigation, and manipulation. The key gap remains: terminal success is an evaluator-side predicate over world state, not an agent-side judgment that is independently scored. Execution failures and closure failures are therefore behaviorally conflated.

Self-Assessment, Confidence, and Termination Decisions. Several benchmarks include a `stop` action [29, 6, 17], but stopping is a control primitive absorbed into task success without independent evaluation. A parallel literature asks whether language models possess calibrated self-knowledge [16] and whether confidence signals can trigger help-seeking or replanning [28, 13]. VIGIL targets a different observable: not latent confidence, but whether the agent produces a semantically correct terminal report that can be verified against hidden world state.

Belief Under Embodied Interaction. Spatial intelligence in vision-language models is increasingly studied beyond static images, spanning metric reasoning [3, 4], perspective-taking [15, 39, 34], and robotics-oriented grounding [40, 31, 38, 43]. These settings show that embodied interaction requires building and updating representations from partial observations—a capacity that remains difficult for current models [36, 32]. Recent work makes this explicit: Theory of Space [41] asks whether



Figure 2: **Evaluation pipeline.** The top row illustrates an example trajectory. The bottom row summarizes the per-step interface: the agent acts from only the current egocentric RGB frame, the task instruction, and bounded dialogue history, without action-success or goal-completion feedback. A terminal `report` is evaluated against the hidden world-state condition by deterministic rules.

foundation models construct spatial beliefs through active exploration; CubeBench [9] diagnoses interactive spatial reasoning under partial observation. VIGIL shares this belief-under-interaction perspective but targets a different output: task-state judgment at closure, not spatial belief during exploration.

3 Benchmark Design

VIGIL separates two outcomes that standard embodied evaluation conflates: whether the agent changed the world correctly, and whether it issued a correct terminal report about that change. This requires controlled task families with hidden goal predicates, a no-feedback interaction contract, and a deterministic scoring rule that evaluates world state and report content independently.

3.1 Task Families

Each frozen episode specifies a task instruction, a family label, step and invalid-action budgets, and a hidden success condition $\mathcal{G}$ checked against the hidden evaluation state. The benchmark contains 1,000 episodes across eight balanced families (125 each), organized as controlled probes over the factors that shape task-state judgment under interaction (Figure 1).

Rather than decomposing by domain-specific subtasks—as in recent skill-diagnostic benchmarks [37, 25]—we decompose along atomic perceptual-motor capacities and progressively compose them. Comparing constituent and composed tasks helps localize behavioral difficulties, without uniquely identifying their internal causes.

The *diagnostic tier* isolates single bottlenecks with short budgets (5–20 steps): **PG** (pixel grounding)—click the correct visible object; **DA** (distance approach)—navigate to a visible target; **VS** (view search)—find a non-visible target through active exploration; **SV** (state verification)—report the categorical state of a visible object without physical interaction. SV is designed to begin with the queried predicate satisfied. Its **W** measures whether that predicate remains satisfied and publicly observable at closure, not recognition accuracy; failures in **B** can include label or field-placement mismatch.

The *compositional tier* chains these capacities under longer budgets (25–40 steps): **AI** (approach-and-interact = DA + interaction); **SI** (search-and-interact = VS + DA + interaction); **SM** (sequential manipulation—multi-step pick-and-place); **CR** (constraint resolving—detouring before target interaction). Full family specifications are in Appendix D.

3.2 Native-Control Contract

Episodes run in AI2-THOR [20] using iTHOR scenes and ProcTHOR [8] houses. The agent receives a single egocentric RGB frame per step and acts through four skills: locomotion (`navigate`, `look`), pixel-grounded object interaction (`interact_pixel`), and terminal reporting (`report`). No

privileged state or action-success feedback is provided: the agent receives no maps, semantic masks, absolute pose, or oracle progress signals. State changes are visible only through subsequent RGB observations.

Following the dialogue-based native-control paradigm of recent embodied evaluation [37, 25], the agent maintains a bounded dialogue history of up to 20 turns (one turn per action step)—including any agent-generated `thought` or `cognitive_state` fields—as the only cross-step memory. History is episode-local; older images are not retained in the baseline. All evaluated weights remain unchanged: the action schema, including `report`, is prompted at inference time rather than learned through benchmark-specific fine-tuning.

This contract is intentionally sparse: terminal judgments must be formed from the agent's own partial-observation interaction history rather than from evaluator-side success signals.

The key mechanism is the terminal `report`: it carries a categorical `status` (e.g., `success`, `fail`, `on/off`, `open/closed`) and a short summary. Unlike a bare `stop`, `report` records *what* the agent judges the task state to be at termination. This makes the terminal act externally scorable: the evaluator can check whether the reported judgment matches the hidden world state, rather than merely whether termination coincided with goal satisfaction. This matters most in state-verification tasks, where the correct terminal act is a categorical state judgment rather than a binary decision to end.

3.3 Disaggregated Outcomes

We separate two outcomes at episode closure: whether the task condition is satisfied in the world, and whether the agent's terminal commitment correctly reflects that state.

For trajectory τ, let s_T be the hidden terminal evaluation state, $\mathcal{G}_{\text{sem}}$ the primary semantic task-goal predicate, a_T the terminal action, and $\texttt{report}_T$ the report content. s_T includes simulator state and required episode records: PG grounding hits, latched VS visibility, and a qualifying target-interaction record for applicable object-state tasks (Appendix C). These records are independent of report correctness. The report status is normalized to one of eight categorical values (`success`, `fail`, `on`, `off`, `open`, `closed`, `unsafe`, `invalid`) and matched against the hidden state by a deterministic rule. For achievement tasks, `success` matches $W = 1$ and negative reports match $W = 0$. For SV, $\mathcal{G}_{\text{sem}}$ includes the queried state and its public observability; the report must match the evaluator-derived categorical label, with an unavailable-evidence fallback detailed in Appendix C.3. We define:

$$W(\tau) = \mathbb{I}[s_T \models \mathcal{G}_{\text{sem}}],$$
$$B(\tau) = \mathbb{I}[s_T \models \mathcal{G}_{\text{sem}} \ \wedge \ a_T = \texttt{report} \ \wedge \ \texttt{match}(\texttt{report}_T, s_T)]\,.$$

W captures world-state completion at termination under the primary semantic task predicate. B captures benchmark success, requiring not only the correct semantic world state but also a matching terminal commitment. If an episode ends by budget exhaustion or invalid-action limit before `report`, then $B = 0$ regardless of the world state. The gap $\Delta = W - B$ counts terminally satisfied states not converted into benchmark success.

Direct reporting measures. Let q_i indicate a submitted report and m_i its exact correctness under the matching rule (zero for a missing report or an unrecognized status). For N episodes, report coverage is $C_R = N^{-1} \sum_i q_i$, all-episode correctness is $R = N^{-1} \sum_i q_i m_i$, and conditional correctness is $R_{|\text{rep}} = \sum_i q_i m_i / \sum_i q_i$ (undefined if no reports). These measures include both $W = 1$ and $W = 0$, whereas B remains a joint outcome. Appendix L also separates any-time attainment from terminal W and reports achievement-task confusion counts.

Importantly, Δ is a decomposition handle rather than a standalone reliability measure: a small gap may reflect either reliable state-coupled commitment or aggressive reporting that coincidentally aligns with low W. We therefore analyze Δ jointly with deterministic closure-failure labels: **False reports** (**FR**) occur when the report claims a status unsupported by the hidden state. **No-report exhaustion** (**NR**) occurs when the step budget expires without a `report`. **Invalid-action-limit** terminations (**IL**) occur when protocol failures and invalid executions together exceed the family-specific limit. Appendix C.2 summarizes the reported labels used in the figures and tables.

Table 2: **Per-family outcomes for 10 anchor models under native control.** Each cell reports **W/B** (%), where **W** = world-state completion and **B** = benchmark success. **Δ** = $W - B$ gap (pp). Diagnostic: PG = pixel grounding, DA = distance approach, VS = view search, SV = state verification. Compositional: AI = approach-and-interact, SI = search-and-interact, SM = sequential manipulation, CR = constraint resolving. Anchor set for the main cross-model and feedback analyses; primary 20-model panel in Appendix F. `(T)` marks thinking serving mode; Qwen3.6-27B and Qwen3.5-27B also use thinking.

Model	All	Δ	Diagnostic Probes				Compositional Tasks			
	W/B		PG	DA	VS	SV	AI	SI	SM	CR
Gemini-3.1-Pro†	**57.7/56.4**	**1.3**	**92.0/92.0**	52.8/52.0	**71.2/71.2**	**95.2/90.4**	**62.4/60.8**	**26.4/25.6**	**43.2/42.4**	**18.4/16.8**
Qwen3.6-27B	46.7/37.8	8.9	87.2/76.8	56.8/**55.2**	47.2/44.0	93.6/80.8	54.4/28.8	12.8/7.2	14.4/6.4	7.2/3.2
Doubao-Seed-1.8†	40.4/37.1	3.3	76.8/75.2	44.8/41.6	29.6/29.6	94.4/80.0	42.4/37.6	5.6/4.8	19.2/17.6	10.4/10.4
Qwen3.5-27B	42.0/36.2	5.8	84.0/80.8	41.6/39.2	48.0/44.8	93.6/85.6	36.8/19.2	15.2/8.0	10.4/9.6	6.4/2.4
Qwen3-VL-32B `(T)`	30.7/26.4	4.3	70.4/68.0	10.4/10.4	43.2/41.6	93.6/81.6	15.2/4.8	2.4/0.8	5.6/3.2	4.8/0.8
GPT-5.4†	29.2/24.7	4.5	30.4/30.4	24.8/24.8	64.8/62.4	92.8/72.0	10.4/0.8	4.0/1.6	4.8/4.0	1.6/1.6
MiMo-Embodied-7B	29.5/21.3	8.2	52.0/34.4	37.6/24.0	31.2/26.4	**95.2**/74.4	14.4/7.2	0.8/0.8	3.2/1.6	1.6/1.6
Claude-Sonnet-4†	25.2/21.1	4.1	38.4/35.2	17.6/16.8	65.6/64.8	73.6/45.6	4.8/4.8	0.0/0.0	1.6/1.6	0.0/0.0
Qwen3-VL-32B	39.2/18.9	20.3	68.8/20.0	44.0/14.4	65.6/48.8	82.4/51.2	34.4/8.0	5.6/0.8	7.2/5.6	5.6/2.4
InternVL3.5-38B	39.9/17.4	22.5	79.2/16.0	**57.6**/18.4	50.4/36.8	86.4/55.2	31.2/7.2	7.2/0.0	5.6/4.8	1.6/0.8

†Closed-source API models. Identifiers: `gemini-3.1-pro-preview`, `doubao-seed-1.8`, `gpt-5.4`, `claude-sonnet-4-20250514`.

Interpretation Note. VIGIL does not claim that every failure labeled by $B = 0$ uniquely identifies an internal cognitive cause. Rather, it provides an externally testable decomposition of closure behavior: whether the agent achieved the relevant task state, whether it issued a terminal judgment, and whether that judgment matched the achieved world state. Exact correctness measures compliance with an operational reporting contract, not pure semantic judgment. The auxiliary summary-state audit (Appendix M) makes some field-placement and linguistic effects visible without replacing the exact score.

4 Experiments

The primary comparison uses the native-control contract from §3: egocentric RGB input, bounded dialogue history, no environment feedback, and mandatory `report` termination. Main-text world-state results (**W**) use the primary semantic goal predicate; dual-metric scoring details are in Appendix C. We evaluate a frozen 1,000-episode set (eight families, 125 each) on 10 anchor models spanning closed-source frontier (Gemini-3.1-Pro [10], Doubao-Seed-1.8 [11], GPT-5.4 [23], Claude-Sonnet-4 [1]), open general VLMs (Qwen3.6-27B [27], Qwen3.5-27B [26]—both served with thinking enabled—Qwen3-VL-32B [2] in Instruct and Thinking variants, and InternVL3.5-38B [33]), and embodied-tuned systems (MiMo-Embodied-7B [12]). This spread tests whether terminal commitment failures are specific to a model class or appear broadly across architectures, scales, and training objectives; the primary 20-model comparison panel is in Appendix F.

The empirical goal is not only to compare benchmark success, but to characterize how embodied models fail at episode closure. We ask three questions: do execution and terminal commitment separate empirically (§4.1)? Do models exhibit distinct closure-failure profiles with interpretable step-level signatures (§4.2)? How do these profiles change under limited action feedback (§4.3)?

4.1 Cross-Model Outcomes: Execution and Terminal Commitment Separate

Table 2 summarizes per-family outcomes for the 10 anchor models.

