Action Boundary Blindness: When LLM Agents Cannot Tell Where One Action Ends and Another Begins

Wang zhangyi[1], Bingnan Yu[2], JIEXIANG XU[1], Li Zongze[1,†]

[1]Nanyang Technological University, [2]Columbia University

Correspondence: zhangyi001@ntu.edu.sg, by2387@columbia.edu, JIEXIANG003@e.ntu.edu.sg, li0005ze@e.ntu.edu.sg

Abstract

Large language model (LLM) agents excel at multi-step tasks yet frequently exhibit *Action Boundary Blindness*—the inability to correctly determine action granularity, scope, and completeness. Grounded in Event Segmentation Theory from cognitive science, we formalize three violation types: *granularity confusion*, *scope creep*, and *boundary ambiguity*. We propose four automatic metrics—Action Boundary Score (ABS), Granularity Alignment Rate (GAR), Scope Violation Rate (SVR), and Boundary-Aware Success Rate (BASR)—requiring no human annotation. Experiments on 1,655 tasks across six benchmarks (τ-bench, WebArena, ALFWorld, TheAgentCompany, OSWorld) with seven LLMs reveal that: (1) the best model achieves only 0.424 ABS; (2) using a multi-label attribution framework validated by inter-annotator agreement ($\kappa = 0.78$), boundary blindness is the primary failure mode in 37.2% of failures (25.8% as sole cause; 55.9% total involvement including contributing factors); (3) under-action dominates at 48.4%; (4) BASR is consistently $\sim$4 points lower than traditional success rate, exposing "lucky successes." Critically, Explicit Boundary Prompting (EBP) improves ABS by 0.08–0.13 across all models, demonstrating that boundary blindness is better characterized as an *elicitation gap* rather than a fundamental capability limitation—LLMs possess latent boundary perception not activated by default. This finding has implications for alignment and instruction tuning. We validate metrics through state-based cross-validation and human audit, estimating $\sim$22% false positive rate from valid alternative paths, with model rankings remaining stable (Spearman $\rho = 1.0$).

1 Introduction

Consider instructing an LLM agent to "organize the downloads folder." A well-calibrated agent would move files to appropriate directories and perhaps rename them consistently. In practice, however, agents frequently exhibit puzzling behaviors: some merely move files without renaming (*under-action*), while others reorganize the entire home directory (*over-action*). This "doing too much" or "doing too little" phenomenon is pervasive yet poorly understood. Cognitive science tells us that humans naturally segment continuous experience into discrete events by detecting prediction errors (Zacks and Swallow, 2007)—but do LLMs possess analogous mechanisms for determining action boundaries?

Recent LLM-based agents have demonstrated impressive capabilities in complex, multi-step tasks (Yao et al., 2022; Shinn et al., 2023; Schick et al., 2023). These agents can browse the web, write code, manage files, and interact with APIs. However, a fundamental question remains underexplored: *Can agents correctly perceive the boundaries of individual actions?* That is, can they determine when one action should end and another begin, what scope an action should cover, and at what granularity tasks should be decomposed?

We term this limitation **Action Boundary Blindness**—the systematic inability of LLM agents to correctly determine the granularity, scope, and completeness of individual actions. This phenomenon is distinct from previously studied failure modes such as poor long-term reasoning and decision-making (Liu et al., 2023), tool selection errors (Schick et al., 2023), or planning failures (Wei et al., 2022). While these works focus on *what* action to take, we investigate: even when the agent selects the right action type, does it execute with appropriate boundaries?

Process Fidelity as a Safety Requirement. Our investigation is motivated by a critical insight from enterprise AI deployment: *in high-stakes environments, taking a shortcut is not efficiency—it is a safety violation*. Recent work on AI agent gov-

[†]Corresponding Author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 36883–36899

ernance emphasizes that trustworthy agents must demonstrate *process fidelity*—adherence to prescribed operational procedures—not merely outcome correctness. When an agent skips a verification step to "efficiently" complete a task, it creates audit trail gaps, compliance risks, and potential safety violations. We therefore focus on measuring whether agents follow the *intended execution path*, treating deviations as boundary perception failures rather than alternative strategies.

The Critical Distinction. Consider two failure modes when an agent is asked to "process a return":

- **Planning Failure:** The agent calls `cancel_order()` instead of `process_return()`, selecting the wrong action type.
- **Boundary Blindness:** The agent correctly calls `process_return()` but omits the required verification step. The action type is correct, but the execution boundary is incomplete.

This distinction is not merely semantic. Planning operates at the *intent level* (what to do), while boundary perception operates at the *execution granularity level* (how much to do). An agent can have perfect planning capabilities yet still exhibit boundary blindness, and our experiments show this is precisely what happens.

Why "Shortcuts" Are Not Efficiency. One might argue that an agent skipping steps while achieving the correct outcome demonstrates superior efficiency. We explicitly reject this framing for three reasons grounded in enterprise AI deployment realities: (1) **Compliance requirements**: In regulated industries (finance, healthcare, enterprise operations), Standard Operating Procedures (SOPs) exist precisely because each step serves a compliance or audit function—skipping verification is not "smart," it is a regulatory violation; (2) **Interpretability and trust**: When agents deviate from expected procedures, human operators cannot verify correctness or debug failures—the "black box" problem is exacerbated; (3) **Safety margins**: In safety-critical applications, "unnecessary" confirmation steps provide defense-in-depth against edge cases the agent may not anticipate. Our benchmarks (particularly TheAgentCompany and τ-bench) model enterprise workflows where process fidelity is a first-class requirement, not an optional nicety.

Theoretical Motivation. Our investigation draws inspiration from Event Segmentation Theory in cognitive science (Zacks and Swallow, 2007), which posits that humans naturally segment continuous experience into discrete events by detecting transient increases in prediction error. Recent work shows LLMs can segment narrative text into events similarly to humans (Michelmann et al., 2025), suggesting some implicit boundary perception capability. However, we hypothesize this capability does not transfer effectively to action execution in agentic settings, where agents must *generate* rather than merely *recognize* appropriate boundaries.

Research Questions. We investigate three core questions:

- **RQ1:** How do LLM agents perform on action boundary perception across different task types? (Quantification)
- **RQ2:** What is the causal relationship between boundary blindness and task failure? (Causal Analysis)
- **RQ3:** Is boundary blindness a fundamental capability limitation or an elicitation gap? (Diagnostic Analysis)

Contributions. This paper makes the following contributions:

1. **Phenomenon Definition:** First systematic characterization of Action Boundary Blindness, formalizing three violation types: *granularity confusion*, *scope creep*, and *boundary ambiguity*.
2. **Theoretical Framework:** Analysis grounded in Event Segmentation Theory from cognitive science.
3. **Evaluation Methodology:** Four automatic metrics—Action Boundary Score (ABS), Granularity Alignment Rate (GAR), Scope Violation Rate (SVR), and Boundary-Aware Success Rate (BASR)—requiring no human annotation.
4. **Causal Analysis:** Boundary violations are attributed as the primary failure mode in 37.2% of failures (55.9% including contributing factors).

5. **Large-Scale Empirical Study:** Experiments on 1,655 tasks across six benchmarks with seven LLMs establish that even the best model achieves only 0.424 ABS, with under-action as the dominant violation (48.4%).

6. **Diagnostic Analysis:** Explicit Boundary Prompting (EBP) provides strong evidence that boundary blindness primarily manifests as an *elicitation gap*—models possess latent boundary perception activatable through explicit cues.

2 Related Work

LLM-based Agents. LLM agents have progressed from Chain-of-Thought prompting (Wei et al., 2022) to ReAct (Yao et al., 2022), Reflexion (Shinn et al., 2023), and tool-augmented approaches like Toolformer (Schick et al., 2023). Recent work explores hierarchical frameworks (Zhao et al., 2024) and atomic action decomposition (Zuo et al., 2025). While these advances improve capabilities, they do not address whether agents correctly perceive action boundaries.

