The Cognitive Bandwidth Bottleneck: Shifting Long-Horizon Agents from Planning with Actions to Planning with Schemas

Anonymous ACL submission

Abstract

Long-horizon LLM agents typically choose from an environment-enumerated list of grounded actions (Planning with Actions, PwA). While this interface reduces formatting risk, scaling the list increases screening load and can bottleneck planning. We study Planning with Schemas (PwS), which replaces action enumeration with a small set of action templates that the agent instantiates into executable actions. We introduce a cognitive-bandwidth perspective of this trade-off and quantify screening vs. instantiation pressure using context length proxy and failure actions proxy. Across four interactive environments, we identify an inflection point: PwA excels at small interfaces but degrades as action lists grow, whereas PwS becomes increasingly competitive and often superior. A distractor-injection stress test reproduces the PwA drop by enlarging the action space alone, isolating screening load and highlighting schema instantiation as the key remaining bottleneck.

1 Introduction

Large Language Model (LLM) agents have made rapid progress on interactive, long-horizon tasks by improving reasoning, memory, and planning components (Xi et al., 2025; Zhang et al., 2025; Shinn et al., 2023). As these agents move from curated benchmarks toward open-ended autonomy, a more basic design choice becomes increasingly consequential: ***As environmental action space scales, what is the optimal action representation for long-horizon agents?*** In real world, the grounded action space frequently grows combinatorially and becomes infeasible to enumerate; therefore, agents are rarely provided with a complete, clean list of all valid grounded actions, and must instead generate executable actions from compact action schemas. This motivates a careful investigation of altering appropriate action representations as a scalability variable rather than an implementation detail.

We consider two paradigms. **Planning with Actions (PwA)** presents an environment-enumerated list of valid grounded actions (e.g., `take apple 1 from desk 1`) and frames decision-making as selection from that list. PwA reduces formatting risk, but as the interface grows, the agent must screen many candidates each step before it can act, creating high **Environment Understanding Pressure (EU)**. In contrast, **Planning with Schemas (PwS)** presents a small set of abstract templates (e.g., `move [OBJ] to [OBJ]`) and requires the agent to instantiate a schema into an executable command. PwS compresses the interface and alleviates EU, but introduces **Schema Instantiation Pressure (SI)**: selecting the right schema, grounding arguments, and producing a correct command.

To formalize this trade-off, we introduce a **Cognitive Bandwidth Perspective** that views an agent step as a pipeline with limited effective budget, where the action interface shifts pressure between EU (screening a large, noisy interface) and SI (grounding and formatting a command). We approximately operationalize these pressures with simple, observable proxies computed from prompts and trajectories: (i) **cumulative context length** as a proxy for EU pressure, and (ii) **invalid** and **repetitive** actions as proxies for SI-related pressure. This framing leads to a concrete question: ***as action spaces scale, when does PwA's EU pressure dominate PwS's SI pressure, and how does that crossover depend on model capability?***

Guided by this framework, we run a systematic evaluation across four interactive environments with progressively larger per-step interfaces: TextCraft (Prasad et al., 2024), WebShop (Yao et al., 2022), ALFWorld (Shridhar et al., 2021), and SciWorld (Wang et al., 2022). Across a diverse set of LLMs (DeepSeek-AI et al., 2024; OpenAI, 2023; Anil et al., 2023) and reasoning-oriented models (DeepSeek-AI et al., 2025; Yang et al., 2025; Bai et al., 2025), we observe a consistent

representation-choice inflection point: PwA is typically favorable at smaller interface scales, but degrades as the enumerated list grows, while PwS becomes increasingly competitive and can dominate in high-dimensional regimes. To isolate mechanism from task semantics, we design a controlled **cognitive-load stress test** in ALFWorld that injects large numbers of syntactically plausible but task-irrelevant distractor actions into the PwA interface, without changing the underlying environment dynamics. This intervention reproduces PwA degradation purely by increasing EU pressure, validating enumeration-induced screening load as a primary limiter. Finally, we analyze what determines the inflection location and find it depends on two capability axes—*PwA proficiency* and *SI capability*; long-reasoning can help, but strong SI appears more closely associated with agentic, multi-turn tool-use post-training (Bai et al., 2025; Fang et al., 2025a). We will make our code available upon acceptance.

Our contributions are summarized as follows:

- We identify action-space representation as a first-class scalability variable for long-horizon LLM agents, contrasting PwA and PwS through EU and SI pressure.
- We introduce the Cognitive Bandwidth Perspective and operational proxies to quantify EU pressure (context length) and SI pressure (invalid/repetitive actions).
- We provide cross-environment, multi-model evidence of a representation-choice inflection point, and validate the mechanism with a controlled stress test.
- We map how the inflection depends on model capability, highlighting SI as a key bottleneck and motivating multi-turn tool-use post-training as a promising way.