Execution and terminal commitment separate empirically. Comparable levels of world-state completion can produce sharply different benchmark success. Doubao-Seed-1.8 and InternVL3.5-38B reach nearly identical **W** (40.4% vs. 39.9%), yet differ by 19.7 pp in **B** (37.1% vs. 17.4%); their sharply different closure gaps ($\Delta = 3.3$ vs. 22.5 pp) explain this benchmark-success divergence. A related decoupling appears between Qwen3.6-27B (46.7%/37.8%) and InternVL3.5-38B

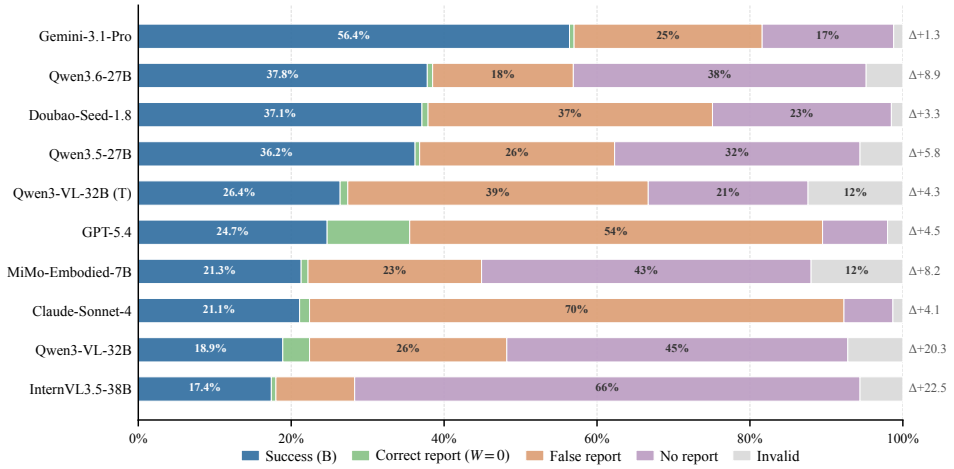


Figure 3: **Disjoint episode outcomes for 10 anchor models** (sorted by **B**). Correct reports with $W = 0$ are separate from **B**; false negatives belong to FR. FR-heavy profiles (Claude-Sonnet-4, GPT-5.4) and NR-heavy profiles (InternVL3.5-38B, Qwen3-VL-32B) are immediately distinguishable despite comparable aggregate success. Label definitions are in Appendix C.2; counterfactual report-policy analysis is in Appendix G.

(39.9%/17.4%). The reverse also holds: Qwen3-VL-32B (`T`) has lower **W** than its instruct counterpart (30.7% vs. 39.2%) but higher **B** (26.4% vs. 18.9%), showing that higher execution does not guarantee higher benchmark success.

The compositional task structure supports comparison between diagnostic and composed tasks. Qwen3.6-27B performs well on PG (87.2%) and reasonably on DA (56.8%) individually, but on AI attains 54.4% **W** and only 28.8% **B**, suggesting that the composed-task bottleneck is not merely constituent navigation or grounding but also conversion of successful interaction into terminal commitment. For open-weight configurations with paired robustness runs, prompt-sensitivity and repeated-run checks quantify variation across the tested prompt variants and reruns (Appendix J).

Compositional tasks reveal where execution floors mask closure failures. Large W–B separations concentrate in diagnostic probes and the simplest compositional task (AI), where execution remains high enough for closure behavior to be observable. On deeper compositional tasks (SI, SM, CR), the gap shrinks because successful world-state attainment itself becomes rare—since Δ is bounded by **W**, low execution compresses the observable signature of closure failure. The compositional tier therefore shows when execution becomes the dominant bottleneck and masks the closure failures that shorter episodes isolate directly.

4.2 Terminal Commitment: Structured Failures and Step-Level Evidence

We now decompose terminal failure modes and examine their step-level signatures to determine whether closure failures have interpretable behavioral structure or are merely stochastic noise.

False commitments and missed commitments define distinct closure regimes. Figure 3 shows that terminal failure takes two dominant forms: unsupported commitment (FR—the agent issues a terminal report whose status is contradicted by the hidden world state) and missed terminal commitment (NR—the agent exhausts the budget without issuing a report). Within terminal $W = 1$ episodes, NR indicates missed reporting despite a satisfied goal; this is distinct from reaching and subsequently losing the goal. FR-heavy models (GPT-5.4 at 54.0%, Claude-Sonnet-4 at 69.9%) frequently issue terminal judgments unsupported by the achieved world state. NR-heavy models (InternVL3.5-38B at 66.1%, Qwen3-VL-32B at 44.6%, MiMo-Embodied-7B at 43.1%) often exhaust the budget without a terminal report; among episodes where the world goal is reached, this appears as failure to convert task-state attainment into terminal commitment. Qwen3-VL-32B combines both modes—substantial FR and NR—producing a Δ of 20.3 pp despite nontrivial **W**, demonstrating that the two failure regimes are not mutually exclusive.

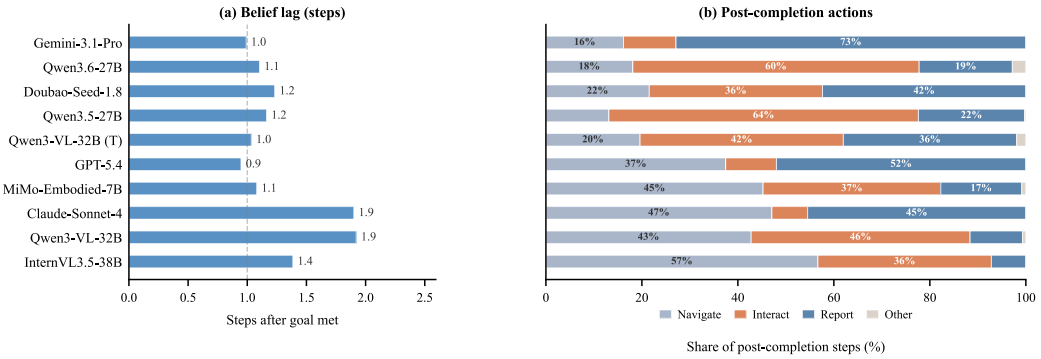


Figure 4: **Terminal-commitment profiles** (sorted by **B** descending). (a) Mean belief lag: steps from the first recorded post-action $W = 1$ to the correct terminal report, conditional on terminal $W = 1$ and a correct report. Model means are 0.9–1.9 steps across all families, including SV (rounded to one decimal place). (b) Among $W = 1$ episodes, action counts are pooled over steps *after* the first recorded post-action $W = 1$ (bars sum to 100%). Navigate groups `navigate` and `look`; Interact denotes `interact_pixel`. NR-heavy models continue navigating or interacting; Gemini-3.1-Pro most often reports.

State verification makes semantic terminal judgment indispensable. SV isolates report content under a preset state and does not require manipulation: **W** exceeds 80% for 9 of 10 anchors (Table 2). Yet **B** drops by 5–31 pp, with the largest gaps in Qwen3-VL-32B (82.4→51.2), InternVL3.5-38B (86.4→55.2), and Claude-Sonnet-4 (73.6→45.6). Because the correct terminal output is categorical (`on`/`off`, `open`/`closed`), a stop-only protocol can observe whether the agent terminates but not whether its terminal state judgment is semantically correct—the exact endpoint measures a categorical reporting contract. For RoboBrain, the auxiliary audit finds that 105/125 summaries agree with the scorable terminal state despite zero exact SV success; this is not zero state-recognition ability (Appendix M).

A single scalar cannot characterize closure reliability. Gemini-3.1-Pro and Claude-Sonnet-4 have similarly small gaps ($\Delta \approx 1$–4 pp; Table 2), yet their closure regimes differ sharply (Figure 3): Gemini combines high **W** (57.7%) with moderate FR (24.6%), whereas Claude-Sonnet-4 reaches much lower **W** (25.2%) but reports aggressively (FR = 69.9%). Claude-Sonnet-4's small gap is not evidence of reliable closure; it reflects frequent terminal commitment despite weak state support, where low **W** limits the number of achieved states available to expose missed conversions. We therefore interpret closure behavior through the joint pattern over **W**, **B**, FR, NR, and IL rather than any single scalar.

Correct closure has a short mean recorded lag. Figure 4a reports model-wise mean lags of 0.9–1.9 steps within the correctly closed subset. These are conditional means, not episode-level upper bounds or direct measures of recognition time. They pool all families, including SV, where a correct first-step report contributes zero lag because the initial $t = 0$ state is excluded. Appendix H gives the counting convention and exact counts.

Most false-success reports receive zero evaluator progress. Across all 10 anchor models, 65–88% of false-success reports have a recorded evaluator progress of zero (Table 13). This score depends on the task predicate: several interaction predicates are binary, so navigation or intermediate actions can leave the score at zero even when they advance execution. Zero therefore does not establish the absence of task-relevant change. The remaining reports mostly have intermediate scores; scores above 0.75 are rare or absent. These statistics describe the evaluator's recorded progress, not a common measure of distance traveled or subgoals completed across families (Appendix H). The false-success label establishes that the claimed goal is unsatisfied, but does not distinguish mistaken state judgment from use of `report` as a default exit.

NR-heavy models drift after task-state attainment instead of closing. InternVL3.5-38B has 201 unreported episodes among its 399 terminal $W = 1$ episodes; the majority of post-completion

Table 3: **Diagnostic intervention: action feedback modeled on proprioceptive signals.** Base = no-feedback contract; +FB = adds `too_far` and `path_blocked` boolean signals. Event counts are per-episode means; outcome changes are in pp. Five +FB retry cells are corrected from complete traces (Appendix I).

	too_far		path_blk		retry		Outcome		Closure	
Model	Base	+FB	Base	+FB	Base	+FB	**ΔW**	**ΔB**	**ΔFR**	**ΔNR**
Gemini-3.1-Pro†	1.29	0.73	2.76	1.12	0.78	0.16	+13.9	+13.0	−9.2	−4.6
Doubao-Seed-1.8†	1.09	0.50	4.40	2.28	1.01	0.11	+12.1	+12.6	−4.6	−8.3
Qwen3.6-27B	2.60	0.73	5.15	2.49	3.56	0.65	+8.0	+12.1	+4.0	−17.6
Qwen3.5-27B	3.19	0.75	4.15	2.24	2.37	0.37	+11.3	+12.1	−0.1	−15.8
GPT-5.4†	0.18	0.27	2.22	1.48	0.01	0.01	+4.5	+2.3	−3.6	+3.6
Claude-Sonnet-4†	0.18	0.24	0.78	1.31	0.18	0.76	+2.7	+0.9	−23.0	+8.5
Qwen3-VL-32B	1.27	0.41	5.29	1.93	2.38	2.28	+2.6	+3.0	+1.5	−8.8
Qwen3-VL-32B (T)	1.99	0.80	1.55	1.07	3.70	1.48	+4.0	+1.4	+4.1	−5.0
InternVL3.5-38B	0.86	0.50	11.94	7.38	1.52	1.19	−0.6	+1.5	−4.0	+0.7
MiMo-Embodied-7B	2.07	0.67	4.24	2.43	4.24	2.46	+3.0	+0.0	+0.6	−7.1

actions are navigation or look commands, with only a small fraction being `report`. A similar pattern holds for Qwen3-VL-32B and MiMo-Embodied-7B. By contrast, Gemini-3.1-Pro issues `report` as the dominant post-completion action, with only 4 unreported $W = 1$ episodes and a small Δ.

4.3 Do Closure Failures Persist After Partial Execution Improvement?

The preceding analysis identifies distinct terminal-failure profiles under the no-feedback contract. A natural question is whether these profiles are downstream consequences of recurrent execution traps (e.g., `path_blocked`, `too_far`) that consume the step budget before reliable closure can occur. To test this, we add a minimal action-feedback intervention: two booleans after each action—`too_far` (interaction beyond the proximity threshold) and `path_blocked` (navigation obstructed). These signals operationalize proprioceptive and low-level controller feedback in physical robots (e.g., out-of-reach manipulation and blocked motion). Crucially, they expose execution outcomes only, not goal completion, task progress, or whether a report should be issued. This probes two execution cues without eliminating all sources of execution difficulty. Table 3 reports paired runs for all 10 anchor models.

Execution improvement does not propagate uniformly to closure. Nine of ten models reduce `path_blocked` events under feedback, and most reduce `too_far`. However, Gemini-3.1-Pro, Doubao-Seed-1.8, Qwen3.5-27B—and partially Qwen3.6-27B—show large gains in **B** (up to $+13$ pp) together with substantial gains in **W**. For the first three, **W** rises by up to $+14$ pp; Qwen3.6-27B attains $+12.1$ pp in **B** but only $+8.0$ pp in **W**. These four configurations combine relatively small baseline closure gaps (1.3–8.9 pp) with lower FR than GPT-5.4 and Claude-Sonnet-4. The paired aggregate changes show that feedback can improve both outcomes, but do not identify which episodes convert from one failure mode to another. For Qwen3.6-27B, $\Delta B = +12.1$ pp exceeds $\Delta W = +8.0$ pp, reducing the aggregate closure gap by 4.1 pp.

NR-heavy and FR-heavy closure failures persist under partial execution repair. The remaining six models show a dissociation between execution improvement and terminal commitment, through three patterns. *FR-heavy persistence*: GPT-5.4 gains +4.5 pp in **W** but only +2.3 pp in **B**; its FR barely moves (ΔFR $= -3.6$ pp, remaining above 50%). *Opposing FR/NR shifts*: Claude-Sonnet-4 is the most extreme case—feedback increases mean steps from 6.8 to 10.8 (paired logs in Appendix I) and FR drops by 23 pp, but NR rises by 8.5 pp, while **B** improves only 0.9 pp. These aggregate changes do not establish episode-level conversion. *NR-heavy persistence*: InternVL3.5-38B reduces `path_blocked` from 11.94 to 7.38 per episode yet gains only +1.5 pp in **B**; its NR changes little (ΔNR $=$ +0.7 pp), so this feedback does not remove its NR-heavy profile. For MiMo-Embodied-7B and the two Qwen3-VL-32B variants, **W** gains are +2.6 to +4.0 pp and **B** gains are 0.0 to +3.0 pp, while FR holds steady or increases (Table 3). These aggregate changes do not identify

episode-level transitions or isolate their causes. These heterogeneous changes motivate measuring execution and terminal reporting separately.

4.4 Companion Metrics and Interaction Controls

Table 4 summarizes companion outcomes and interface controls. Across four primary evaluation runs, final-state scoring misses at most 0.5 pp of achievement, and at most 1.1% of reached goals are lost at closure under the existing predicates, including persistent PG/VS success. Direct report metrics separate coverage from correctness but remain protocol-dependent. Appendix L gives SV retention and confusion counts.