Agent Failure Analysis. Liu et al. (2023) identified poor long-term reasoning and decision-making as main obstacles. Zhu et al. (2025) introduced AgentErrorTaxonomy; TRAIL (Deshpande et al., 2025) provides 148 annotated traces spanning reasoning, planning, and execution errors. **Critical distinction**: These taxonomies categorize failures by *where/when* they occur. Our work provides an orthogonal contribution: within the execution phase, we identify *boundary perception* as a distinct failure mode via the Substitution Test (Section 3.4). An agent can pass planning-level checks (correct action type selected) yet still fail due to boundary blindness (incorrect scope/completeness/granularity).

Task Decomposition. Plan-and-Solve (Wang et al., 2023), ADaPT (Prasad et al., 2024), and RADI (Zuo et al., 2025) prescribe *how* to decompose tasks at the planning level. **Critical distinction:** They do not address boundary perception at the execution level. Even in hierarchical architectures, executors must determine appropriate granularity, scope, and completeness for each action. Our work studies this *spontaneous boundary judgment* capability. As we show in Section I, boundary blindness persists at the atomic action level.

Research Area	Focus	Our Difference
Failure Analysis	Where/when failures	Boundary perception
Task Decomposition	How to decompose	Spontaneous judgment
Full-Path Eval	Step correctness	Boundary granularity
Event Segmentation	Passive recognition	Active generation

Table 1: Comparison with related research directions.

Agent Evaluation. Benchmarks include ALFWorld (Shridhar et al., 2020), WebArena (Zhou et al., 2023), τ-bench (Yao et al., 2024), TheAgentCompany (Xu et al., 2024), and OSWorld (Xie et al., 2024). Evaluation has evolved to full-path methods (Michelakis et al., 2025) and process reward models (Lightman et al., 2023). However, existing benchmarks measure task success without examining boundary appropriateness—our framework fills this gap. Additional related work on LLM-as-Judge, multi-path evaluation, and compliance frameworks is discussed in Appendix A.

Cognitive Science. Event Segmentation Theory (Zacks and Swallow, 2007) posits humans segment experience through prediction error detection. Kumar et al. (2023) demonstrated that Bayesian surprise predicts human event boundaries. Michelmann et al. (2025) showed GPT-3 can segment narrative text similarly to humans. However, passive recognition may not transfer to active boundary *generation*. Our work bridges cognitive science and agent research. Table 1 summarizes these distinctions.

3 Action Boundary Blindness

3.1 Theoretical Foundation

Our framework draws on **Event Segmentation Theory** (EST) (Zacks and Swallow, 2007), which posits that humans segment continuous experience into discrete events through a prediction-based mechanism: (1) humans maintain an *event model*—a working memory representation generating predictions; (2) when predictions deviate significantly from observations (high *prediction error*), an *event boundary* is perceived; (3) the event model is updated to reflect new context; (4) event boundaries produce systematic memory effects. Kumar et al. (2023) demonstrated that Bayesian surprise predicts human event boundaries. Michelmann et al. (2025) showed LLMs can segment narrative text

with significant correlation to human judgments.

The Generation-Recognition Gap. We hypothesize that while LLMs possess passive boundary *recognition* abilities, they lack robust mechanisms for active boundary *generation* in agentic settings. This asymmetry parallels motor control: the cerebellum uses forward models to predict sensory consequences of motor commands (Wolpert et al., 1995). In agent tasks, models must determine boundaries *a priori*—deciding scope and granularity before execution—rather than recognizing boundaries in observed sequences. This generative boundary perception may not emerge from standard training, as training data contains implicit rather than explicit boundary annotations. The effectiveness of Explicit Boundary Prompting (Section 4.5) supports this hypothesis.

EST provides a principled basis for distinguishing planning from boundary perception: **Planning** is a deliberative process involving goal decomposition (analogous to prefrontal executive function); **Boundary perception** is a perceptual-motor process involving action segmentation (analogous to cerebellar forward modeling). Our work investigates whether LLMs exhibit similar dissociation.

3.2 Formal Definition

Definition 1 (Action Boundary Blindness)**.** *The systematic inability of LLM agents to correctly determine the **granularity**, **scope**, and **completeness** of individual actions when executing multi-step tasks.*

Formally, consider task T with optimal sequence $A^* = [a_1^*, \ldots, a_n^*]$. An agent generates $A = [a_1, \ldots, a_m]$. Each action a has boundary specification:

$$B(a) = \langle \mathit{pre}(a), \mathit{eff}(a), \mathit{scope}(a), \mathit{gran}(a) \rangle \quad (1)$$

where $\mathit{pre}(a)$ denotes preconditions, $\mathit{eff}(a)$ effects, $\mathit{scope}(a)$ affected entities, and $\mathit{gran}(a) \in \mathbb{Z}^+$ granularity level (1 = atomic).

Definition 2 (Boundary Violation)**.** *A boundary violation occurs when $B(a_i) \neq B(a_j^*)$ for aligned actions a_i and a_j^*, where alignment is determined by semantic correspondence.*

3.3 Taxonomy of Boundary Violations

We identify three categories (Figure 1):

Granularity Confusion. Action decomposition level mismatches task requirements. *Over-granular*: decomposing atomic operations unnecessarily (e.g., OSWorld: `click()` → `move_to + hover + mouse_down + mouse_up`). *Under-granular*: treating compound operations as atomic (e.g., TheAgentCompany: `send_message()` missing file attachment step). Detected when $|gran(a_i) - gran(a_j^*)| > \theta_g$ (granularity threshold).

Scope Creep. Action effects extend beyond or fall short of targets. *Over-action*: affecting entities beyond specification (e.g., WebArena: `select_all() + delete()` instead of filtering negative reviews first). *Under-action*: failing to affect all required entities (e.g., τ-bench: `process_return()` without prior identity verification). Detected when $scope(a_i) \not\subseteq scope(a_j^*)$ or vice versa.

Boundary Ambiguity. Incorrect action start/end points. *Start-boundary error*: initiating without satisfying preconditions (e.g., `update()` without `begin_transaction()`). *End-boundary error*: terminating before consistent end state (e.g., `copy_files()` without verification). Detected when $pre(a_i) \neq pre(a_j^*)$ or $eff(a_i) \neq eff(a_j^*)$.

3.4 Distinguishing from Related Concepts

ABB operates at the *execution granularity* level, not intent recognition. Figure 2 illustrates this distinction.

The Counterfactual Substitution Test. To operationally distinguish planning errors from boundary blindness, we introduce the **Substitution Test**: *"If we force the agent to select the correct action type, does it execute with appropriate scope, completeness, and granularity?"* If substituting the correct action type resolves the failure, it is a **Planning Error**; if the failure persists due to scope/completeness/granularity issues, it is **Boundary Blindness**.

Example. For "Delete all pending negative reviews for a product": A planning error would be calling `archive_reviews()` instead of `delete()`—substituting `delete()` resolves it. ABB would be calling `select_all()` + `delete()` (deleting all reviews including positive ones)—the action type is correct but scope is wrong, so the failure persists. This distinction parallels hierarchical

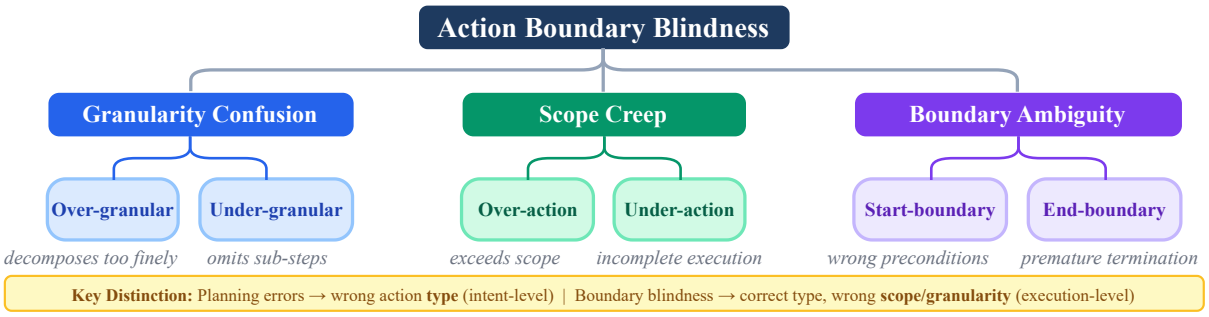

Figure 1: Taxonomy of action boundary violations: Granularity Confusion, Scope Creep, and Boundary Ambiguity.