2 Related Work

LLM-based Agents. Prior work on constructing LLM-based agents can be broadly classified into two categories. **1. Prompting and In-Context Learning.** These methods steer LLM behavior through prompt design, including explicit reasoning (Yao et al., 2023; Wei et al., 2022), self-reflection (Renze and Guven, 2024; Shinn et al., 2023), long-term planning (Nayak et al., 2024; Prasad et al., 2024; Sun et al., 2023), role-playing with personas (Xu et al., 2024), and multi-agent debate (Xu et al., 2025a; Du et al., 2024). They are lightweight yet effective and often transfer well across domains. **2. Training Methods.** These approaches update model parameters via supervised fine-tuning with explicit "golden trajectories" (Yuan et al., 2025; Qiao et al., 2024; Xi et al., 2024) or reinforcement learning with environment rewards (Jin et al., 2025; Wang et al., 2025b; Fang et al., 2025b). While strong on specific benchmarks, they are typically resource-intensive and face scalability challenges.

LLMs in Long Context. While current LLMs exhibit extraordinary capability on short-context problems, they falter on tasks requiring synthesis of information across long historical context in either text modality (Zhang et al., 2024; Wu et al., 2025) or multimodality setup (Wang et al., 2025a; Maharana et al., 2024). One research direction involves multi-agent frameworks (Liu et al., 2025; Zhao et al., 2024; Li et al., 2025a), where multiple agents work collaboratively. Another approach is to memorize critical information via either reinforcement learning (Zhou et al., 2025; Yu et al., 2025) or constructing information database (Chhikara et al., 2025; Xu et al., 2025b; Li et al., 2025b), where a external memory is provided to assist agent reasoning.

3 Action Representations and Cognitive Bandwidth Perspective

This section first describes how we construct the two action representations used in our study, including our procedure for deriving compact action schemas from environment-enumerated grounded action lists. We then introduce the **Cognitive Bandwidth Perspective**, a qualitative stage-wise lens that links action representation to paradigm-induced pressures, and define the operational proxies used in subsequent analyses.

3.1 Derivation of Action Representations

Executable Action List. For ALFWorld and SciWorld, the executable action list is obtained directly from the environment at each timestep, following standard practice. For WebShop, which does not natively provide such a list, we generate it by parsing the observation to extract all currently clickable elements. Finally, for the inherently schema-based TextCraft environment, we choose to only study the schema-based paradigm due to the relatively