Table 4: **Companion outcomes and interface controls.** (a) Primary evaluation runs: attainment uses 875 achievement episodes; report measures use all 1,000 episodes. Rates are percentages. (b) Fresh matched controls: treatment-minus-baseline changes in percentage points. Stop uses 875 achievement episodes; S_{stop} is a judgment-free endpoint, so $\Delta E = S_{\text{stop}} - B_{\text{baseline}}$ compares different endpoints and is not report-accuracy improvement.

(a) Attainment and direct report correctness

Model	Any-time/final W	Lost/reached	C_R	R	$R_{\|\text{rep}}$
RoboBrain2.5-4B	17.8/17.8	0.0	74.4	15.3	20.6
Qwen3-VL-8B	17.3/17.1	0.7	94.6	31.6	33.4
InternVL3.5-38B	33.5/33.3	0.7	28.3	18.0	63.6
Qwen3.6-27B	40.5/40.0	1.1	56.9	38.5	67.7

(b) Fresh controls: no common recovery across models

Model	Neutral $\Delta W/\Delta B$	ACK $\Delta W/\Delta B$	Stop $\Delta W/\Delta E$
RoboBrain2.5-4B	$-1.1/-0.7$	$+0.1/+0.4$	$-1.1/-10.4$
Qwen3-VL-8B	$-0.5/-1.0$	$+0.7/+0.8$	$+2.3/+1.3$
InternVL3.5-38B	$-0.4/+0.9$	$-0.4/-1.5$	$-2.4/+4.1$
Qwen3.6-27B	$-1.8/-0.4$	$+1.1/+0.3$	$+2.4/+1.0$

Neutral wording and content-free acknowledgement (ACK) change report coverage by 2.3–13.4 pp and -2.5 to $+14.7$ pp, respectively, without a common W/B recovery. History and memory effects are also model-dependent (Appendix N). These descriptive controls establish interface sensitivity, not interface independence or a unique cognitive cause.

5 Conclusion and Limitations

VIGIL separates world completion, reporting behavior, and joint success under an RGB-based contract. Across 20 models, systems with comparable W differ by 19.7 pp in B. Feedback, interface, history, and memory interventions have model-dependent effects. Direct reporting metrics and the auxiliary audit qualify exact-score interpretation while retaining the execution–closure distinction.

Limitations. Quantitative results are restricted to AI2-THOR [20] using iTHOR scenes and ProcTHOR [8] houses. The models are evaluated under a prompted action schema without benchmark-specific fine-tuning. Exact scores include interface compliance; the deterministic summary audit covers only explicit state language and is not general semantic understanding. Ten historical SV setup defects require exclusion sensitivity (Appendix M.1). Controls are descriptive; the memory package also changes feedback. Optional reporting, confidence elicitation, a separate stop-and-judgment interface, cross-episode memory, and a second simulator remain untested. Four qualitative physical-robot episodes demonstrate a hardware loop, not quantitative sim-to-real validity or a real-world cost estimate (Appendix O). Behavioral labels do not uniquely identify perception, memory, planning, or termination mechanisms.

Broader impacts. False completion reports can trigger premature handoffs; missing reports can require timeout or human intervention. Their real-world cost ratio is unmeasured. Explicit diagnosis may improve reliability, but contract-specific success does not establish general competence or calibrated self-knowledge, especially in safety-critical deployments.

Acknowledgements and Disclosures

This work was supported by XPENG Robotics. The authors received no third-party funding or support for this work during the 36 months preceding submission. The authors' affiliations with XPENG Robotics are disclosed on the title page; the authors declare no additional competing financial interests related to this work.

References

[1] Anthropic. Introducing Claude 4. `https://www.anthropic.com/news/claude-4`, May 2025.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Boyuan Chen, Zhuo Xu, Sean Kirmani, Danny Driess, Pete Florence, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities. In *CVPR*, 2024.

[4] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. SpatialRGPT: Grounded Spatial Reasoning in Vision-Language Models. In *NeurIPS*, 2024.

[5] Zhili Cheng, Yuge Tu, Ran Li, Shiqi Dai, Jinyi Hu, Shengding Hu, Jiahao Li, Yang Shi, Tianyu Yu, Weize Chen, et al. EmbodiedEval: Evaluate Multimodal LLMs as Embodied Agents. *arXiv preprint arXiv:2501.11858*, 2025.

[6] Jae-Woo Choi, Youngwoo Yoon, Hyobin Ong, Jaehong Kim, and Minsu Jang. LoTa-Bench: Benchmarking Language-oriented Task Planners for Embodied Agents. In *ICLR*, 2024.

[7] Ronghao Dang, Jiayan Guo, Bohan Hou, Sicong Leng, Kehan Li, Xin Li, Jiangping Liu, Yunxuan Mao, Zhikai Wang, Yuqian Yuan, Minghao Zhu, Xiao Lin, Yang Bai, Qian Jiang, Yaxi Zhao, Minghua Zeng, Junlong Gao, Yuming Jiang, Jun Cen, Siteng Huang, Liuyi Wang, Wenqiao Zhang, Chengju Liu, Jianfei Yang, Shijian Lu, and Deli Zhao. RynnBrain: Open Embodied Foundation Models. *arXiv preprint arXiv:2602.14979*, 2026.

[8] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In *NeurIPS*, 2022.

[9] Huan-ang Gao, Zikang Zhang, Tianwei Luo, Kaisen Yang, Xinzhe Juan, Jiahao Qiu, Tianxing Chen, Bingxiang He, Hao Zhao, Hao Zhou, Shilong Liu, and Mengdi Wang. CubeBench: Diagnosing Interactive, Long-Horizon Spatial Reasoning under Partial Observations. In *ICLR*, 2026.

[10] Google DeepMind. Gemini 3.1 Pro model card. `https://deepmind.google/models/model-cards/gemini-3-1-pro/`, 2026.

[11] Dong Guo, Faming Wu, Feida Zhu, et al. Seed1.5-VL Technical Report. *arXiv preprint arXiv:2505.07062*, 2025.

[12] Xiaoshuai Hao, Lei Zhou, Zhijian Huang, Zhiwen Hou, Yingbo Tang, Lingfeng Zhang, Guang Li, Zheng Lu, et al. MiMo-Embodied: X-Embodied Foundation Model Technical Report. *arXiv preprint arXiv:2511.16518*, 2025.

[13] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner Monologue: Embodied Reasoning through Planning with Language Models. In *CoRL*, 2022.

[14] 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. *arXiv preprint arXiv:2502.21257*, 2025.

[15] Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. OmniSpatial: Towards Comprehensive Spatial Reasoning Benchmark for Vision Language Models. In *ICLR*, 2026. See also arXiv preprint arXiv:2506.03135.

[16] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Vu, et al. Language Models (Mostly) Know What They Know. *arXiv preprint arXiv:2207.05221*, 2022.

[17] Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. GOAT-Bench: A Benchmark for Multi-Modal Lifelong Navigation. In *CVPR*, 2024.

[18] Taewoong Kim, Cheolhong Min, Byeonghwi Kim, Jinyeon Kim, Wonje Jeung, and Jonghyun Choi. ReALFRED: An Embodied Instruction Following Benchmark in Photo-Realistic Environments. In *ECCV*, 2024.

[19] Kimi Team, Angang Du, et al. Kimi-VL Technical Report. *arXiv preprint arXiv:2504.07491*, 2025.

[20] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

[21] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martin-Martin, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. BEHAVIOR-1K: A Benchmark for Embodied AI with 1,000 Everyday Activities and Realistic Simulation. In *CoRL*, 2022.

[22] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, et al. Embodied Agent Interface: Benchmarking LLMs for Embodied Decision Making. In *NeurIPS*, 2024.

[23] OpenAI. Introducing GPT-5.4. `https://openai.com/index/introducing-gpt-5-4/`, 2026.

[24] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. TEACh: Task-Driven Embodied Agents That Chat. In *AAAI*, 2022.

[25] Bo Peng, Pi Bu, Keyu Pan, Xinrun Xu, Yinxiu Zhao, Miao Chen, Yang Du, Lin Li, Jun Song, and Tong Xu. How Foundational Skills Influence VLM-based Embodied Agents: A Native Perspective. In *AAAI*, 2026. See also arXiv preprint arXiv:2602.20687.

[26] Qwen Team. Qwen3.5: Towards native multimodal agents. `https://qwen.ai/blog?id=qwen3.5`, February 2026.

[27] Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model. `https://qwen.ai/blog?id=qwen3.6-27b`, April 2026.

[28] Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots That Ask For Help: Uncertainty Alignment for Large Language Model Planners. In *CoRL*, 2023.

[29] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *CVPR*, 2020.

[30] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *ICLR*, 2021.

[31] Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. RoboSpatial: Teaching Spatial Understanding to 2D and 3D Vision-Language Models for Robotics. In *CVPR*, 2025.

[32] Qineng Wang, Baiqiao Yin, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, Saining Xie, Manling Li, Jiajun Wu, and Li Fei-Fei. Spatial Mental Modeling from Limited Views. In *ICLR*, 2026.

[33] Weiyun Wang, Zhangwei Gao, Lixin Gu, Zhe Chen, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[34] Xingrui Wang, Wufei Ma, Tiezheng Zhang, Celso M de Melo, Jieneng Chen, and Alan Yuille. Spatial457: A Diagnostic Benchmark for 6D Spatial Reasoning of Large Multimodal Models. In *CVPR*, 2025.

[35] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

[36] Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. In *CVPR*, 2025.

[37] Rui Yang, Hanyang Chen, Junyu Zhang, et al. EmbodiedBench: Comprehensive Benchmarking Multi-modal Large Language Models for Vision-Driven Embodied Agents. In *ICML*, 2025. See also arXiv preprint arXiv:2502.09560.

[38] Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, Zihan Zheng, Yifan Xu, Muhan Wang, Daohan Lu, Rob Fergus, Yann LeCun, Li Fei-Fei, and Saining Xie. Cambrian-S: Towards Spatial Supersensing in Video. *ICLR*, 2026.

[39] Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. MMSI-Bench: A Benchmark for Multi-Image Spatial Intelligence. In *ICLR*, 2026.

[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. In *CoRL*, 2024.

[41] Pingyue Zhang, Zihan Huang, Yue Wang, Jieyu Zhang, Letian Xue, Zihan Wang, Qineng Wang, Keshigeyan Chandrasegaran, Ruohan Zhang, Yejin Choi, et al. Theory of Space: Can Foundation Models Construct Spatial Beliefs through Active Exploration? In *ICLR*, 2026. See also arXiv preprint arXiv:2602.07055.

[42] Kaizhi Zheng, Xiaotong Chen, Odest Jenkins, and Xin Eric Wang. VLMbench: A Compositional Benchmark for Vision-and-Language Manipulation. In *NeurIPS*, 2022.

[43] Shijie Zhou, Alexander Vilesov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Eric Xin Wang, and Achuta Kadambi. VLM4D: Towards Spatiotemporal Awareness in Vision Language Models. In *ICCV*, 2025.

[44] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *CoRL*, 2023.

A System Prompt Specification

The system prompt is assembled programmatically from four blocks in fixed order. No privileged simulator state crosses the agent–evaluator boundary. The baseline prompt blocks are shown below. The full prompt text and the neutral variant are included in the supplementary material; the prompt-sensitivity experiment is described in Appendix J.2.

Block 1: Task

```
You are an embodied agent in a simulated 3D environment with an egocentric
(first-person) view.
Each step you only see local observations; plan and act under partial
observability.
Ground decisions in the task, the current observation, recent observation
history when available, and the allowed actions schema.
Actions are embodied attempts in the scene, not symbolic shortcuts.
Use only public evidence from this episode.  Do not assume hidden simulator
state or hidden success signals.
For state-recognition tasks, report the observed state label (for example:
on/off or open/closed) rather than mapping the state to task success/fail.
Mere visibility, proximity, or an attempted action is not enough; do not report
success from visual plausibility alone.
Do not claim completion from visual plausibility alone when the task requires
changing world state.
Task:  <TASK_INSTRUCTION>
```

Block 2: Action Schema

```
## Allowed actions - navigate | look | interact_pixel | report
```

Each skill is serialized from the action registry with its required/optional arguments, type constraints, and usage notes. The full specification matches Tables 5–7 (Appendix B) verbatim.

Block 3: Grounding Rules

```
## Grounding Rules - Use only the frame whose 'role' is 'current' for visibility
judgments and interact_pixel targeting. - interact_pixel acts on the object
currently under the specified location in the current RGB frame. - Do not
assume hidden success state; rely on subsequent public observations and allowed
feedback fields from this episode.
```

For models using normalized coordinates, an additional rule is prepended: `For interact_pixel, output x and y in normalized_1000 coordinates:  integers in [0, 1000], where (0,0) is the top-left and (1000,1000) is the bottom-right corner.`

Block 4: Output Format

```
## Output format Return exactly one JSON object and nothing else. Required keys:
'skill_name', 'arguments'. Optional keys:  'thought', 'cognitive_state'. Use
the exact action names from the allowed-actions schema above; do not invent
aliases or synonyms. Do not output prose, markdown, code fences, or multiple
JSON objects. If you include 'thought', keep it to one short sentence.
```

B Action Space and Termination Contract

B.1 Skill Definitions

The action space consists of four skills: two for locomotion (`navigate`, `look`), one for object interaction (`interact_pixel`), and one for terminal self-report (`report`). Table 5 provides the complete specification.

Table 5: **Complete action-space specification.** **R** denotes required; **C** denotes conditionally required (required unless intent is `drop`).