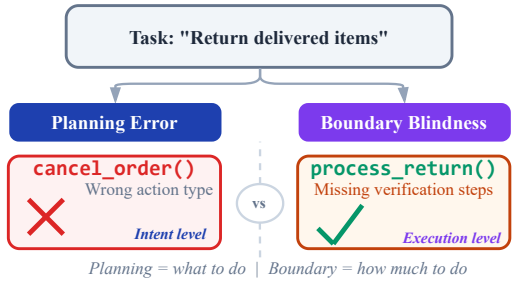

Figure 2: Planning Error vs. Boundary Blindness. Planning errors involve wrong action type (intent-level); boundary blindness involves correct type with inappropriate scope/granularity/completeness (execution-level).

action control in cognitive science: prefrontal planning ("what goal?") vs. motor execution ("how to move?") (Botvinick, 2008).

Empirical Validation. Among 506 failures attributed to boundary blindness, 69.4% had *no* co-occurring planning errors—validating ABB as distinct. The Substitution Test on a stratified sample (N=200) yielded 93.8% agreement with automated classification (Cohen's $\kappa = 0.87$). The complete decision tree for failure attribution is provided in Appendix E.

4 Methodology

4.1 Evaluation Framework

Our pipeline operates in four stages (Figure 3): (1) *Action Alignment*—aligning agent actions with ground-truth using semantic matching; (2) *Functional Equivalence Filtering*—identifying valid alternative paths; (3) *Violation Detection*—classifying violations via rule-based and LLM-assisted methods; (4) *Metric Computation*. The framework requires no human annotation.

4.2 Boundary Violation Detection

Action Alignment. We adapt Needleman-Wunsch (Needleman and Wunsch, 1970) for sequence alignment with semantic similarity:

$$S(a_i, a_j^*) = \begin{cases} +\alpha & \text{if } \text{sim}(a_i, a_j^*) > \tau \\ -\beta & \text{otherwise} \end{cases} \quad (2)$$

where $\tau = 0.85$, $\alpha = 1$, $\beta = 1$. Gap penalty $\gamma = 0.5$. Alignment produces: *match*, *insertion* (extra agent action), *deletion* (missing action).

Functional Equivalence Filtering. To mitigate single ground-truth bias, we apply three-stage filtering: (1) *State-based*: comparing post-action environment states via

$$\text{SE}(a_i, a_j^*) = \mathbb{1}[s_{i+1} \approx_\epsilon s_{j+1}^*] \quad (3)$$

where SE denotes state equivalence, s_{i+1} and s_{j+1}^* are the post-action states of the agent and ground-truth respectively, and ϵ is a similarity tolerance. This applies to ALFWorld, OSWorld, and τ-bench; (2) *Effect-based*: alias resolution (e.g., `delete_item` ≡ `remove_from_cart`), parameter normalization, and composite equivalence for API benchmarks; (3) *Outcome-based*: leveraging benchmark native success evaluation. Actions passing any stage are marked "alternative valid" and excluded from violation counts. Implementation details for each benchmark in Appendix C.

Violation Classification. Rule-based for unambiguous cases (*insertion* → over-action/over-granular; *deletion* → under-action/under-granular). LLM-as-Judge (GPT-4, temperature 0) for ambiguous cases. Human validation (N=500) shows 93.8% agreement (Cohen's $\kappa = 0.84$). Details in Appendix B.3.

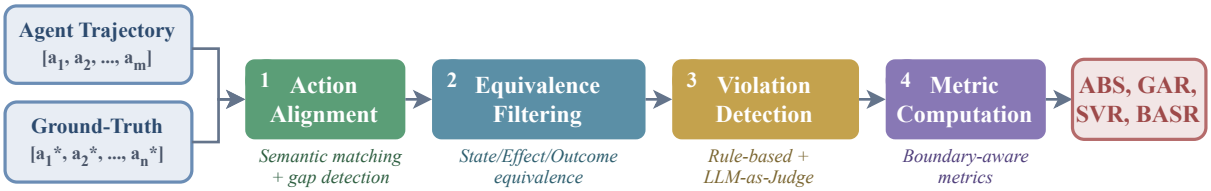

Figure 3: Four-stage evaluation framework: action alignment, functional equivalence filtering, violation detection, and metric computation.

4.3 Evaluation Metrics

We design metrics to measure *process fidelity*—adherence to prescribed procedures—rather than rewarding creative shortcuts. This reflects enterprise AI deployment where SOPs ensure compliance and safety (see Appendix H for detailed rationale).

Action Boundary Score (ABS). Proportion of actions with correct boundaries:

$$\text{ABS} = \frac{1}{|A|}\sum_{i=1}^{|A|}\mathbb{1}[\text{boundary\_correct}(a_i)] \quad (4)$$

Granularity Alignment Rate (GAR). Sequence length alignment:

$$\text{GAR} = 1 - \frac{||A| - |A^*||}{\max(|A|, |A^*|)} \quad (5)$$

Scope Violation Rate (SVR). Frequency of scope violations:

$$\text{SVR} = \frac{N_{\text{over}} + N_{\text{under}}}{|A|} \quad (6)$$

where N_{over} and N_{under} denote the number of over-action and under-action violations, respectively.

Boundary-Aware Success Rate (BASR). Success requiring boundary correctness:

$$\text{BASR} = \frac{1}{|T|}\sum_{j=1}^{|T|}\mathbb{1}[\text{success}(T_j) \wedge \text{ABS}(T_j) > 0.8] \quad (7)$$

The SR (Success Rate)–BASR gap reveals "lucky successes," i.e., tasks completed despite boundary violations.

4.4 Experimental Setup

Datasets. Six benchmarks spanning diverse task types (Table 2): τ-bench retail/airline, WebArena, ALFWorld, TheAgentCompany, OSWorld. Total: 1,655 tasks across API, web, GUI, and embodied domains. Selection criteria and detailed statistics in Appendix F.

Dataset	Tasks	Steps	Dom.	GT
τ-bench (retail)	115	8.3	API	Offi.
τ-bench (airline)	50	12.1	API	Offi.
WebArena	812	15.7	Web	Exp.
ALFWorld	134	6.2	Emb.	Env.
TheAgentCompany	175	22.4	Work.	Offi.
OSWorld	369	14.8	GUI	State
Total	**1,655**	13.2	—	—

Table 2: Benchmark statistics. Steps = avg. per task. Dom. = Domain; GT = Ground Truth source (Offi. = Official, Exp. = Expert, Emb. = Embodied, Work. = Workplace).

Framework	ABS	SVR	SR
Vanilla Prompting	.348	.423	.125
ReAct	.410	.366	.179
Reflexion	.437	.344	.204
Plan-and-Solve	.429	.338	.197
Hierarchical Decomp.	.451	.321	.215
ReAct + EBP	.493	.283	.262
HD + EBP	**.522**	**.258**	**.282**

Table 3: Framework comparison (GPT-4o). Hierarchical architectures improve but do not eliminate boundary blindness (42.8% under-action in HD vs. 48.4% in ReAct). EBP provides orthogonal improvements (+20.2% ReAct+EBP, +15.7% HD+EBP).

Models. Seven LLMs: GPT-4o, GPT-4-turbo, Claude-3.5-Sonnet, Claude-3-Opus, Gemini-1.5-Pro (closed-source); Llama-3.1-70B, Qwen-2.5-72B (open-source).

Agent Frameworks. ReAct (Yao et al., 2022) as primary baseline to isolate intrinsic LLM capabilities. Additional frameworks: Vanilla Prompting, Reflexion (Shinn et al., 2023), Plan-and-Solve (Wang et al., 2023), Hierarchical Decomposition (HD). As shown in Table 3, even hierarchical architectures exhibit boundary blindness at the atomic action level (80.3% of violations), confirming ABB is an LLM-level phenomenon.