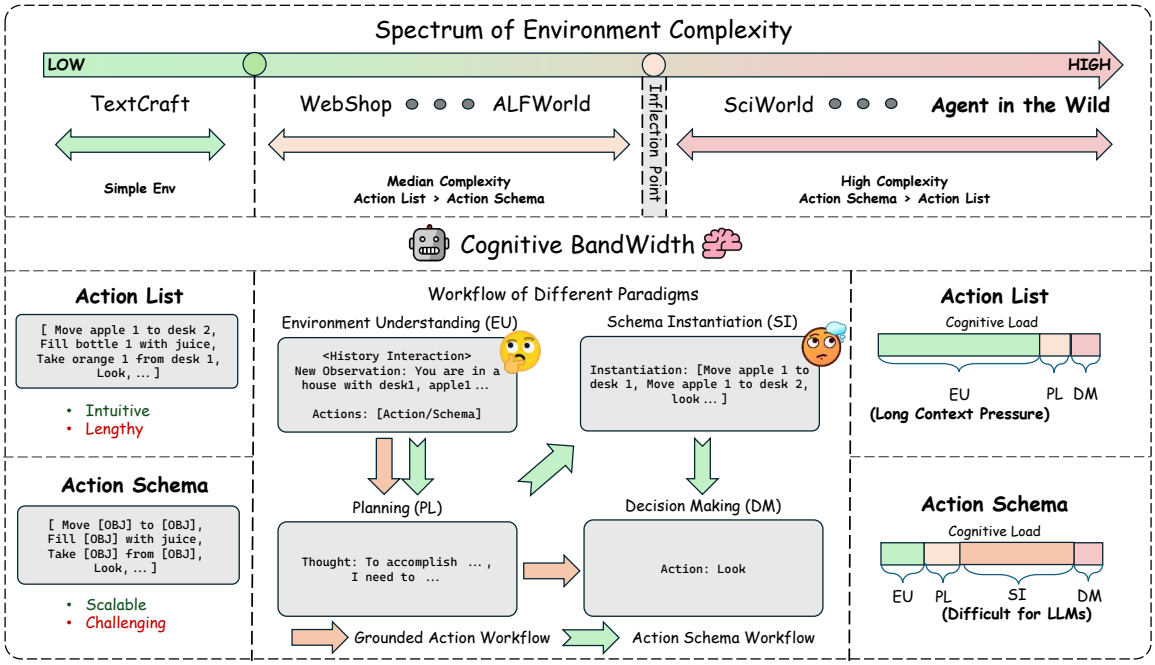

Figure 1: The relationship between environment complexity and agent paradigm effectiveness. In simple environments, Grounded Actions suffice. As complexity grows past an "inflection point", the Action Schema paradigm becomes superior by shifting the cognitive load from processing long action lists (high EU load) to a challenging but scalable Schema Instantiation (SI) step.

Environment	TextCraft	WebShop	ALFWorld	SciWorld
Action List Length	-	3∼10	30∼40	400∼600
Schema List Length	∼3	2	11	26

Table 1: The correlation between per step action space size and environment complexity. Larger per step action space indicates more complex environment.

low difficulty of the environment, evidenced by satisfying performance over various LLMs.

Abstract Action Schema Derivation. For TextCraft and WebShop, the action schemas (e.g., craft [num] [ITEM], click [BUTTON]) are intrinsic to the environments' design and were used directly. For ALFWorld and SciWorld, we derive the schemas by abstracting the complete grounded action lists provided by their environments. Specifically, we replace all object arguments in the actions with a generic [OBJ] token and collect the set of unique resulting templates. We provide the per step grounded action list size and the length of action schemas in Table 1. The comprehensive list of the action schemas used in our experiments is available in Appendix A.

3.2 Cognitive Bandwidth Perspective

To diagnose why performance shifts between planning with actions (PwA) and planning with schemas (PwS) as environments scale, we adopt a **cognitive bandwidth perspective** that decomposes an agent step into stages and tracks where the interface induces additional *processing pressure*. We use "cognitive bandwidth" as a descriptive shorthand for a model's *effective* capacity under long, cluttered contexts; we do *not* claim to measure an intrinsic scalar capacity of an LLM. Instead, we operationalize stage-wise pressure using observable proxies computed from contexts and trajectories, and use them to guide experimental design.

Stage decomposition. We decompose a long-horizon agent's workflow into discrete stages. Let t index interaction steps, H_t denote the textual interaction history up to step $t-1$, o_t the current environment observation, and I_t the textual interface that represents the available action space (a grounded action list for PwA, or a schema/template list for PwS). We consider two common workflows:

- **PwA (action-based): Environment Understanding (EU)** → Planning (PL) → Decision Making (DM), where DM selects an action from the grounded list.
- **PwS (schema-based):** EU → PL → **Schema Instantiation (SI)** → DM, where SI converts an abstract template into a fully-specified executable command (template choice, argument grounding, and formatting).

Model	TextCraft			WebShop			ALFWorld			SciWorld		
	Actions	Schema	Delta	Actions	Schema	Delta	Actions	Schema	Delta	Actions	Schema	Delta
Large Language Models												
Qwen2.5-7B	-	26.0	-	22.0	28.5	6.5	49.0	27.0	-22.0	8.0	8.2	0.2
Qwen3-235B-A22B	-	95.0	-	23.0	27.4	4.4	58.0	5.0	-53.0	43.3	62.2	18.9
Llama-4-Scout	-	65.0	-	17.2	23.3	6.1	27.0	12.0	-15.0	26.9	41.4	14.5
DeepSeek-V3	-	59.0	-	24.7	13.3	-11.4	50.0	7.0	-43.0	22.0	25.5	3.5
Kimi-K2	-	83.0	-	31.6	27.2	-4.4	69.0	45.0	-24.0	40.7	52.9	12.2
Minimax-01	-	15.0	-	13.2	10.0	-3.2	29.0	6.0	-23.0	19.3	28.7	9.4
GPT-4.1-mini	-	96.0	-	20.3	14.6	-5.7	44.0	12.0	-32.0	46.0	44.8	-1.2
GPT-4.1	-	100.0	-	27.1	23.9	-3.2	61.0	16.0	-45.0	43.3	62.2	18.9
Gemini-2.0-flash	-	76.0	-	26.8	17.3	-9.5	40.0	1.0	-39.0	34.5	38.9	4.4
Large Reasoning Models												
DeepSeek-R1	-	97.0	-	36.7	36.0	-0.7	65.0	14.0	-51.0	48.4	46.9	-1.5
Seed-OSS-36B	-	96.0	-	2.0	0.0	-2.0	48.0	25.0	-23.0	49.5	67.4	17.9
Qwen3-235B-A22B$_{\text{Thinking}}$	-	96.0	-	39.7	20.0	-19.7	47.0	23.0	-24.0	47.7	67.3	19.6
LongCat	-	94.0	-	20.3	27.6	7.3	61.0	41.0	-20.0	41.1	47.8	6.7
GPT5-mini	-	88.0	-	11.7	24.7	13.0	19.0	0.0	-19.0	36.0	20.5	-15.5
Gemini-2.5-Flash	-	88.0	-	18.0	15.0	-3.0	40.0	8.0	-32.0	57.2	69.1	11.9
Claude-4.0-Sonnet	-	98.0	-	64.0	64.0	0.0	87.0	27.0	-60.0	60.2	69.1	8.9