Skill	Argument	Type	Description	Req.
`navigate`	`mode`	enum	`forward`, `backward`, `turn_left`, `turn_right`	R
	`magnitude`	number	Step count (fwd/bwd) or rotation in degrees (turns)	R
`look`	`direction`	enum	`up`, `down`	R
	`magnitude`	number	Pitch change in degrees	R
`interact_pixel`	`intent`	enum	One of eight canonical intents (Table 6)	R
	`x`	int	Horizontal pixel coordinate in the current RGB frame	C
	`y`	int	Vertical pixel coordinate in the current RGB frame	C
`report`	`status`	enum	Terminal status (Table 7)	R
	`summary`	string	Brief justification for the chosen status	R

B.2 Canonical Interaction Intents

Table 6 lists the eight canonical intents accepted by `interact_pixel`. Common aliases (e.g., `open` $\rightarrow$ `open_access`, `toggle_on` $\rightarrow$ `activate`) are normalized at runtime; models may emit either form.

Table 6: **Canonical intents** for `interact_pixel`.

Intent	Semantics	Requires (x, y)
`ground`	Point/click/locate without changing world state	Yes
`open_access`	Open a container or access point	Yes
`close_access`	Close a container or access point	Yes
`activate`	Turn on / toggle on	Yes
`deactivate`	Turn off / toggle off	Yes
`pick`	Pick up an object	Yes
`place`	Place a held object at target	Yes
`drop`	Drop the currently held object	No

B.3 Report Statuses

Table 7: **Terminal report statuses** accepted by `report`.

Status	Semantics
`success`	Agent claims the task goal has been achieved
`fail`	Agent concedes it cannot complete the task
`unsafe`	Agent flags a safety concern
`invalid`	Agent flags the task as invalid or unsolvable
`on`	State-recognition report: observed state is *on*
`off`	State-recognition report: observed state is *off*
`open`	State-recognition report: observed state is *open*
`closed`	State-recognition report: observed state is *closed*

In the current frozen pack, `unsafe` and `invalid` are interface-level fallback statuses, not task-specific target labels.

B.4 Step Budgets and Termination

Each task family enforces a fixed step budget and an invalid-action limit (see Table 9 in Appendix D for per-family values). Exceeding the step budget terminates the episode as `fail_no_report`; exceeding the invalid-action limit terminates as `invalid_action_limit_exceeded`.

B.5 Termination Contract

The `report` skill is the *sole* agent-initiated termination mechanism. The agent must invoke it with a status and justification to end the episode; the benchmark never issues a report automatically. Three terminal conditions exist:

1. **Agent report:** the agent invokes `report`; the status is matched against hidden world state.
2. **Budget exhaustion:** the step budget is reached without a report; scored as `fail_no_report`.
3. **Invalid-action limit:** the combined count of protocol failures and invalid executions exceeds the family-specific limit; scored as `invalid_action_limit_exceeded`.

Invalid executions include non-recoverable simulator failures and invalid constraint violations; IL is not a pure formatting-error category.

C Scoring Details

C.1 Dual-Metric Evaluation

Each episode is evaluated under two success metrics simultaneously:

- **Semantic** (primary): tolerant of minor imprecision in object placement or state matching.
- **Strict**: exact simulator predicate check with no tolerance.

The main paper reports aggregate **W**, **B**, and $\Delta = W - B$ under the semantic world predicate paired with strict terminal `report` matching (see §3.3). The strict world metric is retained as a validation variant to expose whether apparent world-state completion would rest on evaluation leniency. Both use simulator state and evaluator-maintained episode records. PG retains successful grounding hits and VS latches visibility. Object-state tasks with specified interactable targets additionally require a pixel interaction resolved to the success target, with simulator success or an `already_in_target_state` response. Without this record, $W = 0$; a physical state change is not required if the target is already in the requested state. This gate applies to the object-state AI, SI, and CR tasks in the frozen pack.

C.2 Reported Closure-Failure Labels

The main analysis reports three closure-failure labels, summarized in Table 8. They are computed post-hoc from the episode settlement record and are used consistently in Figure 3 and Table 3.

Table 8: **Reported closure-failure labels.** These are the diagnostic labels reported in the main figures and tables. FR and NR separate incorrect terminal content from missing terminal commitment; IL captures forced termination due to repeated invalid actions.

Label	Definition
FR	False report: the agent submits a `report` whose status does not match the hidden world state or expected state label. This includes premature success claims and incorrect categorical state reports.
NR	No report: the step budget expires without any `report`. When $W = 1$, this is the post-attainment drift case analyzed in §4.2.
IL	Invalid-action-limit termination: protocol failures and invalid executions together exceed the family-specific limit, forcing episode termination.

These labels distinguish observed closure outcomes, not unique causes. IL captures limit-triggered invalid actions; FR and NR can also reflect interface mismatch, state tracking, perception, or termination behavior. A syntactically valid but wrong categorical label is counted as FR rather than IL.

C.3 Deterministic Report-Matching Rules

The function $\texttt{match}(\texttt{report}_T, s_T)$ referenced in §3.3 operates in two modes, determined by the episode's success-condition type. No family-specific branching exists in the match logic itself; family-specific behavior arises only from how the world-success predicate $\mathcal{G}_{\text{sem}}$ is evaluated.

Goal-completion mode (PG, DA, VS, AI, SI, SM, CR). The episode success condition checks a world-state predicate (e.g., `agent_near_target`, `object_state`, `object_held`). The report status is first normalized to one of eight canonical values (`success`, `fail`, `unsafe`, `invalid`, `on`, `off`, `open`, `closed`); any other value is noncanonical and cannot match. This differs from the canonical negative label `invalid`. The match rule is:

$$\texttt{match} = \big(\texttt{status} = \texttt{success} \ \wedge \ W{=}1\big) \ \vee \ \big(\texttt{status} \in \{\texttt{fail}, \texttt{unsafe}, \texttt{invalid}\} \ \wedge \ W{=}0\big).$$

A correct `fail` report when $W = 0$ therefore counts as a match ("honest fail"), and a `success` report when $W = 0$ is a false-success report. Under this goal-completion rule, `unsafe` and `invalid` are handled as negative reports in the same matching set as `fail`.

State-verification mode (SV). SV episodes use a `report_status` success condition with an explicit expected label derived from the hidden simulator state via:

$$\texttt{is_toggled} \mapsto \texttt{on}/\texttt{off}, \qquad \texttt{is_open} \mapsto \texttt{open}/\texttt{closed}.$$

For benchmark success, the queried predicate must hold and the target must be publicly observable at closure ($W = 1$), and the report must match the expected label. When the target is visible but the queried predicate is false, the expected label is its categorical complement; when public evidence is unavailable, the implementation uses `fail`. A correct negative or complementary report can therefore match while $B = 0$. If the state is not publicly observable at closure, the episode is not a benchmark success regardless of report content. The match rule is exact string equality between the normalized report status and the expected label:

$$\texttt{match} = \big(\texttt{status} = \texttt{expected_label}\big).$$

A stop-only protocol cannot represent this judgment because the correct terminal output is categorical, not binary.

D Task Family Specifications

The eight task families are organized into a **diagnostic tier** (PG, DA, VS, SV) that isolates atomic competencies and a **compositional tier** (AI, SI, SM, CR) that chains them under increasing partial-observability demand. Table 9 summarizes the key parameters; detailed descriptions follow.

Table 9: **Task family specification summary.** "Vis." indicates whether the target is visible at episode start. "Steps" and "Inv." are the episode step budget and invalid-action limit. "Skills" lists the expected skill repertoire; exact initial poses and distances vary by authored episode and are simulator-validated during pack construction.

Family	Layer	Vis.	Steps	Inv.	Expected Skills
PG Pixel Grounding	Diagnostic	✓	5	3	`interact_pixel`(ground), `report`
DA Distance Approach	Diagnostic	✓	12	4	`navigate`, `report`
VS View Search	Diagnostic	×	20	6	`navigate`, `look`, `report`
SV State Verification	Diagnostic	✓	5	2	`look`, `report`
AI Approach and Interact	Compositional	✓	25	8	`navigate`, `interact_pixel`, `report`
SI Search and Interact	Compositional	×	35	10	`navigate`, `look`, `interact_pixel`, `report`
SM Sequential Manipulation	Compositional	✓	30	10	`navigate`, `interact_pixel`, `report`
CR Constraint Resolving	Compositional	✓	40	12	`navigate`, `interact_pixel`, `report`

PG: Pixel Grounding. The target object is visible and reachable at episode start; the agent must click (ground) it via `interact_pixel`(ground), then submit a `report`. This isolates visual grounding—the ability to map a natural-language reference to a pixel coordinate in the egocentric frame. Success criterion: `target_grounded` (the grounding click falls on the correct object instance). Difficulty varies by target object size: large (e.g., fridge, television), medium (e.g., microwave, cabinet), or small (e.g., egg, pen, watch).

DA: Distance Approach. The target is visible but initially outside the success threshold; the agent must navigate close enough, then report. This isolates spatial navigation under egocentric observation—coarse depth estimation and obstacle avoidance without any interaction. Success criterion: `agent_near_target` (Euclidean distance ≤ 1.5 m).

VS: View Search. The target is *not visible* at episode start. The agent must change viewpoint through `navigate` and/or `look` actions until the target becomes visible, then report. This isolates active search under partial observability—the ability to systematically explore the environment. Success criterion: `object_state`(visible=True), latched once achieved.

SV: State Verification. The agent starts with a visible object whose state (on/off or open/closed) must be identified and reported. This probes categorical state reporting under the prompted schema; errors can reflect perception, language, field placement, or termination. Success criterion: `report_status` matching the oracle state label.

AI: Approach and Interact. The target is visible at episode start. The agent must approach as needed, perform a single pixel-grounded interaction (activate, deactivate, open, close, or pick), then report. This composes DA (approach) and PG-level grounding with a state-changing interaction. Success criterion: `object_state` with the target-interaction gate in Appendix C, or `object_held`, depending on intent.

SI: Search and Interact. The target is not visible at episode start. The agent must find the target, approach as needed, perform the required pixel-grounded interaction, and report. This is the full perception–navigation–interaction pipeline under maximal partial observability. Success criterion: same as AI; the initial lack of visibility motivates search rather than adding a separate scoring condition.

SM: Sequential Manipulation. The instruction requires an ordered manipulation chain, such as opening a container, picking an object, and placing it at a destination. Four functional templates exist: `reveal_pick` (open container, pick hidden object), `put_into` (place object in receptacle), `rearrange` (move object between receptacles), and `open_pick_place` (open, pick, place elsewhere). Success criterion: `object_held` or `object_at_receptacle`.

CR: Constraint Resolving. The task setup specifies an obstructed direct path to a visible target. The agent must detour as needed, perform the target interaction, and report. Success criterion: the same target-state or holding predicate as AI, including the applicable target-interaction gate. Path obstruction motivates the task; obstacle removal or constraint resolution is not independently scored.

E Benchmark Episode Details

E.1 Episode Generation Pipeline

Episodes are constructed through a three-stage pipeline:

1. **LLM proposal**: a language model drafts candidate tasks conditioned on scene inventories and family-specific constraints (target visibility, start-pose requirements, available intents, object categories).
2. **Simulator checks**: candidate episodes are instantiated in AI2-THOR to check object existence, state accessibility, agent reachability, and success conditions. Contract checks cover initialization, scene setup, success-spec types, and family-specific constraints. These

checks did not establish valid initialization for every frozen task: subsequent checks identified ten unsupported Dresser SV setup definitions. Appendix M.1 discloses the defects and their exclusion sensitivity.

3. **Human review**: a human auditor reviews a stratified sample for ambiguity, instruction quality, and difficulty calibration. Manually approved episodes receive priority in pack assembly.

E.2 Pack Composition

The evaluation uses pack `mixed_mainline_manual_first_balanced_1000_v1`, containing 1,000 episodes with exactly 125 per task family. These episodes are selected from a larger authored pool spanning diverse AI2-THOR environments, including ProcTHOR-generated scenes.

Table 10: **Episode distribution** in the 1,000-episode evaluation pack.

Family	Episodes	Max Steps
PG Pixel Grounding	125	5
DA Distance Approach	125	12
VS View Search	125	20
SV State Verification	125	5
AI Approach and Interact	125	25
SI Search and Interact	125	35
SM Sequential Manipulation	125	30
CR Constraint Resolving	125	40
Total	**1,000**	

E.3 Simulator Configuration

All episodes run in AI2-THOR using iTHOR and ProcTHOR scenes with fixed rendering parameters: resolution 640×480, field of view 90°, visibility distance $6.0\,\text{m}$, interaction distance limit $1.5\,\text{m}$, and instance segmentation enabled for grounding evaluation. Segmentation is used only by the evaluator for offline grounding and settlement; it is never exposed to the agent, which receives only RGB frames under the default contract. The agent's initial position, rotation, and camera horizon are specified per episode and validated against the scene setup.

F Primary 20-Model Comparison Panel

Table 11 reports per-family **W/B** for the primary 20-model panel used for cross-model comparison. The panel includes models that reliably operate the native-control interface and are not dominated by invalid-action-limit failures or zero benchmark success. The main text (Table 2) uses 10 anchor models; this table adds 10 additional open-weight models spanning MoE variants, smaller checkpoints, and embodied-tuned systems, including Kimi-VL [19], RoboBrain2.5-4B [14], MiMo-Embodied-7B [12], rynnbrain-8B [7], the Qwen3-VL and Qwen3.x families [2, 35, 26, 27], InternVL3.5 [33], and closed-source APIs Gemini 3.1 Pro [10], Doubao-Seed-1.8 [11], GPT-5.4 [23], and Claude-Sonnet-4 [1]. In that 10-anchor set, Qwen3-VL-32B Instruct and Qwen3-VL-32B `(T)` Thinking are counted as two distinct anchor models (`(T)` = public Thinking serving mode); Qwen3.6-27B and Qwen3.5-27B are served with thinking enabled (one anchor row each); every other checkpoint appears once. Suffixes such as `A3B`/`A17B`/`A22B` denote *activated* expert parameters in MoE-style checkpoints.