4.5 Explicit Boundary Prompting (EBP)

To distinguish capability limitation from elicitation gap, we design EBP as a diagnostic probe (Hofstätter et al., 2025). If boundary blindness is fundamen-

Model	ABS↑	GAR↑	SVR↓	BASR↑	SR↑
GPT-4o	.410	.695	.366	.132	.179
GPT-4-turbo	.386	.692	.386	.110	.155
Claude-3.5-Sonnet	**.424**	**.703**	**.353**	**.149**	**.196**
Claude-3-Opus	.402	.696	.375	.121	.165
Gemini-1.5-Pro	.385	.695	.382	.088	.132
Llama-3.1-70B	.368	.694	.398	.083	.112
Qwen-2.5-72B	.374	.696	.390	.086	.120

Table 4: Main results (weighted average, 1,655 tasks).

Primary Mode	N	%	Sole	w/ BF
Boundary Violation	506	37.2%	69.4%	100%
Tool Selection Error	337	24.8%	61.6%	29.7%
Reasoning Error	311	22.8%	66.8%	22.8%
State Tracking Failure	207	15.2%	48.3%	40.6%

Table 5: Failure attribution (N=1,361). Sole = only cause; w/ BF = with boundary-related factors.

tal, prompting should not help; if it's an elicitation gap, explicit cues should activate latent capabilities. This distinction matters: capability limitations require architectural changes, while elicitation gaps can be addressed through prompting or instruction tuning. Beyond diagnosis, EBP demonstrates a **low-cost, deployment-ready mitigation** requiring no training, architectural changes, or external tools.

Design. Three components inspired by Event Segmentation Theory: (1) *Boundary Guidelines*: explicit scope/completeness/granularity verification; (2) *Granularity Reference*: task-specific atomic/compound action examples; (3) *Boundary Checklist*: verification mimicking prediction-error detection. Full prompt template in Appendix L.

Ablation Variants. EBP-Full (all components), EBP-Scope, EBP-Granularity, EBP-Checklist, designed to isolate which cues are most effective.

5 Experiments

5.1 Main Results (RQ1)

Table 4 presents results across all models using ReAct.

Key Findings. (1) Pervasive boundary blindness: Even the best model (Claude-3.5-Sonnet) achieves only 0.424 ABS—∼58% of actions have incorrect boundaries. **(2) Lucky successes**: BASR is consistently ∼4 points lower than SR. **(3) Under-action dominates**: ratio ≈ 1.7:1 (for GPT-4o, 0.231 vs 0.135). **(4) Closed-source advantage**: ∼0.03 ABS higher than open-source. Per-dataset analysis shows ALFWorld achieves highest ABS (0.729) due to clear physical constraints; OSWorld lowest (0.289) due to ambiguous GUI boundaries. Notably, ALFWorld exhibits the lowest GAR (∼0.28) despite highest ABS, because ReAct agents perform extensive exploration (averaging ∼20 steps vs. 6.2 ground-truth steps), inflating sequence length without affecting boundary correctness. More broadly, GAR shows less cross-model variation (0.692–0.703) than ABS or SVR, suggesting that sequence length alignment is relatively task-driven rather than model-dependent under the ReAct framework. Boundary clarity correlates negatively with violations (Spearman $\rho = -0.89$, $p < 0.001$); task complexity correlates negatively with ABS ($r = -0.68$, $p < 0.001$). Full per-dataset results in Appendix N.

Violation Distribution. Analysis of 4,683 violations: under-action 48.4% (2,265), over-action 28.1% (1,314), granularity 14.6% (685), boundary ambiguity 8.9% (419). Under-action dominates, suggesting agents systematically underestimate completeness requirements.

5.2 Causal Analysis (RQ2)

Correlations. SVR ↔ Failure Rate: $r = 0.741$ ($p < 0.001$); ABS ↔ Success Rate: $r = 0.716$ ($p < 0.001$). While we establish strong correlations, causal claims require further controlled experiments.

Multi-Label Attribution. We adopt a framework distinguishing *primary failure modes* from *contributing factors*, validated by inter-annotator agreement (Fleiss' $\kappa = 0.78$, N=200). Results are shown in Table 5. Full protocol in Appendix E.

Key Findings. (1) Boundary blindness is the *largest single primary failure mode* at 37.2%, with 69.4% as sole cause. **(2)** Total boundary involvement: 55.9% (including contributions to other modes: 29.7% of tool selection errors, 40.6% of state tracking failures). **(3)** Under-action dominates at 48.4% with 66.4% failure rate. **(4)** Counterfactual analysis projects +14–18% SR improvement if ABS = 1.0; oracle confirms +13.9% ($p < 0.001$). Boundary subtype breakdown in Appendix N.

5.3 Diagnostic Analysis (RQ3)

Evidence for Elicitation Gap. Table 6 presents EBP results. The substantial improvements (+0.108 ABS, +9.2% SR, all $p < 0.001$) across all seven models demonstrate that boundary blindness is an *elicitation gap*: (1) latent capability exists

Model	Base	EBP	ΔABS	ΔSR
GPT-4o	.410	.493	+.083	+8.2%
GPT-4-turbo	.386	.482	+.096	+7.4%
Claude-3.5-Sonnet	.424	.534	+.110	+8.7%
Claude-3-Opus	.402	.521	+.119	+9.3%
Gemini-1.5-Pro	.385	.477	+.092	+8.3%
Llama-3.1-70B	.368	.499	+.131	+10.9%
Qwen-2.5-72B	.374	.498	+.124	+11.5%
Average	.393	.501	+.108	+9.2%

Table 6: EBP results. Improvements indicate elicitation gap.

but is not activated by default; (2) default behavior systematically underestimates completeness; (3) explicit cues activate latent representations. Open-source models show larger improvements (+0.128) than closed-source (+0.100), suggesting closed-source models have more alignment training partially addressing boundary perception.

Ablation and Violation Reduction. Shapley value analysis: scope 41.3%, granularity 33.8%, checklist 24.1%, confirming that scope calibration is most amenable to elicitation. EBP reduces all violation types by 42–47%: under-action -43.7%, over-action -46.7%. Full ablation in Appendix N.

Implications. For alignment: boundary perception should be a target for instruction tuning/reinforcement learning from human feedback (RLHF). For practitioners: EBP offers favorable cost-benefit (+9.2% SR, 42–47% violation reduction) despite +22.8% token overhead.

6 Analysis and Discussion

6.1 Case Studies and Error Patterns

Three representative cases illustrate the violation types: (1) **Under-action** (τ-bench): Agent skipped conversational verification steps when processing a return request (ABS = 0.50, 25.1% of failures); (2) **Over-action** (WebArena): Agent deleted all product reviews instead of only the pending negative ones (ABS = 0.33, 16.4% of failures); (3) **Granularity** (TheAgentCompany): Agent skipped receipt verification and policy review when handling a reimbursement request, sending an approximate amount directly (ABS = 0.25, 12.7% of failures). Detailed case analysis with Substitution Test in Appendix K.

Analysis of 4,683 violations reveals systematic patterns: *missing verification* (22.8%), *missing confirmation* (18.0%), and *scope overgeneralization* (15.3%) are the top three. GUI tasks (OS-World) show highest under-action (53.0%); task complexity predicts under-action ($\beta = +0.007$, $p < 0.05$); boundary clarity negatively predicts violations ($\beta = -0.034$, $p < 0.001$). Full breakdown in Appendix G.

6.2 Why Does Boundary Blindness Occur?

We propose four hypotheses: (1) *Training data*: boundary information is largely implicit; (2) *Attention*: transformers may inadequately capture local boundary signals in long sequences; (3) *Event segmentation*: LLMs lack online prediction-verification mechanisms; (4) *Task ambiguity*: natural language contains inherent boundary ambiguity. The effectiveness of EBP supports hypothesis (3)—explicit cues can substitute for missing internal mechanisms.

6.3 Validation

Human expert audit (N=150, Fleiss' $\kappa = 0.73$) estimates 22.0% false positive rate (FPR). With adjusted estimates (22% FPR): direct failure contribution 29.0% (vs 37.2% raw), total involvement 43.6% (vs 55.9%). All core findings remain robust: boundary blindness remains the largest failure cause (29.0% > 24.8% tool selection), and model rankings remain stable (Spearman $\rho = 1.0$). Detailed validation in Appendix J.