Table 2: Model performance comparison across environments of different per step action space complexity. Delta is defined as the model performance gap between PwS paradigm and PwA paradigm. Delta = PwS - PwA.

Conceptually, each stage incurs a latent load $L^{(t)}_{\text{stage}}$, and difficulty accumulates over long horizons. However, since $L^{(t)}_{\text{stage}}$ is not directly observable, we operationalize the interface-induced load shift via two measurable *pressure proxies*: **screening pressure** in EU (when the agent must parse a lengthy context under PwA) and **instantiation pressure** in SI (when the agent must recognize objects and generate executable actions).

Proxy for EU screening pressure: cumulative context length. We approximate EU screening pressure at step t by the number of input tokens the model must process to interpret the state and the context:

$$C_t(\mathcal{M}) = \left|\text{Tok}_{\mathcal{M}}(\text{prompt}(H_t, o_t, I_t))\right|,$$

where $\text{Tok}_{\mathcal{M}}(\cdot)$ is the tokenizer used with model $\mathcal{M}$. In our setup, we select tokenizer of GPT-4o for consistent token count approximation. $\text{prompt}(\cdot)$ is the *exact* text passed to the LLM at step t, including interaction history, current observation, and the actions to be selected or instantiated. Because H_t grows with the trajectory and I_t varies with the different action representations, C_t provides a direct, paradigm-dependent proxy for screening burden.

We summarize context pressure at the episode level with:

$$\bar{C} = \sum_{t=1}^{T} C_t,$$

where T is the number of interaction steps in the episode.

Proxy for SI instantiation pressure: trajectory failure modes. To capture difficulties in producing valid, grounded, and non-wasteful commands, we quantify two failure modes in agent trajectories following Zhang et al. (2025): (1) **Invalid Actions**, where the generated command is rejected by the environment (e.g., malformed syntax or illegal arguments), and (2) **Repetitive Actions**, redundant and repetitive steps that fail to advance the task state. For an episode of length T, we compute:

$$r_{\text{inv}} = \sum_{t=1}^{T} \mathbb{I}[a_t \text{ invalid}],$$

$$r_{\text{rep}} = \sum_{t=1}^{T} \mathbb{I}[a_t \text{ is the first action of repetitive loop}].$$

where a_t is the action executed (or attempted) at step t. While these failures can arise from multiple stages, they are especially diagnostic for PwS

because SI requires correct template selection, argument grounding, and formatting; elevated r_{inv} and r_{rep} indicate higher instantiation pressure and wasted exploration. Our trajectory labeling is implemented via rule-based filters and an LLM judge (Kimi-K2); details are provided in Appendix B.

4 Experiments and Analysis

4.1 Experiment Setup

Environments. Our evaluation testbed comprises TextCraft (Prasad et al., 2024), WebShop (Yao et al., 2022), ALFWorld (Shridhar et al., 2021), and SciWorld (Wang et al., 2022), ordered by increasing per-step action-space complexity (Table 1).

Evaluation Paradigm. Following Yuan et al. (2025), we evaluate on 100 sampled tasks per environment with a 30-step interaction budget per episode. At each step, the agent receives the full interaction history; the action information (grounded list or schema list) is appended to the observation only when the set of available actions changes, for context efficiency and comparison fairness. We report the standard metric for each benchmark: success rate for ALFWorld and average reward elsewhere. Environment and interaction details are in Appendix C; selected models are listed in Appendix D.