RoboBrain2.5-4B [14] has SV $W = 94.4\%$ and exact $B = 0.0\%$: 124/125 reports use `status=success`; the remaining `open` report is incorrect. Many summaries nevertheless contain a matching explicit state. Appendix M reports the auxiliary summary-state audit alongside exact scores; neither W nor the exact zero should be interpreted as a recognition score.

Table 11: **Primary 20-model comparison panel under native control.** Each cell reports **W/B** (%). Diagnostic: PG = pixel grounding, DA = distance approach, VS = view search, SV = state verification. Compositional: AI = approach-and-interact, SI = search-and-interact, SM = sequential manipulation, CR = constraint resolving. Models are grouped by category.

Model	All	Diagnostic Probes				Compositional Tasks			
	W/B	PG	DA	VS	SV	AI	SI	SM	CR
Embodied / robotics-tuned									
MiMo-Embodied-7B	29.5/21.3	52.0/34.4	37.6/24.0	31.2/26.4	95.2/74.4	14.4/7.2	0.8/0.8	3.2/1.6	1.6/1.6
RoboBrain2.5-4B	27.4/14.6	78.4/78.4	19.2/14.4	13.6/12.0	94.4/0.0	8.0/8.0	0.0/0.0	4.0/2.4	1.6/1.6
rynnbrain-8B	24.1/17.3	17.6/11.2	50.4/38.4	4.8/1.6	90.4/60.0	20.8/18.4	0.8/0.8	6.4/6.4	1.6/1.6
Open general and reasoning VLMs									
Kimi-VL-A3B (T)	17.3/8.8	10.4/9.6	26.4/3.2	3.2/2.4	95.2/53.6	0.8/0.8	0.0/0.0	1.6/0.0	0.8/0.8
Qwen3-VL-4B	30.7/25.6	59.2/56.8	48.0/42.4	23.2/18.4	74.4/47.2	29.6/28.8	0.8/0.8	8.0/8.0	2.4/2.4
Qwen3-VL-8B	26.8/25.3	53.6/52.8	31.2/30.4	23.2/19.2	94.4/88.0	8.8/8.8	0.0/0.0	1.6/1.6	1.6/1.6
Qwen3-VL-30B-A3B (T)	23.2/20.4	64.8/62.4	8.0/8.0	16.0/15.2	80.8/65.6	8.8/5.6	0.0/0.0	4.8/4.0	2.4/2.4
Qwen3-VL-32B	39.2/18.9	68.8/20.0	44.0/14.4	65.6/48.8	82.4/51.2	34.4/8.0	5.6/0.8	7.2/5.6	5.6/2.4
Qwen3-VL-32B (T)	30.7/26.4	70.4/68.0	10.4/10.4	43.2/41.6	93.6/81.6	15.2/4.8	2.4/0.8	5.6/3.2	4.8/0.8
Qwen3-VL-235B-A22B (T)	35.2/30.5	80.8/78.4	4.8/4.8	63.2/60.0	88.8/68.8	23.2/15.2	3.2/2.4	12.8/10.4	4.8/4.0
Qwen3.5-27B	42.0/36.2	84.0/80.8	41.6/39.2	48.0/44.8	93.6/85.6	36.8/19.2	15.2/8.0	10.4/9.6	6.4/2.4
Qwen3.5-397B-A17B (T)	42.9/33.8	78.4/70.4	42.4/41.6	56.8/52.0	91.2/75.2	49.6/19.2	12.0/5.6	5.6/2.4	7.2/4.0
Qwen3.6-27B	46.7/37.8	87.2/76.8	56.8/55.2	47.2/44.0	93.6/80.8	54.4/28.8	12.8/7.2	14.4/6.4	7.2/3.2
Qwen3.6-35B-A3B (T)	29.9/20.6	53.6/42.4	12.8/12.8	36.0/34.4	92.8/52.8	29.6/15.2	5.6/2.4	4.0/1.6	4.8/3.2
InternVL3.5-30B-A3B	32.3/18.8	38.4/32.0	57.6/15.2	56.0/37.6	89.6/61.6	8.0/0.0	4.8/2.4	2.4/0.8	1.6/0.8
InternVL3.5-38B	39.9/17.4	79.2/16.0	57.6/18.4	50.4/36.8	86.4/55.2	31.2/7.2	7.2/0.0	5.6/4.8	1.6/0.8
Closed-source frontier									
Gemini-3.1-Pro†	57.7/56.4	92.0/92.0	52.8/52.0	71.2/71.2	95.2/90.4	62.4/60.8	26.4/25.6	43.2/42.4	18.4/16.8
Doubao-Seed-1.8†	40.4/37.1	76.8/75.2	44.8/41.6	29.6/29.6	94.4/80.0	42.4/37.6	5.6/4.8	19.2/17.6	10.4/10.4
GPT-5.4†	29.2/24.7	30.4/30.4	24.8/24.8	64.8/62.4	92.8/72.0	10.4/0.8	4.0/1.6	4.8/4.0	1.6/1.6
Claude-Sonnet-4†	25.2/21.1	38.4/35.2	17.6/16.8	65.6/64.8	73.6/45.6	4.8/4.8	0.0/0.0	1.6/1.6	0.0/0.0

† Closed-source API models.

Thinking vs. Instruct serving modes. Several checkpoints offer both Instruct and public "Thinking" serving modes. On the frozen evaluation pack, Qwen3-VL-32B (T) [2] achieves higher aggregate **B** (26.4 vs. 18.9) with a substantially smaller Δ (4.3 vs. 20.3 pp) compared to its Instruct counterpart, shifting the profile from NR-heavy toward a more balanced mixture. We treat this as a behavioral observation rather than a mechanistic claim: serving mode changes the execution/reporting tradeoff under a fixed contract, but the current evidence does not identify which internal reasoning differences cause the shift. For the remaining anchor checkpoints, we do not add further Instruct/Thinking pairing ablations beyond the Qwen3-VL-32B [2] split above—each appears under one fixed deployed configuration.

G Report-Policy Baseline Details

Table 12 reports counterfactual benchmark success when a report is substituted or appended at the recorded final state, leaving the executed trajectory and closure time unchanged. This baseline is not a model-visible oracle experiment: hidden state is used only offline to rescore what would have happened under a different terminal report.

Policies. **Actual** is the observed benchmark success rate. **Always-success** appends or substitutes `report(status=success)` at the final state and recomputes whether that report matches the expected terminal label. For trajectories that originally ended by no-report exhaustion, this should be read as a forced final report at the exhausted state, not as evidence that the model would have chosen the correct stopping time. This baseline can succeed on goal-completion tasks only when the final world predicate is satisfied, but it fails state-verification episodes whose correct terminal labels are `open`/`closed` or `on`/`off`. **Random-report** reports the chance-level expectation under a uniform draw over the two admissible labels for each episode: {`success`, `fail`} for goal-completion tasks, and {`open`, `closed`} or {`on`, `off`} for state-verification tasks. Equivalently, it yields $0.5W$ in expectation on each fixed final state. **Oracle-report** uses the evaluator's correct terminal label at the same

Table 12: **Counterfactual report policies (10 anchor models). B** = actual benchmark success (%). **Always = B** when a success report is substituted or appended at closure. **Rand. = B** under uniform-random report. Bold Always entries exceed actual **B**.

Model	B	Always	Rand.
Gemini-3.1-Pro†	**56.4**	45.8	28.8
Doubao-Seed-1.8†	37.1	28.6	20.2
Qwen3.6-27B	37.8	35.0	23.4
Qwen3.5-27B	36.2	30.3	21.0
Qwen3-VL-32B (T)	26.4	19.0	15.3
GPT-5.4†	24.7	17.6	14.6
Claude-Sonnet-4†	21.1	16.0	12.6
MiMo-Embodied-7B	21.3	17.6	14.8
Qwen3-VL-32B	18.9	**28.9**	19.6
InternVL3.5-38B	17.4	**29.1**	20.0

final state; therefore its **B** equals **W**. It is a fixed-trajectory upper-bound sanity check rather than a model-side report-format baseline.

The always-success policy exceeds actual **B** only for Qwen3-VL-32B [2] (+10.0 pp) and InternVL3.5-38B [33] (+11.7 pp). It recovers terminally satisfied achievement episodes with missing or incorrect reports, but removes every SV success because `success` is not an admissible state label. The net change combines these opposing effects and does not isolate a reporting failure mode. Random-report is a chance reference, not a lower bound on actual performance: its expectation is $0.5W$ and can exceed actual **B**.

Implementation. The counterfactual analysis uses the same saved 1,000-episode native-control traces as the corresponding rows in the main model panel.

H Belief Lag and Premature Commitment

Table 13 provides step-level detail on the 10 anchor models' terminal-commitment timing. The main-text analysis (§4.2) reports (i) model-wise mean lags of 0.9–1.9 steps to a correct report and (ii) 65–88% of false-success reports with zero recorded evaluator progress. Lag starts at the first recorded post-action step with $W = 1$, excluding the initial state at $t = 0$, and ends at the correct report. It conditions on terminal $W = 1$ and a correct report across all eight families; an SV episode reporting correctly on its first step contributes zero lag. The means therefore depend on task composition. Progress lies in $[0, 1]$ but is predicate-dependent: state, holding, and placement conditions use binary values, while distance-approach progress depends on absolute target distance rather than improvement from the initial state. The recorded field retains the maximum progress observed so far. A zero value therefore need not imply that no navigation or intermediate task-relevant action occurred, and scores are not a common measure of partial achievement across families.

H.1 Conditional Report Rates

Table 14 reports conditional report rates for the 10 anchor models. **P(rep | W=0)** is the probability that the model issues a report when the world condition is *not* satisfied; high values indicate a stronger tendency to terminate under world failure, which reflects indiscriminate reporting when paired with high FR. **P(¬rep | W=1)** is the probability that the model fails to report when the world condition *is* satisfied; high values indicate missed closure. Note that these conditional *rates* are recoverable from stop timing alone—they measure whether the agent terminated, not what it said. Report *content* becomes essential for a different reason: determining whether the agent's stated judgment matches the hidden world state (e.g., `open` vs. `closed` for state-verification tasks), which is the basis for the FR/NR decomposition (Figure 3) and the counterfactual analysis in Table 12.

FR-heavy models (Claude-Sonnet-4 [1], GPT-5.4 [23]) report in >85% of W=0 episodes, whereas NR-heavy models (InternVL3.5-38B [33], Qwen3-VL-32B [2]) miss closure in >40% of W=1 episodes. These opposite conditional profiles overlap in aggregate scalar success (B ≈ 17–25%).

Table 13: **Closure lag and premature commitment (10 anchor models).** **W+** = episodes with world-state completion (count out of 1,000). **Lag** = mean steps from the first recorded post-action $W = 1$ to the correct terminal report, conditional on terminal $W = 1$ and a correct report; all families, including SV, are pooled. M_+ = $W = 1$ episodes without a report, regardless of closure cause (count). For false-success reports: count (**FS**), % issued at zero evaluator progress (**@0**).

	Goal reached			False success	
Model	**W+**	**Lag** ↓	M_+	**FS**	**@0 %**
Gemini-3.1-Pro†	577	1.0	4	225	80
Doubao-Seed-1.8†	404	1.2	17	353	88
Qwen3.6-27B	467	1.1	72	165	75
Qwen3.5-27B	420	1.2	45	219	77
Qwen3-VL-32B (T)	307	1.0	21	260	65
GPT-5.4†	292	0.9	4	302	72
Claude-Sonnet-4†	252	1.9	15	650	88
MiMo-Embodied-7B	295	1.1	56	198	81
Qwen3-VL-32B	392	1.9	166	153	86
InternVL3.5-38B	399	1.4	201	49	84

Table 14: **Conditional report rates (10 anchor models).** C_R = fraction of episodes where a `report` is issued. **P(rep | W=0)** = report rate conditioned on world failure. **P(¬rep | W=1)** = missed-closure rate conditioned on world success. **W=0** and **W=1** columns show the conditioning set sizes.

			Conditional rates (%)			Conditioning	
Model	**W** ↑	**B** ↑	C_R	**P(rep | W=0)** ↓	**P(¬rep | W=1)** ↓	**W=0**	**W=1**
Gemini-3.1-Pro†	57.7	56.4	81.6	57.4	0.7	423	577
Doubao-Seed-1.8†	40.4	37.1	75.1	61.1	4.2	596	404
Qwen3.6-27B	46.7	37.8	56.9	32.6	15.4	533	467
Qwen3.5-27B	42.0	36.2	62.3	42.8	10.7	580	420
Qwen3-VL-32B (T)	30.7	26.4	66.7	55.0	6.8	693	307
GPT-5.4†	29.2	24.7	89.5	85.7	1.4	708	292
Claude-Sonnet-4†	25.2	21.1	92.3	91.7	6.0	748	252
MiMo-Embodied-7B	29.5	21.3	44.9	29.8	19.0	705	295
Qwen3-VL-32B	39.2	18.9	48.2	42.1	42.3	608	392
InternVL3.5-38B	39.9	17.4	28.3	14.1	50.4	601	399

I Action-Feedback Intervention Details

The minimal action-feedback intervention (§4.3, Table 3) adds two boolean signals—`too_far` (interaction attempted beyond the 1.5 m proximity threshold) and `path_blocked` (navigation blocked by an obstacle)—after each action, alongside the standard RGB frame and dialogue history. These are limited execution-outcome cues inspired by controller feedback, not a full proprioceptive or controller-state interface. The `retry` columns in Table 3 report mean per-episode counts of redundant `interact_pixel` steps that repeat an identical invocation (every repeat after the first occurrence of each invocation signature increments the episode total). The intervention is outside the baseline contract. An episode-level recount corrects five +FB retry means in Table 3: Qwen3.5-27B, 0.16 to 0.37; Qwen3-VL-32B, 0.42 to 2.28; Qwen3-VL-32B (T), 0.41 to 1.48; InternVL3.5-38B, 0.01 to 1.19; MiMo-Embodied-7B, 0.39 to 2.46. These corrections concern retry aggregation. Complete per-episode traces give 371, 2,278, 1,484, 1,189, and 2,461 repeated interactions over 1,000 episodes, respectively. The retry definition and all $W/B/\mathrm{FR}/\mathrm{NR}$ changes are unchanged; API retries are excluded.