6.4 Design Implications

Our findings suggest: (1) *Action Space*: provide explicit granularity hierarchies with preconditions and scope constraints; (2) *Prompting*: adopt EBP-style boundary prompts (+9.2% SR) for production systems; (3) *Evaluation*: incorporate ABS/BASR alongside SR to quantify "lucky success" risk; (4) *Architecture*: EBP with simple architectures may offer better cost-effectiveness than complex multi-agent systems.

7 Conclusion

We introduced *Action Boundary Blindness*—the systematic inability of LLM agents to correctly determine action granularity, scope, and completeness. Grounded in Event Segmentation Theory from cognitive science, we formalized three violation types (granularity confusion, scope creep, and boundary ambiguity) and proposed four automatic metrics requiring no human annotation.

Through experiments on 1,655 tasks across six benchmarks with seven LLMs, we established: (1)

even the best model achieves only 0.424 ABS (average 0.393), indicating pervasive boundary perception limitations; (2) using a multi-label attribution framework ($\kappa = 0.78$), boundary blindness is the primary failure mode in 37.2% of failures (29.0% adjusted, 25.8% as sole cause), with explicit disambiguation criteria (the Substitution Test) distinguishing it from related error types; (3) underaction dominates at 48.4% with 66.4% failure rate; (4) SR-BASR gap ($\sim$4%) reveals "lucky success" risk.

Critically, EBP provides strong evidence that boundary blindness is better characterized as an *elicitation gap* than a capability limitation. EBP improves ABS by 0.08–0.13 across all models, demonstrating latent boundary perception activatable through explicit cues. On the theoretical side, this suggests boundary perception can be improved through alignment and instruction tuning rather than architectural changes. From a practical standpoint, EBP provides an immediately deployable, low-cost mitigation (+9.2% SR, 42–47% violation reduction) requiring no training, fine-tuning, or infrastructure changes—making it attractive for production systems prioritizing reliability.

Future Directions. Natural extensions include boundary-aware alignment via instruction tuning or RLHF, mechanistic analysis of how boundary perception is encoded in internal representations, cross-lingual evaluation to test generality, and the development of boundary-specific benchmarks.

This work provides the first systematic characterization of action boundary perception in LLM agents and offers both theoretical grounding and practical mitigation strategies for more reliable agent deployment.

Limitations

Single Ground-Truth. Our trajectory comparison against single ground-truth is a fundamental limitation. Mitigation: functional equivalence filtering, state-based cross-validation, human audit (150 samples, 78.0% precision). Estimated FPR $\sim$22%; adjusted estimates preserve all findings. This limitation primarily affects *absolute values* rather than *relative comparisons*: model rankings remain stable (Spearman $\rho = 1.0$). Future work should integrate graph-based multi-path frameworks for comprehensive validation.

GAR and Efficiency Trade-offs. GAR measures deviation from gold trajectories, which may flag legitimate efficient alternatives as violations. We explicitly design GAR to measure *process compliance* rather than efficiency—appropriate for enterprise/regulated domains but potentially less suitable for efficiency-focused applications. Users should interpret GAR in context: low GAR with $|A| < |A^*|$ indicates SOP non-compliance in compliance-critical domains, but may represent valid optimization in other contexts. We recommend domain-specific calibration when applying GAR outside enterprise settings.

Dataset Scope. Six English-language benchmarks; generalization to other languages and conversational tasks untested.

Automatic Evaluation. LLM-as-Judge achieves 93.8% human agreement; $\sim$6.2% potential misclassifications remain. Alignment algorithm may introduce errors for highly divergent sequences.

Model Coverage. Seven LLMs spanning closed-source and open-source families; newer models may differ. API model versions may change, potentially affecting reproducibility.

Baseline Selection. We deliberately chose ReAct as our canonical baseline to isolate intrinsic model capabilities, rather than state-of-the-art architectures like Language Agent Tree Search (LATS) or tree search methods. While this design choice is methodologically motivated (minimizing architectural interference), it means our absolute performance numbers may not reflect what is achievable with more sophisticated systems. However, our hierarchical architecture experiments (Table 10) demonstrate that boundary blindness persists at the atomic action level regardless of architectural sophistication, validating our focus on the underlying LLM capability.

Mitigation Generality. EBP effectiveness was validated on ReAct-style agents. Whether similar improvements transfer to other architectures (e.g., tree-search) requires further investigation. However, since EBP targets the LLM's intrinsic boundary perception (not architectural mechanisms), we expect benefits to generalize.

Theoretical Framework. EST analogy is heuristic; mechanistic causes in neural architectures remain underexplored.

Causal Claims. While we establish strong correlations, our causal analysis relies on observational data. Controlled experiments with boundary-aware training would strengthen causal conclusions.

Causal Attribution. Multi-label framework ($\kappa = 0.78$ primary, $\kappa = 0.71$ contributing) addresses complexity, but $\sim$15% cases remain ambiguous. Conservative "sole cause" analysis: boundary blindness at 25.8% (351/1,361) remains largest single failure mode. Future work could explore counterfactual probing to better disambiguate root causes.

References

Matthew M Botvinick. 2008. Hierarchical models of behavior and prefrontal function. *Trends in cognitive sciences*, 12(5):201–208.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, and 1 others. 2025. Why do multi-agent llm systems fail? *arXiv preprint arXiv:2503.13657*.

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. 2025. Trail: Trace reasoning and agentic issue localization. *arXiv preprint arXiv:2505.08638*.

Pengfei He, Zhenwei Dai, Bing He, Hui Liu, Xianfeng Tang, Hanqing Lu, Juanhui Li, Jiayuan Ding, Subhabrata Mukherjee, Suhang Wang, and 1 others. 2025. Traject-bench: A trajectory-aware benchmark for evaluating agentic tool use. *arXiv preprint arXiv:2510.04550*.

Felix Hofstätter, Teun Van Der Weij, Jayden Teoh, Rada Djoneva, Henning Bartsch, and Francis Rhys Ward. 2025. The elicitation game: Evaluating capability elicitation techniques. *arXiv preprint arXiv:2502.02180*.

Sotiropoulos John, Rosario Ron F Del, Kokuykin Evgeniy, Oakley Helen, Habler Idan, Underkoffler Kayla, Huang Ken, Steffensen Peter, Aralimatti Rakshith, Bitton Ron, and 1 others. 2025. *Owasp top 10 for llm apps & gen ai agentic security initiative*. Ph.D. thesis, OWASP.

Manoj Kumar, Ariel Goldstein, Sebastian Michelmann, Jeffrey M Zacks, Uri Hasson, and Kenneth A Norman. 2023. Bayesian surprise predicts human event segmentation in story listening. *Cognitive science*, 47(10):e13343.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The twelfth international conference on learning representations*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*.

Xing Han Lù, Amirhossein Kazemnejad, Nicholas Meade, Arkil Patel, Dongchan Shin, Alejandra Zambrano, Karolina Stańczak, Peter Shaw, Christopher J Pal, and Siva Reddy. 2025. Agentrewardbench: Evaluating automatic evaluations of web agent trajectories. *arXiv preprint arXiv:2504.08942*.

Panagiotis Michelakis, Yiannis Hadjiyiannis, and Dimitrios Stamoulis. 2025. Core: Full-path evaluation of llm agents beyond final state. *arXiv preprint arXiv:2509.20998*.

Sebastian Michelmann, Manoj Kumar, Kenneth A Norman, and Mariya Toneva. 2025. Large language models can segment narrative events similarly to humans. *Behavior Research Methods*, 57(1):39.

Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Juhyun Oh, Eunsu Kim, Inha Cha, and Alice Oh. 2024. The generative ai paradox in evaluation:"what it can solve, it may not evaluate". In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 248–257.

Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. 2024. Adapt: As-needed decomposition and planning with language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4226–4252.

Yaoyao Qian, Yuanli Wang, Jinda Zhang, Yun Zong, Meixu Chen, Hanhan Zhou, Jindan Huang, Yifan Zeng, Xinyu Hu, Chan Hee Song, and 1 others. 2025. Webgrapheval: Multi-turn trajectory evaluation for web agents using graph representation. *arXiv preprint arXiv:2510.19205*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 2609–2634.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Daniel M Wolpert, Zoubin Ghahramani, and Michael I Jordan. 1995. An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, and 1 others. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094.