4.2 Identifying the Inflection Point

Table 2 compares *planning with actions* (PwA) and *planning with schemas* (PwS) across environments with increasingly large per-step action spaces. The central observation is a *representation crossover*: the better interaction paradigm depends on action-space complexity. In low-to-moderate complexity settings (e.g., WebShop and ALFWorld), PwA typically achieves higher reward than PwS. When the grounded list is short, enumeration provides an explicit, executable command to copy, effectively removing schema instantiation from the pipeline and reducing action-generation errors.

This advantage weakens—and reverses—in the high-complexity regime of SciWorld. Here, the enumerated action list is large and often dominates the prompt, increasing the burden of screening candidates before the agent can plan and commit to an action. In contrast, PwS replaces a long grounded list with a compact schema menu, substantially compressing the representation while shifting difficulty to a single instantiation step.

Figure 2: **Context token growth under enumeration (PwA) vs. schema (PwS) interfaces.** Prompt token count (history, observation, interface) versus normalized progress t/T in ALFWorld (left) and SciWorld (right), split by successful (top) and failed (bottom) episodes. We plot the episode cumulative token growth from trajectory of Kimi-K2 which succeed/fail under both PwA and PwS for demonstration. The 32K context-length dashed line is shown as a visual reference for comparison across panels.

We refer to this crossover as the representation inflection point. Empirically, it lies between ALFWorld and SciWorld in our testbed: below it, PwA is preferable; above it, PwS becomes the more scalable choice.

4.3 EU Pressure: Context Token Growth

The Cognitive Bandwidth Perspective predicts that PwA should incur increasing *EU pressure* as the enumerated interface grows. We measure cumulative context length C_t (Section 3.2) over normalized episode progress t/T, separately for successful and failed episodes (Figure 2). The *relative* PwA–PwS gap is robust.

In ALFWorld (near the inflection point), enumeration increases the prompt length but only moderately: by the end of an episode, PwA inputs are about 1.7–$1.8\times$ longer than PwS (approximately +1.4k tokens in successful runs and +1.9k in failures). This suggests that, at this scale, the additional screening induced by the grounded list remains manageable, so PwA can still benefit from avoiding SI.

In SciWorld, the action footprint becomes dominant. PwA prompts grow rapidly with progress and

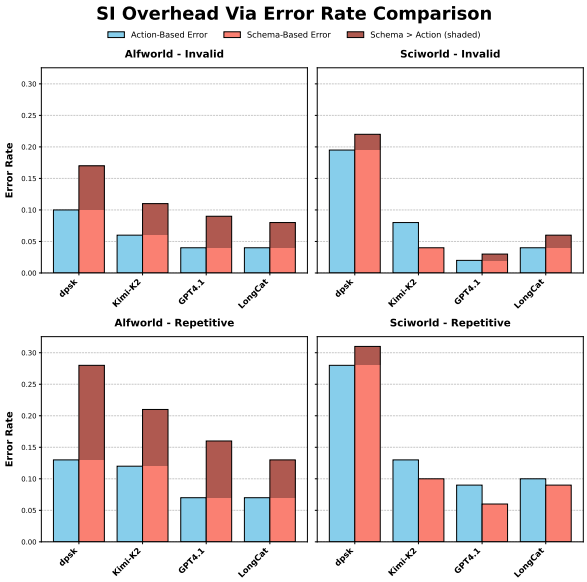

Figure 3: Action-level failure modes under PwA and PwS on ALFWorld (near the inflection point) and SciWorld (beyond it). We report invalid-action rate r_{inv} and repetitive-action rate r_{rep} aggregated over episodes; lower is better.

frequently exceed a 32k-token reference budget, reaching about $10\times$ the PwS footprint in successful episodes ($\approx$ +41k tokens) and up to $19\times$ in failures ($\approx$ +106k tokens). When the action consumes most of the input budget, the model must devote substantially more capacity to filtering and faithfully selecting/copying from a long candidate list, consistent with screening overload as a mechanistic driver of PwA degradation. We next quantify the opposing cost of PwS—*schema instantiation pressure*—via action-level failure modes.

4.4 SI Pressure: Failure Actions

PwS compresses the action representation but requires the agent to *instantiate* a template into an executable command—selecting an appropriate schema, grounding its arguments to entities in the observation/history, and producing a well-formed string accepted by the environment. We measure this *schema instantiation* (SI) pressure using the action-level error rates r_{inv} (invalid actions rejected by the environment) and r_{rep} (repetitive actions that do not advance the task state), as defined in Section 3.2. Invalid actions are detected via environment-specific rejection signatures; repetitive actions are labeled with rule-based filters plus an LLM judge (Kimi-K2).