J Robustness and Reproducibility Checks

This section collects checks that support the stability of the reported closure profiles without expanding the primary comparison panel: a neutral-report prompt variant and paired reruns over the same 20 open-weight configurations for which paired robustness runs were available.

J.1 Run Metadata and Determinism

Prompt policy. The primary comparison uses the baseline prompt in Appendix A, including its two instructions against reporting completion from visual plausibility alone. State-verification tasks request an observed state label rather than a generic success/failure judgment. The action schema, grounding rules, and output constraints are shared across task families; only the task instruction and model-specific coordinate convention vary. The neutral-wording condition is described separately below.

Frozen pack and run randomness. All models are evaluated on the same frozen 1,000-episode selection; episode IDs and success specifications are fixed. Any "seed" in run logs affects only job scheduling or tie-breaking in the inference stack, not which episodes enter the aggregate.

Evaluation profile. The default profile is `pure_rgb_dialogue_history_baseline`: single-frame egocentric RGB, 20-turn text-only dialogue history, no depth, no pose, no visual history frames, and no agent-side memory.

Infrastructure. Open models were served through vLLM and closed models through their listed APIs. Models use pixel or normalized image coordinates as specified in Appendix A. We evaluate the existing models without benchmark-specific training. Episode order within each pack run is deterministic (sorted by episode ID). Specific closed-source model identifiers are listed in Table 2 footnotes.

J.2 Prompt Sensitivity Analysis

The default system prompt includes two anti-hallucination instructions in the task block (Appendix A): *"Mere visibility, proximity, or an attempted action is not enough; do not report success from visual plausibility alone"* and *"Do not claim completion from visual plausibility alone when the task requires changing world state."* To test whether closure-failure profiles are artifacts of these instructions, we run a **neutral-report** prompt variant that replaces both lines with neutral reporting guidance while keeping the rest of the evaluation contract identical: same frozen episode pack, same profile (`pure_rgb_dialogue_history_baseline`), same scoring rules. Table 15 reports per-family **W/B** for all 20 open-weight configurations for which paired neutral-prompt runs were available. Closed-source models are omitted because paired neutral-prompt runs were not available. The replacement sentences are: "Decide when to report based on the task instruction, the current observation, and the action history available in this episode" and "For tasks that require changing world state, use the observations available after your actions when deciding whether the requested condition has been achieved." Thus this intervention changes the wording of the two instructions rather than simply deleting them.

Across all 20 configurations, aggregate **W** shifts by at most 3.5 pp and aggregate **B** by at most 4.8 pp relative to the corresponding default-prompt runs. NR-heavy models remain NR-heavy (InternVL3.5-38B [33]: 68.2% NR vs. 66.1% under the default prompt) and low-Δ models remain low-Δ (Qwen3-VL-8B [2]: $\Delta = 1.3$ vs. 1.5 pp). For the paired open-weight configurations, the main closure-failure profiles are therefore unlikely to be artifacts of the specific anti-hallucination instructions in the default prompt.

J.3 Repeated-Run Robustness for Open-Weight Models

To check whether the open-weight conclusions depend on a single serving run, we repeat the default no-feedback evaluation for the same 20 open-weight configurations used in the prompt-sensitivity check on the same frozen 1,000-episode pack and the same `pure_rgb_dialogue_history_baseline` profile. Table 16 reports the mean and half-range

Table 15: **Prompt sensitivity: neutral-report variant (20 open-weight configurations).** Same evaluation contract as the main analysis except that both anti-hallucination instructions are replaced by neutral reporting guidance. Each cell reports **W/B** (%). Diagnostic: PG = pixel grounding, DA = distance approach, VS = view search, SV = state verification. Compositional: AI = approach-and-interact, SI = search-and-interact, SM = sequential manipulation, CR = constraint resolving. Configurations are grouped by model family.

Model	All	Diagnostic Probes				Compositional Tasks			
	W/B	PG	DA	VS	SV	AI	SI	SM	CR
Embodied / robotics-tuned									
MiMo-Embodied-7B	28.1/23.1	53.6/50.4	28.8/24.0	33.6/29.6	95.2/73.6	8.0/3.2	0.0/0.0	4.0/2.4	1.6/1.6
RoboBrain2.5-4B	27.6/15.3	76.0/76.0	22.4/20.8	11.2/9.6	94.4/0.0	11.2/11.2	0.0/0.0	4.8/4.0	0.8/0.8
rynnbrain-8B	22.8/17.3	26.4/18.4	30.4/23.2	4.8/3.2	92.0/68.8	21.6/17.6	0.8/0.8	5.6/5.6	0.8/0.8
Open general and reasoning VLMs									
Kimi-VL-A3B (T)	16.8/11.2	12.0/12.0	21.6/8.0	4.0/3.2	94.4/64.8	0.8/0.8	0.0/0.0	1.6/0.8	0.0/0.0
Kimi-VL-A3B-Instruct	18.4/0.0	0.0/0.0	49.6/0.0	1.6/0.0	94.4/0.0	0.0/0.0	0.0/0.0	1.6/0.0	0.0/0.0
Qwen3-VL-4B	32.3/22.5	62.4/57.6	48.0/40.0	28.0/20.0	76.8/20.0	27.2/26.4	1.6/1.6	10.4/10.4	4.0/4.0
Qwen3-VL-4B (T)	24.9/15.5	45.6/45.6	12.8/12.0	37.6/33.6	95.2/26.4	2.4/2.4	0.0/0.0	5.6/4.0	0.0/0.0
Qwen3-VL-8B	27.0/25.7	56.8/56.0	28.0/27.2	24.8/22.4	95.2/88.8	8.0/8.0	0.0/0.0	2.4/2.4	0.8/0.8
Qwen3-VL-8B (T)	22.8/20.0	38.4/35.2	13.6/12.8	29.6/28.8	95.2/80.0	3.2/1.6	0.0/0.0	1.6/0.8	0.8/0.8
Qwen3-VL-30B-A3B (T)	23.0/20.7	58.4/56.8	8.0/8.0	22.4/22.4	79.2/66.4	11.2/8.0	0.0/0.0	4.0/3.2	0.8/0.8
Qwen3-VL-32B	39.1/23.7	66.4/24.0	38.4/18.4	66.4/54.4	86.4/71.2	35.2/13.6	8.0/0.8	6.4/4.8	5.6/2.4
Qwen3-VL-32B (T)	27.2/24.1	68.0/66.4	4.0/4.0	37.6/32.0	93.6/83.2	8.0/4.0	0.0/0.0	3.2/1.6	3.2/1.6
Qwen3.5-2B	17.7/7.4	0.8/0.8	42.4/0.8	1.6/0.0	95.2/57.6	0.0/0.0	0.0/0.0	1.6/0.0	0.0/0.0
Qwen3.5-9B	32.2/24.0	58.4/29.6	52.8/44.8	28.8/26.4	94.4/82.4	15.2/5.6	0.0/0.0	4.0/1.6	4.0/1.6
Qwen3.5-27B	40.7/36.3	77.6/76.0	37.6/37.6	47.2/44.8	95.2/88.8	39.2/24.0	9.6/4.8	12.0/10.4	7.2/4.0
Qwen3-VL-2B	12.7/0.0	0.0/0.0	1.6/0.0	3.2/0.0	95.2/0.0	0.0/0.0	0.0/0.0	1.6/0.0	0.0/0.0
Qwen3-VL-2B (T)	14.2/6.7	0.8/0.8	6.4/3.2	8.8/8.0	95.2/41.6	0.8/0.0	0.0/0.0	1.6/0.0	0.0/0.0
InternVL3.5-2B	14.6/0.1	0.0/0.0	3.2/0.0	16.8/0.8	94.4/0.0	0.8/0.0	0.0/0.0	1.6/0.0	0.0/0.0
InternVL3.5-30B-A3B	32.4/18.8	39.2/31.2	56.0/20.0	59.2/43.2	86.4/52.0	10.4/0.8	6.4/3.2	1.6/0.0	0.0/0.0
InternVL3.5-38B	41.2/18.9	77.6/16.8	57.6/24.0	55.2/39.2	85.6/57.6	34.4/8.8	12.0/0.0	5.6/4.0	1.6/0.8

across the two runs. This robustness panel is intentionally broader than the primary 20-model comparison panel in some directions and narrower in others: it includes low-compliance and supplementary size/serving-mode variants, while closed-source API models and the largest open-weight runs are kept as single-run evaluations because paired reruns were not available under the same resource budget.

The paired runs preserve the qualitative profile assignments used in the main analysis. Aggregate **B** is highly stable for most configurations: 19 of 20 have a half-range below 1 pp, and the largest half-range is Qwen3-VL-32B (T) [2] at 2.7 pp. The most important closure regimes are also stable: InternVL3.5-38B [33] remains strongly NR-heavy (NR = 66.2$\pm$0.0), Qwen3-VL-32B [2] remains high-Δ and NR-heavy ($\Delta = 19.6 \pm 0.7$, NR = 45.0$\pm$0.4), and low-Δ models such as Qwen3-VL-8B [2] remain low-gap across both runs. The repeated-run panel should therefore be read as a robustness check over open-weight configurations, not as a replacement for the primary 20-model comparison panel.

K Qualitative Trajectory Examples

To complement the aggregate results, we visualize two representative trajectories under the same native-control contract used in the main evaluation. The examples highlight the two most interpretable failure surfaces: a single-frame state-verification judgment and a short approach–interaction episode.

K.1 State Verification: Same Frame, Opposite Reports

The SV episode `procthor_microwave_seed298` asks the agent to observe a microwave and report whether it is *open* or *closed*. Initially, the microwave is closed and publicly observable, so the expected report is `closed`. A report without physical action suffices; moving away can remove the public evidence required at closure.

Table 16: **Repeated-run robustness for 20 open-weight configurations.** Each cell reports mean ± half-range over two identical-contract runs on the same frozen episode pack. **W** = world-state completion, **B** = benchmark success, $\Delta = W - B$, **FR** = false-report rate, and **NR** = no-report rate; all values are percentages.

Model	W	B	Δ	FR	NR
Qwen3.5-27B	42.6±0.6	36.6±0.4	6.1±0.2	25.8±0.3	31.1±1.0
Qwen3-VL-4B	30.9±0.2	25.8±0.2	5.1±0.1	51.2±0.9	16.1±0.4
Qwen3-VL-8B	26.9±0.1	25.3±0.0	1.6±0.1	63.1±0.1	5.2±0.2
Qwen3-VL-32B (T)	28.8±1.9	23.8±2.7	5.1±0.8	42.0±2.6	20.0±0.8
Qwen3.5-9B	33.4±0.5	23.3±0.1	10.1±0.4	16.6±0.0	31.6±0.1
MiMo-Embodied-7B	29.4±0.2	21.2±0.1	8.2±0.0	22.8±0.1	42.5±0.6
Qwen3-VL-30B-A3B (T)	23.0±0.3	20.1±0.3	2.9±0.0	57.2±0.4	8.2±0.5
Qwen3-VL-32B	39.0±0.2	19.4±0.5	19.6±0.7	25.8±0.1	45.0±0.4
Qwen3-VL-8B (T)	22.0±0.5	18.8±0.7	3.2±0.2	52.0±0.2	11.8±1.0
InternVL3.5-30B-A3B	31.9±0.4	18.6±0.2	13.4±0.2	16.8±0.9	54.9±0.9
Qwen3-VL-4B (T)	24.1±1.1	17.6±0.7	6.4±0.4	57.4±0.3	8.4±0.4
InternVL3.5-38B	40.0±0.1	17.5±0.2	22.5±0.1	10.0±0.3	66.2±0.0
rynnbrain-8B	23.7±0.4	16.6±0.6	7.1±0.2	25.3±1.1	50.8±0.4
RoboBrain2.5-4B	27.9±0.5	15.1±0.5	12.9±0.1	58.2±0.8	9.9±1.2
Kimi-VL-A3B (T)	17.3±0.1	9.1±0.3	8.3±0.2	38.3±0.4	42.6±0.4
Qwen3.5-2B	17.2±0.2	6.6±0.1	10.6±0.1	2.2±0.4	34.0±0.7
Qwen3-VL-2B (T)	14.9±0.1	3.8±0.4	11.1±0.6	26.7±0.4	1.0±0.2
InternVL3.5-2B	13.6±0.2	0.2±0.1	13.5±0.2	2.7±0.2	30.9±0.4
Kimi-VL-A3B-Instruct	18.5±0.0	0.0±0.0	18.5±0.0	1.0±0.3	78.0±0.2
Qwen3-VL-2B	12.7±0.0	0.0±0.0	12.7±0.0	1.2±0.2	0.1±0.1

K.2 Grounded Interaction: Terminal Report Failures

The AI episode `procthor_fridge_seed314` requires the agent to approach and open a refrigerator. The target is visible from the start, but success requires approaching within interaction range, executing the correct pixel-grounded interaction, and reporting only after the world state changes.