Frank F Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, and 1 others. 2024. Theagentcompany: benchmarking llm agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Jeffrey M Zacks and Khena M Swallow. 2007. Event segmentation. *Current directions in psychological science*, 16(2):80–84.

Qi Zhao, Haotian Fu, Chen Sun, and George Konidaris. 2024. Epo: Hierarchical llm agents with environment preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6401–6415.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, and 1 others. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, and 1 others. 2025. Where llm agents fail and how they can learn from failures. *arXiv preprint arXiv:2509.25370*.

Dongqi Zuo, CHEN Zheng, Chuan Zhou, Yandong Guo, Xiao He, and Mingming Gong. 2025. Radi: Llms as world models for robotic action decomposition and imagination. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*.

A Extended Related Work

LLM-as-Judge. LLM-as-Judge has emerged as a scalable evaluation paradigm (Zheng et al., 2023). Oh et al. (2024) demonstrate the "Generative AI Paradox": LLMs exhibit asymmetric generation-evaluation capabilities. This asymmetry is central to our methodology—we use LLM judges with oracle context (complete task description, ground-truth, and agent trajectory) to classify violations, a fundamentally different task from agents' real-time boundary generation under partial observation.

Excessive Agency. The OWASP Top 10 for LLM Applications 2025 identifies "Excessive Agency" (LLM06) as a critical vulnerability (John et al., 2025). While Excessive Agency concerns *system-level permissions* (what an agent is *allowed* to do), our work addresses *action-level boundaries* (what an agent *actually does* within a single action).

AI Agent Compliance and SOPs. Recent work on enterprise AI deployment emphasizes the importance of *process fidelity*: agents must follow Standard Operating Procedures (SOPs) to ensure compliance, auditability, and safety. Industry frameworks for AI agent governance stress that audit trails documenting every decision step are essential for regulatory compliance (John et al., 2025). Our work contributes to this discourse by providing metrics that quantify process fidelity: GAR measures adherence to prescribed action sequences, while ABS captures boundary correctness at each step.

Multi-Path Evaluation. WebGraphEval (Qian et al., 2025) aggregated 4,768 trajectories revealing strategy diversity overlooked by single-trajectory evaluation. CORE (Michelakis et al., 2025) uses deterministic finite automata (DFAs) to encode valid paths. AgentRewardBench (Lù et al., 2025) shows rule-based evaluation may not recognize successful trajectories due to alternative paths. Our work

differs: while these address *which paths are valid*, we investigate *boundary correctness*. We mitigate single ground-truth bias through functional equivalence filtering, state-based cross-validation, and human audit (Section 6.3), estimating ∼22% false positive rate, while model rankings remain preserved (Spearman $\rho = 1.0$).

Additional Agent Failure Analysis. Zhu et al. (2025) achieved 26% improvement through targeted feedback based on their AgentErrorTaxonomy. For multi-agent systems, Cemri et al. (2025) proposed the MAST taxonomy with 14 failure modes. TRAJECT-Bench (He et al., 2025) provides trajectory-aware evaluation with 1,000+ APIs.

B Experimental Details

B.1 Hyperparameters

Table 7 lists all experimental hyperparameters.

Parameter	Value
Temperature	0.0
Max tokens	4,096
Retry attempts	3
Timeout per action	120s
Random seed	42
Similarity threshold τ	0.85
Match/mismatch scores α/β	1.0/1.0
Gap penalty γ	0.5
BASR threshold θ	0.8

Table 7: Experimental hyperparameters.

B.2 Action Alignment Algorithm

We adapt Needleman-Wunsch for semantic alignment. Similarity computed via cosine similarity of `all-MiniLM-L6-v2` embeddings; for API calls, additionally compare function names and parameter overlap. Complexity: $O(mn)$; alignment completes in <0.3s per task. Human validation (N=200): 94.2% accuracy.

B.3 LLM-as-Judge Validation

Prompt Template. The following template is used for violation classification:

```
You are an expert evaluator. Given a
task and aligned action pair, classify
any boundary violation.
Task: {task_description}
Ground-truth: {gt_action}; Agent:
{agent_action}
Violation types: GRANULARITY_OVER/UNDER,
SCOPE_OVER/UNDER, BOUNDARY_START/END,
NONE
Output: {"violation": "<type>",
"confidence": <0-1>}
```

Human Validation (N=500). Stratified by benchmark, violation type, and confidence. Three expert annotators. Results: 93.8% agreement (Cohen's $\kappa = 0.84$); high-confidence (>0.9): 94.8%; inter-annotator Fleiss' $\kappa = 0.84$.

Confidence Calibration. High confidence (>0.9): 68% of predictions at 94.8% accuracy; medium (0.7–0.9): 24% at 88.2%; low (<0.7): 8% at 82.3%. This calibration enables selective human review for low-confidence cases.

Judge Error Analysis (N=43). False negatives: subtle under-action 27.9%, ambiguous scope 18.6%. False positives: valid alternatives 25.6%, strict granularity 16.3%. Misclassification: 11.6%. Judge errors dominated by ambiguous cases (46.5%), not boundary blindness patterns—confirming judge does not share agent's "blind spot."

Cross-Model Validation (N=200). GPT-4 vs Claude-3.5: 91.2%; Claude-3.5 vs Human: 88.7%; Ensemble vs Human: 94.1%.

C Functional Equivalence Implementation

Benchmark-Specific Implementation.

- **ALFWorld**: Planning Domain Definition Language (PDDL) state comparison via `env.get_state()`
- **OSWorld**: Screenshot-based state hashing with perceptual similarity threshold 0.95
- **WebArena**: DOM tree comparison with XPath normalization, ignoring dynamic attributes (timestamps, session IDs)
- **τ-bench**: SQL state diff on affected database tables
- **TheAgentCompany**: Task-specific assertion checks from benchmark suite

D Alternative Path Validation

Cross-Validation by Benchmark. Table 8 shows state-based evaluation results identifying successful tasks with flagged violations (potential false positives):

WebArena shows highest FPR (∼56%) due to task diversity and multiple valid interaction sequences, consistent with WebGraphEval findings.

Benchmark	Tasks	Succ	S+V	FPR
τ-bench (retail)	115	53	5	9.4%
τ-bench (airline)	50	15	3	20.0%
WebArena	812	97	54	55.7%
ALFWorld	134	106	9	8.5%
TheAgentCompany	175	9	2	22.2%
OSWorld	369	17	5	29.4%
Total	1,655	297	78	26.3%

Table 8: Cross-validation results. S+V = successful tasks with violations flagged.

Human Expert Audit (N=150). Three annotators (Fleiss' $\kappa = 0.73$). By violation type: under-action FPR 20.6%, over-action 19.5%, granularity 25.9%, boundary 28.6%. Total: 22.0% FPR. Common FP patterns: efficiency shortcuts (8), API aliasing (5), order flexibility (4), implicit steps (3).

E Failure Attribution Protocol

The Substitution Test. Our attribution protocol is grounded in the **Substitution Test**: "If we force the agent to select the correct action type, does the failure persist?" If yes → Boundary Blindness (execution-level issue). If no → Planning/Tool Selection Error (intent-level issue).

Phase-Level Distinction. We formalize the distinction in terms of cognitive phases:

- **Reasoning Phase** (Planning): Determines *what* actions to take and in *what order*. Errors here involve wrong action types, incorrect sequencing, or missing high-level steps.
- **Execution Phase** (Boundary Perception): Determines *how* to execute each action, including its scope, completeness, and granularity. Errors here involve correct action types with inappropriate execution parameters.