Figure 3 reveals a regime shift across the inflection point. In ALFWorld, PwS consistently exhibits higher r_{inv} and r_{rep} than PwA. This indicates that when the grounded list is still small enough to screen, the dominant extra burden of switching to PwS is SI itself: mistakes in template choice, entity grounding, or formatting lead to rejected actions and stalled retries, contributing to PwS underperformance.

In SciWorld, the pattern largely reverses: PwA shows higher repetition and non-trivial invalid outputs, *despite* being provided with an enumerated list of valid actions. This is consistent with EU-driven long-context brittleness: when actions and history dominate the prompt, the agent more often fails to reliably screen candidates, understand the environment and faithfully select/copy actions, leading to loops and malformed outputs. By compressing the action representation, PwS reduces screening pressure and stabilizes downstream generation, yielding lower error rates for most models and explaining why PwS becomes competitive or superior in high-complexity environments. Full comparison involving more models are provided in Appendix E.

4.5 Robustness to Schema Variation

Our schemas for ALFWorld and SciWorld abstract away surface forms by replacing object arguments with a generic placeholder. To assess whether our findings depend on this notation, we construct an alternative prompt template that uses slot-specific tokens (e.g., `[OBJ1]`, `[OBJ2]`, `[LOC]`) while keeping all other instructions and evaluation identical, and re-ran on ALFWorld and SciWorld with three non-thinking models and one thinking model. The differences are small and show no consistent direction across models; the qualitative patterns and conclusions in earlier sections remain unchanged. Full results are in Appendix G.

5 Drivers of the Inflection Point

Section 4.2 established a representation-choice crossover: as per-step action spaces grow, PwA increasingly suffers from long, cluttered grounded action lists (EU pressure), while PwS trades reduced screening for schema instantiation (SI) overhead. However, comparing different environments inevitably entangles *action representation scale* with *task semantics*. Here we isolate the causal contribution of representation-induced screening pressure by scaling only the textual actions within a fixed environment, and then characterize how the

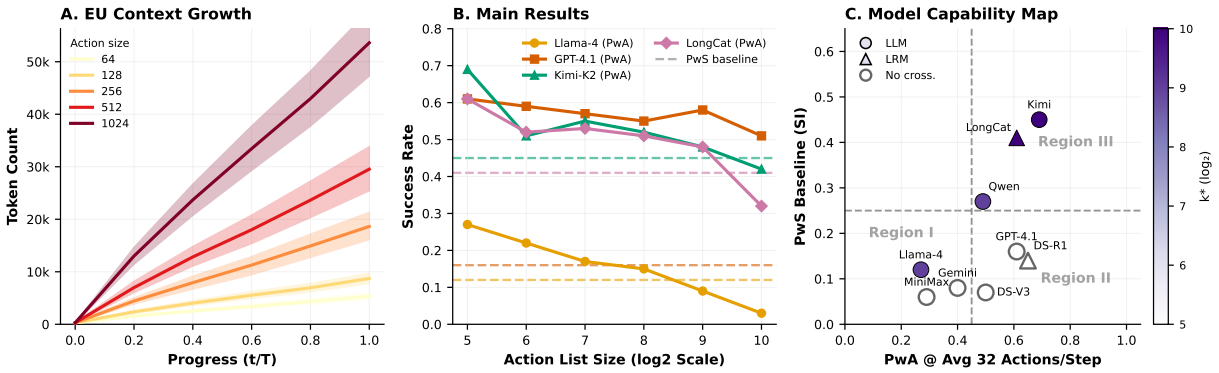

Figure 4: **Controlled EU-load stress test in ALFWorld via distractor injection.** (A) Prompt token count versus normalized progress t/T under enumerated interfaces of size k, increasing monotonically with k. (B) PwA success rate versus k (x-axis in $\log_2$ scale); dashed lines denote the PwS baseline (unchanged by distractors). (C) Capability map: x-axis PwA performance at $k = 32$, y-axis PwS baseline (SI); color indicates the inflection point k^* (in $\log_2$), the smallest tested k where $\mathrm{Perf}_{\mathrm{PwA}}(k) < \mathrm{Perf}_{\mathrm{PwS}}$; open markers indicate no crossover.

resulting inflection point depends on model capability.