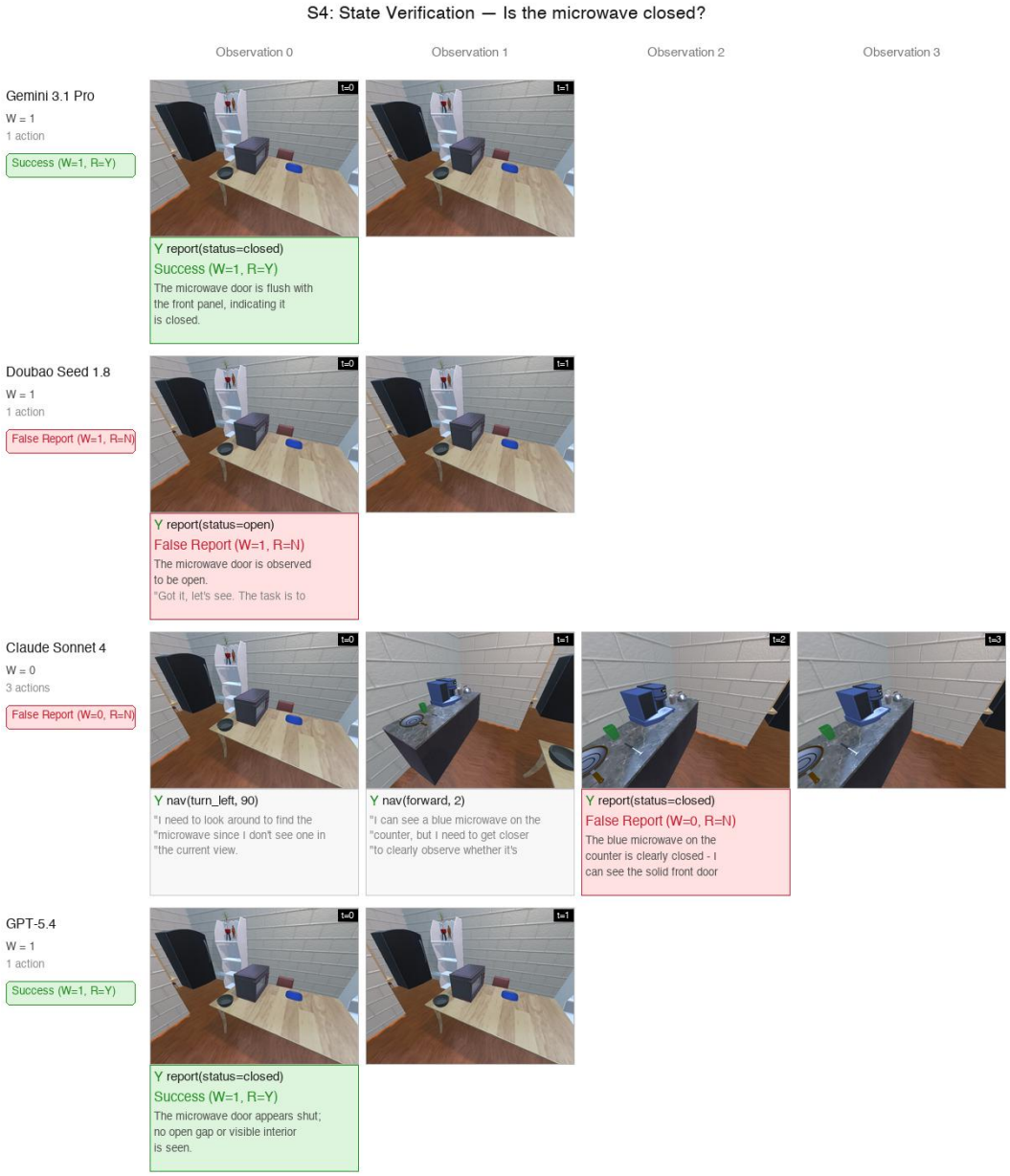


Figure 5: **State-verification trajectory example.** The same initial observation can lead to correct closure (Gemini-3.1-Pro [10] and GPT-5.4 [23]) or a false report (Doubao-Seed-1.8 [11]). Claude-Sonnet-4 [1] reports `closed` after moving away: the physical predicate still holds, but lost public observability makes the expected label `fail`. This is an exact-contract failure, not an opposite-state report. Columns show observation frames, including $t = 0$; annotations give the following action.



Figure 6: **Approach-and-interact trajectory example.** Gemini-3.1-Pro [10] opens the refrigerator and reports success after the state change. Doubao-Seed-1.8 and Claude-Sonnet-4 falsely report success. GPT-5.4 describes the still-closed fridge but uses `closed` where the achievement-task contract requires `fail`: a report-label mismatch rather than an opposite-state claim.

L Any-Time Attainment and Direct Report Correctness

These companion analyses use the four complete 1,000-episode evaluation runs for RoboBrain2.5-4B, Qwen3-VL-8B, InternVL3.5-38B, and Qwen3.6-27B. They retain the original scores and denominators. They are distinct from the fresh matched interface campaign in Appendix N.

L.1 Task-Split Attainment and Loss at Closure

For the 875 achievement episodes, let g_t be the evaluator's primary semantic world predicate at recorded post-action step t. We define any-time attainment $A = \mathbb{I}[\exists t \geq 1 : g_t = 1]$ and loss at closure $L = A(1 - W)$. We report $\sum A/875$, $\sum L/875$, and $\sum L / \sum A$; W remains the final-state metric. PG retains grounding hits and VS latches visibility once achieved; loss is therefore possible only for predicates that can revert. These measures concern recorded evaluator states, not unobserved intermediate simulator frames.

Table 17: **Achievement-task attainment** in the primary evaluation runs. Counts are exact; percentages use 875 episodes except the last column, which conditions on any-time attainment. Independent rounding can make displayed differences non-additive.

Model	$A/W/L$ counts	Any-time (%)	Final W (%)	Lost (%)	Lost/reached (%)
RoboBrain2.5-4B	156/156/0	17.8	17.8	0.0	0.0
Qwen3-VL-8B	151/150/1	17.3	17.1	0.1	0.7
InternVL3.5-38B	293/291/2	33.5	33.3	0.2	0.7
Qwen3.6-27B	354/350/4	40.5	40.0	0.5	1.1

SV is intended to start with the queried state satisfied, so achievement-style any-time success is not informative. Ten historical setup exceptions are analyzed separately in Appendix M.1; Table 18 retains the original denominator. We instead track any post-action $W = 0$, terminal W, and return to $W = 1$ by closure. The SV predicate includes public observability: a loss of W need not mean a physical reversal of the object's state. Table 18 reports evaluator outcomes, not recognition accuracy. Combining achievement and SV terminal counts recovers the primary overall W: 27.4%, 26.8%, 39.9%, and 46.7%.

Table 18: **SV evaluator-state retention**, with all percentages over 125 SV episodes. "Returned" means an observed post-action $W = 0$ followed by terminal $W = 1$.

Model	Terminal $W = 1$	Any post-action $W = 0$	Terminal $W = 0$	Returned
RoboBrain2.5-4B	94.4	5.6	5.6	0.0
Qwen3-VL-8B	94.4	5.6	5.6	0.0
InternVL3.5-38B	86.4	15.2	13.6	1.6
Qwen3.6-27B	93.6	6.4	6.4	0.0

In these four archived configurations, terminal scoring undercounts achievement by only 0.0–0.5 pp, with loss in 0.0–1.1% of reached episodes. This does not establish the same rate for other configurations. InternVL has 201 unreported episodes among 399 terminally satisfied episodes, compared with only two lost goals among 293 reached achievement episodes; missed reporting and reached-then-lost behavior are therefore distinct diagnostics.

L.2 Exact Correctness and Confusion Counts

The report measures in §3.3 include both completed and incomplete tasks. Correct negative reports count toward R but not B. Missing reports count as incorrect in R and are excluded from the conditional denominator. Submitted reports with noncanonical status labels count as incorrect in both rates; the canonical label `invalid` follows the negative-report rule below. The four archives contain no match-unknown reports.

InternVL combines low coverage with higher conditional correctness; RoboBrain reports frequently but has lower exact correctness, including its SV field-placement anomaly. These are protocol scores,

Table 19: **Direct exact-report correctness** on primary evaluation runs, each with 1,000 episodes. Counts are reports/exact-correct; all rates are percentages.

Model	Reports/correct	Coverage C_R	All-episode R	$R_{\|\text{rep}}$
RoboBrain2.5-4B	744/153	74.4	15.3	20.6
Qwen3-VL-8B	946/316	94.6	31.6	33.4
InternVL3.5-38B	283/180	28.3	18.0	63.6
Qwen3.6-27B	569/385	56.9	38.5	67.7

not pure semantic-judgment accuracy. Table 20 gives disjoint achievement-task outcomes, grouping `fail`, `unsafe`, and `invalid` as negative reports according to the existing matching rule. SV categorical confusion counts are kept separate rather than forced into binary success/failure labels. The supplement includes complete expected-label versus submitted-label counts and all no-report episodes; the latter include forced closures beyond NR alone.

Table 20: **Disjoint report outcomes** in the four primary evaluation runs. Achievement rows sum to 875: CS/CF are correct success/failure; FS/FF are false success/failure; O is another report label; M is any missing report. SV rows sum to 125 and retain categorical semantics.

Model	CS	CF	FS	FF	O	M
RoboBrain2.5-4B	146	7	466	0	0	256
Qwen3-VL-8B	143	58	618	2	0	54
InternVL3.5-38B	105	2	49	0	43	676
Qwen3.6-27B	277	2	165	1	0	430

	SV correct report	SV incorrect report	SV no report
RoboBrain2.5-4B	0	125	0
Qwen3-VL-8B	115	10	0
InternVL3.5-38B	73	11	41
Qwen3.6-27B	106	18	1

M Auxiliary Summary-State Audit

The exact machine-readable score remains unchanged. We audit terminal SV summaries using a deterministic, model-independent rule whose inputs are only the summary and queried attribute. The rule requires a bare state label or an explicit present-state construction such as "is open", "appears to be off", or "observed state: closed". It recognizes only the queried pair: `on`/`off` or `open`/`closed`. Generic `success`/`fail` never implies an object state. Summaries containing negation, uncertainty, or conditionals are left unmapped rather than interpreted by polarity inversion. Spatial "on the table", conflicting assertions, past-only descriptions, and imperatives are not mapped. An unavailable categorical terminal label is also unmapped; missing reports are separate. This deliberately narrow rule does not resolve arbitrary object references or provide complete semantic interpretation.

For example, "no signs of being powered on" and "no visible open gap" are left unmapped because they do not assert the corresponding positive state. The supplementary script and examples specify the rule, which was refined during the analysis rather than preregistered. Expected answers are used only after mapping, to measure agreement. The same rule is applied to all 2,500 SV episodes from the primary evaluation.

For RoboBrain, this audit gives 105 agreements, 17 disagreements, and three unmapped cases among 125 reports: 84.0% agreement overall and 86.1% among 122 mapped cases. An earlier lexical audit gave 104/17/4; the difference reflects the refined assertion rule, and the two analyses are not pooled. Its exact SV B remains zero. Many summaries therefore contain explicit matching state content despite a generic `status`, while the audit still finds opposite-state assertions. Table 21 gives the revised audit for all 20 primary configurations alongside exact scores.

Agreement is neither a replacement B nor a perception score; it can coexist with $W = 0$. Unmapped summaries are not evidence of incorrect reasoning. Exact SV B and summary agreement have

different eligibility conditions. Per-episode outputs, rejection reasons, unmapped examples, and the known-setup exclusion flag are included in the supplementary files.

Table 21: **SV exact scores and summary-state audit** with the assertion-only rule for all 20 primary configurations, each with 125 episodes. A/D/U/M are agreeing/disagreeing/unmapped/missing-report counts and sum to 125. S is agreement over all episodes; $S_{|\mathrm{map}}$ conditions on A+D.

Model	Exact W/B (%)	A/D/U/M	S (%)	$S_{\|\mathrm{map}}$ (%)
MiMo-Embodied-7B	95.2/74.4	74/18/33/0	59.2	80.4
RoboBrain2.5-4B	94.4/0.0	105/17/3/0	84.0	86.1
rynnbrain-8B	90.4/60.0	76/17/7/25	60.8	81.7
Kimi-VL-A3B (T)	95.2/53.6	69/19/14/23	55.2	78.4
Qwen3-VL-4B	74.4/47.2	88/18/15/4	70.4	83.0
Qwen3-VL-8B	94.4/88.0	101/5/19/0	80.8	95.3
Qwen3-VL-30B-A3B (T)	80.8/65.6	63/14/44/4	50.4	81.8
Qwen3-VL-32B	82.4/51.2	51/2/27/45	40.8	96.2
Qwen3-VL-32B (T)	93.6/81.6	62/8/50/5	49.6	88.6
Qwen3-VL-235B-A22B (T)	88.8/68.8	57/6/52/10	45.6	90.5
Qwen3.5-27B	93.6/85.6	47/4/73/1	37.6	92.2
Qwen3.5-397B-A17B (T)	91.2/75.2	38/2/73/12	30.4	95.0
Qwen3.6-27B	93.6/80.8	63/8/53/1	50.4	88.7
Qwen3.6-35B-A3B (T)	92.8/52.8	40/6/38/41	32.0	87.0
InternVL3.5-30B-A3B	89.6/61.6	80/26/11/8	64.0	75.5
InternVL3.5-38B	86.4/55.2	72/9/3/41	57.6	88.9
Gemini-3.1-Pro	95.2/90.4	96/5/24/0	76.8	95.0
Doubao-Seed-1.8	94.4/80.0	86/11/26/2	68.8	88.7
GPT-5.4	92.8/72.0	45/7/70/3	36.0	86.5
Claude-Sonnet-4	73.6/45.6	33/8/58/26	26.4	80.5

Fresh paired-campaign check. We separately applied the earlier summary-state rule to fresh baselines, restricted to SV episodes with $W = 1$ and exact $B = 0$: agreement was 102/118 for RoboBrain2.5-4B, 1/8 for Qwen3-VL-8B, 0/37 for InternVL3.5-38B, and 0/9 for Qwen3.6-27B. These counts are distinct from the primary-run assertion audit, use the earlier lexical rule, and are not pooled. Among these four configurations, the large summary/status discrepancy is concentrated in RoboBrain; this does not imply that the interface has no effect on other models.