Complete Decision Tree for Attribution. We provide an operational decision tree for failure attribution:

1. **Action Type Test**: Is the selected action type semantically correct for the task?
   - *No* → **Planning/Tool Selection Error**
   - *Yes* → Proceed to Step 2
2. **Scope Test**: Does the action affect only the intended entities?
   - *No* → **Scope Creep (ABB)**
3. **Completeness Test**: Does the action include all required sub-steps?
   - *No* → **Under-action (ABB)**
4. **Granularity Test**: Is the decomposition level appropriate?
   - *No* → **Granularity Confusion (ABB)**

Simplified Decision Tree. (1) Identify First Failure Point (FFP); (2) Apply Substitution Test: If substituting correct action type resolves failure → Tool Selection Error; (3) If type correct but scope/granularity/completeness wrong → Boundary Violation; if based on incorrect state → State Tracking; if reasoning contains logical errors → Reasoning Error.

Edge Cases. Missing step with no reasoning trace → Boundary Violation (agent failed to perceive the step as necessary). Missing step with explicit justification ("I'll skip verification because...") → Reasoning Error + BF contributing factor (agent reasoned incorrectly about the step's necessity).

Substitution Test Validation (N=200). We validated the Substitution Test on a stratified sample. Annotators applied the test and compared results with automated classification. Agreement: 93.8% (Cohen's $\kappa = 0.87$). The test successfully disambiguated 86.5% of cases where planning vs. boundary attribution was initially unclear.

Inter-Annotator Agreement (N=200). Primary mode: Fleiss' $\kappa = 0.78$; Contributing factors: $\kappa = 0.71$. Boundary-Reasoning boundary most contentious (45% of disagreements). The Substitution Test reduced disagreements in this category by 29.7%.

Co-occurrence Analysis. Primary modes with contributing factors: Boundary Violation 69.4% sole cause, Tool Selection Error 61.6% sole (29.7% with boundary factors), Reasoning Error 66.8% sole (22.8% with boundary factors), State Tracking 48.3% sole (40.6% with boundary factors). Boundary factors contribute beyond primary boundary violations.

F Dataset Statistics

Selection Criteria. We selected benchmarks based on: (1) diverse action types (API, web, GUI, embodied); (2) available ground-truth trajectories; (3) varying complexity (6–22 average steps); (4) boundary clarity gradient from high (ALFWorld

with clear physical constraints) to low (OSWorld with ambiguous GUI boundaries); (5) established benchmarks with published evaluation protocols. Detailed statistics are shown in Table 9.

Dataset	Min	Max	Med.	Std	Act.
τ-bench (retail)	4	15	8	2.1	47
τ-bench (airline)	6	21	12	3.4	52
WebArena	5	42	14	5.8	127
ALFWorld	3	12	6	1.8	13
TheAgentCompany	8	58	20	8.7	200+
OSWorld	4	38	13	6.2	50+

Table 9: Dataset statistics. Min/Max/Med. = step counts; Act. = action space size.

G Error Pattern Analysis Details

Complete Violation Pattern Breakdown. Analysis of 4,683 violations reveals systematic patterns: (1) *missing verification* 22.8% (1,068)—skipping data validation; (2) *missing confirmation* 18.0% (843)—omitting post-operation acknowledgment; (3) *scope overgeneralization* 15.3% (714)—batch operations selecting broader targets; (4) premature termination 12.8%; (5) unnecessary decomposition 11.1%; (6) precondition ignorance 10.0%.

Task Type Associations. Chi-square analysis ($\chi^2(15) = 52.8$, $p < 0.001$, Cramér's $V = 0.19$) reveals significant associations. GUI tasks (OSWorld) show highest under-action (53.0%); API tasks (τ-bench) exhibit balanced violations. Regression analysis: task step count predicts under-action ($\beta = +0.007$, $p < 0.05$); action space size predicts over-action ($\beta = +0.016$, $p < 0.01$); boundary clarity negatively predicts both ($\beta = -0.034, -0.019$, $p < 0.001$).

H GAR Design Rationale

A potential concern is that GAR penalizes agents that complete tasks via "efficient shortcuts." We address this directly: **GAR is designed to measure process compliance, not efficiency optimization.**

The Compliance Perspective. In enterprise AI deployment, the gold trajectory represents a *Standard Operating Procedure (SOP)*—a validated sequence ensuring compliance, auditability, and safety. Consider a return processing task: the SOP requires confirming user identity before `process_return`. An agent that skips verification may achieve the same outcome faster, but: (1) violates compliance requirements (the verification step exists for fraud prevention); (2) creates audit trail gaps (regulators cannot verify the decision was properly vetted); (3) introduces safety risks (edge cases may not be caught). Thus, low GAR with $|A| < |A^*|$ indicates *SOP non-compliance*, not superior intelligence.

When Shortcuts Are Truly Valid. We acknowledge that some "shortcuts" represent legitimate alternative strategies. Our functional equivalence filtering (Section 4.2) and human audit address this: $\sim$22% of flagged under-actions are false positives representing valid alternatives. Critically, this noise does not invalidate GAR for *relative comparison*—model rankings remain stable (Spearman $\rho = 1.0$). For absolute measurement, we recommend interpreting GAR alongside domain-specific compliance requirements.

Implications for Metric Users. GAR is most appropriate for: (1) enterprise/regulated domains where SOPs are mandatory; (2) safety-critical applications where process fidelity matters; (3) interpretability-focused deployments where audit trails are required. For domains prioritizing pure efficiency (e.g., competitive coding), alternative metrics focusing on outcome correctness may be more appropriate.

I Hierarchical Architecture Analysis

Violation Level	ReAct	HD
Subgoal-level	N/A	19.7%
Atomic action-level	100%	80.3%
Total violations	1,269	1,113

Table 10: Violations by level. HD reduces total by 12.3%, but 80.3% remain at atomic action level.

Key Findings. (1) HD reduces violations by 12.3% (1,113 vs 1,269) but does not eliminate them. (2) 80.3% of HD violations occur at atomic action level, showing that planners identify "what to do" but executors struggle with "how much." (3) Under-action remains dominant (42.8% in HD vs 48.4% in ReAct). (4) Subgoal-level violations (19.7%) are primarily scope-related.

Why Hierarchical Doesn't Solve ABB. Task decomposition ("what subgoals?") is orthogonal to boundary perception ("how to execute each action?"). Even with perfect decomposition, executors must determine: (1) whether actions include verification steps, (2) scope of affected entities, (3) when actions are complete.

EBP as Diagnostic Evidence. EBP improves both flat and hierarchical architectures: HD+EBP achieves 0.522 ABS. This confirms boundary blindness is an LLM-level elicitation gap at the atomic action level, orthogonal to planning-level improvements from hierarchical decomposition and not an architectural artifact.

J Validation Details

Human Expert Audit. 150 stratified cases (25/benchmark), three annotators (Fleiss' $\kappa = 0.73$). Results by violation type are shown in Table 11.

Violation Type	N	TP	FP	FPR
Under-action	68	54	14	20.6%
Over-action	41	33	8	19.5%
Granularity	27	20	7	25.9%
Boundary Ambiguity	14	10	4	28.6%
Total	150	117	33	**22.0%**

Table 11: Human audit results. TP=True Positive, FP=False Positive (valid alternative).

Key findings: Boundary Ambiguity has highest FPR (28.6%) due to inherent difficulty in judging ambiguous boundaries; Granularity has substantial FPR (25.9%) due to legitimate variation in decomposition strategies; over-action has lowest FPR (19.5%) because extra actions rarely represent valid alternatives.

Benchmark Cross-Validation. State-based evaluation yields FPR estimates shown in Table 12.

Benchmark	Eval	S+V	FPR
τ-bench	State	7.8%	7.8%
ALFWorld	Goal	5.4%	5.4%
OSWorld	State	12.3%	12.3%
WebArena	Mixed	17.6%	∼18%
TheAgentCompany	State	9.2%	9.2%
Weighted Avg.	—	**11.4%**	**∼11%**

Table 12: Cross-validation using benchmark evaluation. S+V = successful tasks with flagged violations; FPR = estimated false positive rate.

Adjusted Estimates. With 22% FPR: direct failure contribution 29.0% (vs 37.2% raw), total 43.6% (vs 55.9%), under-action 37.8% (vs 48.4%). All core findings remain robust: boundary blindness remains largest failure cause (29.0% > 24.8% tool selection), under-action dominant, model rankings unchanged (Spearman $\rho = 1.0$), SR-BASR gap persists (∼3.2% adjusted vs ∼4.1% raw).