5.1 Cognitive-Load Stress Test

Specifically, we conduct a controlled cognitive-load stress test in ALFWorld, which lies near the empirical crossover regime from Section 4.2. At each step t, ALFWorld provides a set of valid grounded actions A_t (on average ~ 35 actions per step). We construct an augmented action list of total size k by injecting $k - |A_t|$ distractor actions $D_t^{(k)}$ and shuffling:

$$\tilde{A}_t^{(k)} = \mathrm{Shuffle}\left(A_t \cup D_t^{(k)}\right).$$

We sweep $k \in \{32, 64, 128, 256, 512, 1024\}$, where $k = 32$ serves as an approximation to the native action space scale. At each step, distractors are freshly sampled and the resulting list is shuffled to ensure valid actions are dispersed.

Distractors are syntactically plausible and schema-aligned but *non-advancing*: they are invalid commands that do not change the world state. Importantly, this intervention leaves the transition and reward behavior induced by the original valid action set A_t unchanged; it only enlarges the *textual action spaces* the agent must screen. Consequently, any performance change can be attributed to increased EU screening burden (and the increased risk of selecting an invalid option), rather than altered task structure. Example distractors are shown in Appendix H.

We intentionally push to $k = 1024$ (exceeding SciWorld's typical per-step list length) because SciWorld's valid actions are often semantically dense and mutually confusable, whereas our injected distractors are, by construction, easier to dismiss; using a larger k better approximates a comparable "needle-in-a-haystack" EU burden.

Unless stated otherwise, all evaluation details follow Section 4.1. PwS is evaluated with the fixed ALFWorld schema set (Appendix A) and is unaffected by distractor injection, serving as a constant baseline for each model.

5.2 Manipulation Check

Figure 4(A) verifies that distractor injection increases EU pressure as measured by the cumulative context length proxy introduced in Section 3.2. Using a fixed reference tokenizer (GPT-4o) to report token counts on a consistent approximation scale across conditions, we observe that the prompt length grows monotonically with k throughout the trajectory. This confirms that the intervention successfully increases the amount of interface text the model must parse and process before committing to an action. Shaded regions denote 95% bootstrap confidence intervals over sampled trajectories with 1000 resamples.

5.3 PwA Degrades with Scale

Figure 4(B) shows that PwA success rate generally declines as k increases, even though the task semantics and the set of valid actions remain unchanged. This reproduces—within a single environment—the qualitative degradation previously observed when moving from smaller interfaces (ALFWorld) to substantially larger ones (SciWorld). The result supports the Cognitive Bandwidth Perspective's mechanistic account: a grounded action list can

impose an EU screening bottleneck that limits long-horizon performance as the context grows.

Because the PwS baseline is constant under distractor injection, the stress test also shows that a within-environment representation inflection can be induced purely by scaling the enumerated interface. Some models eventually satisfy $\mathrm{Perf}_{\mathrm{PwA}}(k) < \mathrm{Perf}_{\mathrm{PwS}}$ at large k (a crossover), while others do not cross within our tested range. The latter indicates that, for those models, PwS remains bottlenecked by SI overhead to such an extent that action enumeration remains preferable even under substantial action noise. More results are provided in appendix I.

5.4 Capability and Performance Crossover

To summarize the stress test at the model level, we define the inflection location $k^* = \min\{\, k \in \mathcal{K} : \mathrm{Perf}_{\mathrm{PwA}}(k) < \mathrm{Perf}_{\mathrm{PwS}} \,\}$ over $\mathcal{K} = \{32, 64, 128, 256, 512, 1024\}$ as the smallest tested interface size at which PwA under distractor injection becomes worse than the PwS baseline; if no crossover occurs within $\mathcal{K}$, we mark "No cross."

Figure 4(C) explains variation in k^* via a two-axis capability map. The x-axis captures PwA proficiency at the native action space scale (approximated by $k = 32$), while the y-axis captures SI capability under PwS, summarized by the PwS baseline success rate. Color encodes k^* (in $\log_2$ scale).

Crucially, Figure 4(C) shows that *a crossover does not by itself imply strong SI*: there are two qualitatively different routes to $\mathrm{Perf}_{\mathrm{PwA}}(k)$ falling below $\mathrm{Perf}_{\mathrm{PwS}}$—(i) *PwS is strong* (high SI), so the schema interface becomes competitive once enumeration is expensive, or (ii) *PwA is fragile*, so enumeration collapses under EU pressure even if PwS is also weak. This distinction is visible in the map: **Longcat** and **Kimi-K2** sit in the high-SI, high-PwA region (Region III) and exhibit small k^* (early crossover), supporting the intended schema-advantage interpretation; in contrast, **Llama-4** crosses but lies near the low-capability region (Region II), so its crossover is better explained by PwA deteriorating under scaling than by strong PwS.