M.1 Sensitivity to Historical SV Setup Defects

Subsequent simulator initialization checks identified ten historical SV definitions that requested open/close setup on a Dresser target without the required capability. This is a task-setup defect, not model failure. The ten IDs represent 8.0% of the original SV subset; nine retained tasks were repaired as Drawer targets in the revised release, and one was removed from that selection. Historical scored trajectories and the main tables are preserved for comparison with the submission, but these cases must not be treated as validated SV initializations.

Table 22 excludes all ten definitions uniformly across the 20 primary archives, leaving 990 episodes overall and 115 SV episodes per model. The 875 achievement episodes and their any-time analysis are unchanged. Absolute overall W/B shifts are below 0.3 pp; SV rates change more because of the smaller denominator. RoboBrain still has zero exact SV B and 97 agreeing versus 17 disagreeing mapped summaries among the 115 retained episodes. This sensitivity analysis does not certify the remaining initializations or replace fresh evaluation on repaired tasks. The supplement provides exact retained counts, IDs, and the setup-diagnostic provenance.

Table 22: **Sensitivity to ten historical SV setup defects.** Rates are percentages. The original overall denominator is 1,000; excluding ten definitions gives 990 overall and 115 SV episodes. This is exclusion-based analysis of archived trajectories, not corrected simulator reruns. Exact integer counts are in the supplement.

Model	Original overall W/B	Retained overall W/B	Retained SV W/B
MiMo-Embodied-7B	29.5/21.3	29.4/21.1	100.0/77.4
RoboBrain2.5-4B	27.4/14.6	27.3/14.7	99.1/0.0
rynnbrain-8B	24.1/17.3	23.9/17.1	94.8/61.7
Kimi-VL-A3B (T)	17.3/8.8	17.1/8.6	100.0/55.7
Qwen3-VL-4B	30.7/25.6	30.6/25.7	77.4/49.6
Qwen3-VL-8B	26.8/25.3	26.7/25.2	99.1/92.2
Qwen3-VL-30B-A3B (T)	23.2/20.4	23.0/20.4	84.3/69.6
Qwen3-VL-32B	39.2/18.9	39.2/18.9	86.1/53.9
Qwen3-VL-32B (T)	30.7/26.4	30.6/26.3	98.3/85.2
Qwen3-VL-235B-A22B (T)	35.2/30.5	35.2/30.5	93.0/72.2
Qwen3.5-27B	42.0/36.2	42.0/36.2	98.3/89.6
Qwen3.5-397B-A17B (T)	42.9/33.8	42.9/33.7	95.7/78.3
Qwen3.6-27B	46.7/37.8	46.8/37.8	98.3/84.3
Qwen3.6-35B-A3B (T)	29.9/20.6	29.8/20.4	97.4/53.9
InternVL3.5-30B-A3B	32.3/18.8	32.2/18.7	93.9/64.3
InternVL3.5-38B	39.9/17.4	39.9/17.2	90.4/56.5
Gemini-3.1-Pro	57.7/56.4	57.9/56.6	100.0/94.8
Doubao-Seed-1.8	40.4/37.1	40.4/37.1	99.1/83.5
GPT-5.4	29.2/24.7	29.1/24.5	97.4/74.8
Claude-Sonnet-4	25.2/21.1	25.1/21.0	76.5/47.0

N Matched Interface, History, and Memory Controls

We evaluate interface, history, and memory controls using matched task selections. Four open models were selected to span failure profiles: RoboBrain2.5-4B and Qwen3-VL-8B are FR-heavy, InternVL3.5-38B is NR-heavy, and Qwen3.6-27B is intermediate. Interface deltas are treatment minus a fresh matched baseline, not minus the primary Table 11 values. Comparisons are descriptive; they are not significance tests or evidence of a universal causal explanation.

N.1 Neutral Wording and Content-Free Acknowledgement

Neutral wording changes only the two prompt sentences governing when to report. This fresh control is distinct from the broader 20-configuration neutral-wording experiment in Appendix J.2. The acknowledgement condition appends the fixed message `action_received` after every action without changing the system prompt; it contains no action success, state, pose, or error information.

Table 23: **Fresh interface controls** on 1,000 matched episodes per condition. All entries are percentage-point changes; C_R is report coverage and R is all-episode exact-report correctness.

	Neutral wording			`action_received`			
Model	ΔC_R	ΔW	ΔB	ΔC_R	ΔW	ΔB	ΔR
RoboBrain2.5-4B	+4.4	−1.1	−0.7	+4.5	+0.1	+0.4	+0.7
Qwen3-VL-8B	+13.4	−0.5	−1.0	+14.7	+0.7	+0.8	+2.0
InternVL3.5-38B	+2.3	−0.4	+0.9	−2.5	−0.4	−1.5	−1.5
Qwen3.6-27B	+3.4	−1.8	−0.4	+1.0	+1.1	+0.3	+0.3

Neutral wording raises coverage for every tested model while W decreases in all four. Acknowledgement changes coverage by up to 14.7 pp, but its W, B, and R effects are small and mixed. These comparisons do not support the absence of a receipt cue as a common explanation for these four profiles; they do not distinguish uncertainty from state-tracking failure. Appendix J provides the broader checks: neutral wording changes aggregate W by at most 3.5 pp and B by at most 4.8 pp across 20 open-weight configurations; repeated-run B half-range is below 1 pp for 19/20 configurations.

N.2 Payload-Free Stop

The semantic report is replaced by a required payload-free `stop` on the same 875 non-SV episodes. This intervention changes the action name, schema, and prompt together. Stop success requires terminal $W = 1$ and a successful stop; it carries no semantic judgment and is a different endpoint from B.

Table 24: **Stop-only control**, percentages over 875 achievement episodes. The last column compares different endpoints and must not be interpreted as improved report accuracy.

Model	Baseline W/B	Stop W/success	ΔW (pp)	Stop success $-B$ (pp)
RoboBrain2.5-4B	19.3/17.7	18.2/7.3	-1.1	-10.4
Qwen3-VL-8B	15.8/14.7	18.1/16.0	+2.3	+1.3
InternVL3.5-38B	33.6/12.6	31.2/16.7	-2.4	+4.1
Qwen3.6-27B	35.8/26.9	38.2/27.9	+2.4	+1.0

InternVL's stop endpoint rises by 4.1 pp while W falls by 2.4 pp, so this is not evidence of better judgment. RoboBrain's terminal-action coverage falls from 69.1% to 48.5%, showing sensitivity of termination propensity to the changed contract. Stop cannot express the categorical state judgment required by SV, although its world predicate remains measurable.

N.3 Visual History and Structured Within-Episode Memory

We evaluate four paired baselines and seven history/memory conditions, each covering the same 1,000 frozen episode IDs (Table 25). The paired baselines are $W/B = 30.5/25.6$, $38.1/20.2$, $27.3/26.2$, and $40.7/18.4\%$ for Qwen3-VL-32B (T), Qwen3-VL-32B, Qwen3-VL-8B, and InternVL3.5-38B, respectively; they are distinct from the primary comparison runs.

Both controls retain the configured 20-turn dialogue history and original episode budgets. The visual-history profile supplies up to six recent RGB observations, including the current frame, with temporal roles and the current-frame targeting rule. The number of frames grows from one to six as the episode progresses; the history condition provides no action feedback. The memory profile instead retains one current RGB frame and adds structured action/stall, spatial-search, and object-interaction records, reset within each episode.

The memory package is a *combined memory-and-feedback intervention*: its structured records include the preceding action, whether the agent is holding an object, and action-error information such as `too_far`, `hand_full`, and interaction failures. Consequently, its effects cannot be attributed to memory alone. These records remain local to the episode, and the intervention is distinct from both the RGB-only baseline and the limited-feedback condition in Table 3.

Table 25: **History and memory controls.** Deltas are treatment minus paired control in percentage points. The evaluated scopes differ between interventions.

Intervention	Model	ΔW	ΔB
Six-frame history	Qwen3-VL-32B (T)	-1.9	-0.5
	Qwen3-VL-32B	+2.5	+3.6
	Qwen3-VL-8B	-1.8	-2.6
Memory + feedback	InternVL3.5-38B	+0.6	+2.1
	Qwen3-VL-32B (T)	+5.2	+4.2
	Qwen3-VL-32B	-1.8	+2.6
	Qwen3-VL-8B	-2.5	-5.9

Both outcomes rise for one history configuration and fall for two; we do not extrapolate across model families. Memory raises B for three configurations and lowers it for one. Qwen3-VL-32B illustrates the distinction between outcomes: B rises 2.6 pp while W falls 1.8 pp. Across tested additions that preserve the exact B definition, none improves both W and B for every evaluated configuration. This supports diagnostic use of the controls, not a claim of interface invariance.

We did not test an optional-report success contract, cross-episode persistent memory, an isolated ablation of richer controller feedback beyond Table 3, or a second simulator. Although W does not depend on report correctness for a fixed trajectory, changing the reporting contract can change that trajectory; W is therefore not an optional-report counterfactual. Appendix G is fixed-trajectory accounting, not an interaction intervention.

O Qualitative Hardware Demonstrations

Four physical-robot search/approach demonstrations exercise an observation–action–report loop: object approach, room entry and fan search, and two person-approach examples. The four videos are included in the code demo and will be hosted on the project page, `https://xpeng-robotics.github.io/vigil/`. Two clips are external recordings; two are step-sampled person-approach replays from controlled interventions. Replay timing is not rollout latency, and a model's success report is not an independently verified physical outcome. These qualitative examples demonstrate use of the protocol in a hardware loop; they do not estimate an FR/NR cost ratio or establish quantitative transfer of the AI2-THOR model profiles. They are excluded from all quantitative tables.

P Data and Code Availability

The paper's numerical comparisons use the original frozen 1,000-episode selection. Corrected task definitions are maintained separately, including the SV target corrections discussed in Appendix M.1. Evaluation on corrected tasks constitutes a new experiment; those results should not be substituted for the original comparison. The release defaults to neutral wording, whereas the primary comparison uses the baseline prompt in Appendix A. Both prompt texts are supplied separately.

The supplementary material includes the deterministic summary-state mapper, per-episode SV audit, unmapped examples, report confusion counts, and companion metrics. Evaluation code, task definitions, and demonstration videos are available at `https://github.com/xpeng-robotics/VIGIL`. The project page is `https://xpeng-robotics.github.io/vigil/`. Original benchmark code and task definitions use Apache-2.0; third-party simulator, scene, and model terms remain applicable.

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 abstract and introduction state the paper's central contribution as making terminal report behavior at episode closure observable through explicit reporting, and the experiments support this scope through disaggregated outcome analysis, failure profiles, and step-level evidence (Abstract; Secs. 1, 4).

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper includes a dedicated limitations discussion covering simulator scope, the distinction between observable reports and latent belief, dependence on a fixed public contract, and the fact that the benchmark isolates one failure axis rather than replacing broader embodied evaluation (Sec. 5).

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: This paper is an empirical benchmark and analysis paper and does not present formal theoretical results requiring proofs.

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: [No]

   Justification: The paper specifies the task selection, scoring contract, prompts, budgets, and robustness protocols. Complete model-specific serving configurations are not enumerated for every evaluated system (Appendix J.1).

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: Code and corrected task definitions are available at `https://github.com/xpeng-robotics/VIGIL`; the project page is `https://xpeng-robotics.github.io/vigil/`. The supplement provides audit outputs, but full original runtime configurations and trajectories are not included (Appendix P).

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: The paper specifies the frozen task selection, observation/action contract, model panel, prompts, episode budgets, scoring, and matched interface/history/memory interventions. Evaluated models are not fine-tuned on the benchmark (Sec. 3; Appendices A–E, J.1, N).

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: The paper reports variability-oriented robustness statistics for the main open-weight configurations via paired repeated runs (mean±half-range over two runs) under identical contracts and frozen episodes, with run-level setup details and sample scope documented in the appendix (Sec. 4.1; Appendix J, J.3, Table 16).

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: The appendix reports the inference framework (vLLM) and closed-source API identifiers (Appendix J.1), but does not disclose specific GPU hardware, per-run wall-clock time, or total project compute.

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: The authors have reviewed the NeurIPS Code of Ethics, and the quantitative evaluation uses simulated household tasks; the qualitative hardware demonstrations do not contribute human-subject measurements or quantitative benchmark outcomes.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper includes a broader impacts paragraph discussing both potential benefits for diagnosing embodied-agent reliability and risks from treating contract-specific performance as general competence or calibrated self-knowledge (Sec. 5).

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: The paper does not release a new high-risk generative model or a scraped web-scale dataset; its main contribution is an evaluation benchmark and analysis protocol for embodied agents in simulation.

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: [Yes]

    Justification: The paper cites all external assets it builds on. The release records Apache-2.0 terms for original benchmark code and task definitions, with upstream simulator, scene, and model terms retained separately. All assets are used within their terms of use.

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 benchmark design, task family specifications, scoring rules, and evaluation protocol are documented in the paper (Sec. 3) and appendix (Appendix D–J.1).

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: The paper does not involve crowdsourcing experiments or research with human subjects.

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: The paper does not involve human subjects research.

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: Evaluated multimodal foundation models are central to the paper's empirical study, and the appendix also documents an LLM-assisted episode proposal stage in benchmark construction (Secs. 1–4; Appendix E).