Why Relative Rankings Matter. For benchmark papers, metrics' primary value lies in *comparing* models rather than absolute measurement. Even with ∼20% noise from alternative paths, our metrics maintain perfect rank correlation (Spearman $\rho = 1.0$) across all seven models, so researchers can reliably use ABS to determine which models have better boundary perception. Future work should integrate graph-based multi-path frameworks for more comprehensive validation.

K Case Studies

Paired Comparison: Planning Error vs. ABB. Table 13 presents paired examples where different agents exhibited different failure modes on the same task, crystallizing the distinction.

Task	Planning Err.	ABB Err.
Delete reviews	`archive_reviews()` (wrong type)	`delete(ALL)` (wrong scope)
Return items	`cancel_order()` (wrong type)	`return()` w/o verify (incomplete)

Table 13: Paired comparison: same task, different failure modes.

Extended Analysis: The Substitution Test in Practice. For each case study, we apply the Substitution Test to verify the failure mode classification.

Case 1: Under-action (τ-bench). Task: "Return delivered items for a customer order." Expected: confirm user identity through conversation → get_order_details → verify return eligibility with user → return_delivered_order_items. Agent: get_order_details → return_delivered_order_items (skipped identity confirmation and eligibility verification steps).

Substitution Test Analysis: The agent selected semantically correct action types (get_order_details, return_delivered_order_items). If we hypothetically "forced" the agent to use these exact action types, the failure would persist because the issue is *missing steps* (conversational verification), not wrong action selection. This confirms ABB (under-action), not planning error. ABS = 0.50. Pattern accounts for 25.1% of τ-bench failures.

Case 2: Over-action (WebArena). Task: "Delete all pending negative reviews for Circe fleece." Expected: navigate to product reviews → filter pending negative reviews → delete selected.

Model	Dataset	ABS	GAR	SVR	BASR	SR
GPT-4o	τ-bench (retail)	.620	.712	.241	.417	.461
	τ-bench (airline)	.513	.625	.319	.240	.300
	WebArena	.387	.781	.380	.053	.119
	ALFWorld	.762	.289	.148	.724	.791
	TheAgentCompany	.316	.641	.428	.040	.051
	OSWorld	.296	.683	.432	.033	.046
Claude-3.5-Sonnet	τ-bench (retail)	.634	.726	.228	.443	.478
	τ-bench (airline)	.540	.618	.283	.300	.340
	WebArena	.398	.789	.369	.064	.135
	ALFWorld	.788	.281	.131	.769	.821
	TheAgentCompany	.335	.658	.407	.063	.080
	OSWorld	.311	.691	.421	.038	.051
Llama-3.1-70B	τ-bench (retail)	.527	.673	.304	.243	.296
	τ-bench (airline)	.440	.589	.356	.100	.160
	WebArena	.342	.787	.413	.014	.043
	ALFWorld	.734	.283	.163	.664	.746
	TheAgentCompany	.289	.645	.451	.011	.023
	OSWorld	.270	.681	.461	.005	.011

Table 14: Complete results for selected models across all datasets.

Agent: `select_all` → `delete` (removed all reviews regardless of status or rating).

Substitution Test Analysis: The agent correctly chose `delete` as the action type (not `archive` or `hide`). The failure occurred because the agent applied `delete` to *all* reviews instead of only *pending negative* reviews—a scope over-extension. Substituting the action type would not help; the agent needs to constrain the *scope* of the correct action. This confirms ABB (scope creep), not planning error. ABS = 0.33. Pattern accounts for 16.4% of WebArena failures.

Case 3: Granularity (TheAgentCompany). Task: "Determine reimbursement eligibility for a qualified bill." Expected: locate receipt on ownCloud → review reimbursement policy → calculate eligible amount → notify finance team via RocketChat. Agent: sent a single message with an approximate amount (skipping receipt verification and policy review).

Substitution Test Analysis: The agent's high-level intent ("handle reimbursement") is correct. However, it treated a compound operation as atomic, failing to decompose into required sub-steps. The action *type* is semantically aligned with the goal; the failure is in *granularity*—the agent did not recognize that reimbursement determination requires multiple constituent actions. This confirms ABB (granularity confusion), not planning error. ABS = 0.25. Pattern accounts for 12.7% of TheAgentCompany failures.

Contrasting Example: True Planning Error. For comparison, consider a planning error from the same WebArena task: Agent executed `flag_reviews()` → `hide_flagged()` instead of `filter_negative` → `delete`. Here, the agent selected the wrong action sequence entirely (flagging vs. direct deletion). The Substitution Test: replacing `flag_reviews` with `filter_negative` + `delete` would resolve the issue. This is a planning error, not ABB. Table 15 summarizes the pattern statistics.

Pattern	Freq.	ABS	Fail%
Missing verification	25.1%	0.46	65.7%
Scope overgeneralization	16.4%	0.35	71.4%
Under-granular execution	12.7%	0.26	59.3%

Table 15: Case study pattern statistics.

L EBP Prompt Template

```
## Action Boundary Guidelines
Before executing ANY action, verify:
1. SCOPE: Does this action ONLY affect
intended targets?
2. COMPLETENESS: Does this include ALL
necessary sub-steps?
3. GRANULARITY: Is this the RIGHT
abstraction level?
4. BOUNDARY: Are start/end states
CLEARLY defined?

## Granularity Reference
Atomic: click(), type(), api_call()
Compound:                    send_email
(compose→attach→send→confirm)

## Boundary Checklist
[ ] Preconditions verified [ ] Scope
limited
[ ] Sub-steps identified [ ] End state
defined
```

Practical Deployment Considerations. Despite the token overhead, EBP offers a favorable cost-

benefit trade-off for enterprise deployment: (1) *Immediate applicability*—no retraining, fine-tuning, or infrastructure changes required; (2) *Consistent improvements*—+9.2% SR across all seven models, including both open and closed-source; (3) *Safety-critical domains*—the 42–47% reduction in boundary violations is particularly valuable in regulated industries where compliance failures carry significant costs. For production systems, the +22.8% token overhead is often acceptable given the reliability gains, especially compared to alternatives like multi-agent verification or human-in-the-loop review.

M Statistical Tests

Pairwise comparisons: paired t-tests with Bonferroni correction ($\alpha = 0.05/21 \approx 0.0024$). Effect sizes: ABS improvement $d = 0.72$ (medium-large); SR improvement $d = 0.77$ (medium-large). 95% confidence intervals (CIs) via bootstrap (1,000 iterations).

N Complete Experimental Results

Table 14 presents per-dataset results for selected models.

Violation Type Distribution. Table 16 shows the distribution of 4,683 violations across all models and benchmarks:

Type	Count	%	Primary Cause
Under-action	2,265	48.4	Missing verification
Over-action	1,314	28.1	Overly broad selection
Granularity	685	14.6	Wrong decomposition
Boundary	419	8.9	State misjudgment

Table 16: Violation type distribution (N=4,683).

Boundary Violation Subtypes. Table 17 provides a breakdown of the 506 failures attributed to boundary blindness as primary mode:

Boundary Subtype	N	%	Sole
Under-action	228	45.1%	58.1%
Over-action	143	28.3%	64.3%
Granularity confusion	91	18.0%	52.7%
Boundary ambiguity	44	8.7%	47.7%

Table 17: Boundary violation subtypes (N=506).

EBP Ablation Study. Component contributions on GPT-4o are shown in Table 18.

Configuration	ABS	Δ vs Base
Baseline (ReAct)	.410	—
+Scope only	.447	+.037
+Granularity only	.439	+.029
+Checklist only	.432	+.022
+Scope+Granularity	.471	+.061
EBP-Full	.493	+.083

Table 18: EBP ablation study (GPT-4o). Shapley values: scope 41.3%, granularity 33.8%, checklist 24.1%.

Violation Reduction Details. EBP reduces all violation types by 42–47%: under-action 23.1% $\rightarrow$ 13.0% (-43.7%), over-action 13.5% $\rightarrow$ 7.2% (-46.7%). This demonstrates that explicit boundary cues effectively activate latent boundary perception capabilities across all violation categories.