Overall, the stress test establishes that (i) scaling the enumerated interface alone is sufficient to trigger PwA degradation, and (ii) whether the crossover reflects a *meaningful* schema advantage depends critically on SI capability. This shifts the focus to a follow-up question: **what enables reliable schema instantiation in the first place.**

6 What Matters for Capable PwS Agents

We outline two findings about what does—and does not—shift the inflection left.

Long Reasoning: Beneficial, but Not Critical. Models with long-reasoning capabilities exhibit improved task performance, but this advantage does not translate to a greater aptitude for schema instantiation: the PwS–PwA gap on ALFWorld is not closed by enabling long reasoning (Appendix F). Thus, **we hypothesize that long-reasoning capability, while generally advantageous, is not the decisive factor in mastering PwS.**

A promising recipe: post-training for multi-turn tool use. A plausible contributor to stronger SI is the model's post-training methodology. Notably, technical reports for Kimi-K2 (Bai et al., 2025) and LongCat (MeituanLongCatTeam et al., 2025) describe extensive agentic post-training that incorporates multi-turn tool-use data (e.g., large collections of tool specifications and trajectories generated under diverse tool sets), often coupled with reinforcement learning; these models also perform strongly on tool-use benchmarks such as τ-bench (Yao et al., 2024), τ^2-bench (Barres et al., 2025), and ACEBench (Chen et al., 2025). We hypothesize that multi-turn tool-use training improves SI because it repeatedly requires the model to (i) choose among abstract tool schemas and (ii) populate structured arguments correctly to trigger execution—skills that closely overlap with template selection and argument grounding in PwS. While correlational, this evidence motivates post-training on multi-turn, tool-call–structured data as a promising direction for schema-based agents in real-world applications.

7 Conclusion

Through a cognitive-bandwidth lens, we contrast Planning with Actions (PwA), which pays a screening cost that rises with the enumerated list, and Planning with Schemas (PwS), which compresses the interface at the cost of schema-instantiation overhead. Across four environments and a controlled stress test, the location of the inflection between these regimes tracks two capability axes—PwA proficiency and SI capability—rather than long-reasoning depth, pointing to multi-turn tool-use post-training as a promising recipe for scalable schema-based agents.

Limitations

This work is restricted to four public, text-based simulators (TextCraft, WebShop, ALFWorld, SciWorld) and inference-only evaluation of off-the-shelf models; we make no claims about open-world deployment. The Cognitive Bandwidth Perspective is a qualitative lens, not a quantitative probe of model capacity, and the EU/SI proxies (cumulative context length, invalid/repetitive actions) are observable approximations rather than direct measurements of internal load. Simulator-based findings may not transfer to embodied or production settings, where any real-world use would require additional alignment, monitoring, and access control beyond what we evaluate here.

Ethics Statement

Our research does not involve human subjects or the collection of new personally identifiable information. We use only public, text-based simulators and off-the-shelf models for inference-only evaluation, and all prompts used in this study are included in the appendix; no harmful content is involved. All LLM evaluations were conducted via public APIs (OpenRouter), with a total experimental cost estimated at 5,000 USD; details on model selection are in Appendix D. We provide all prompt templates and system implementations in the appendix to facilitate full reproducibility, and all code and data will be publicly released to foster future research. All experiments in this work were conducted solely for research purposes and strictly adhere to the licensed usage policies of the respective datasets and models.

References

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, and 33 others. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, and 80 others. 2025. Kimi K2: open agentic intelligence. *CoRR*, abs/2507.20534.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. τ^2-bench: Evaluating conversational agents in a dual-control environment. *CoRR*, abs/2506.07982.

Chen Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Yuefeng Huang, Wulong Liu, Xinzhi Wang, Defu Lian, Baoqun Yin, Yasheng Wang, and Wu Liu. 2025. Acebench: Who wins the match point in tool learning? *CoRR*, abs/2501.12851.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready AI agents with scalable long-term memory. *CoRR*, abs/2504.19413.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 80 others. 2024. Deepseek-v3 technical report. *CoRR*, abs/2412.19437.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, and 1 others. 2025a. Towards general agentic intelligence via environment scaling. *arXiv preprint arXiv:2509.13311*.

Tianqing Fang, Zhisong Zhang, Xiaoyang Wang, Rui Wang, Can Qin, Yuxuan Wan, Jun-Yu Ma, Ce Zhang, Jiaqi Chen, Xiyun Li, Hongming Zhang, Haitao Mi, and Dong Yu. 2025b. Cognitive kernel-pro: A framework for deep research agents and agent foundation models training. *CoRR*, abs/2508.00414.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *CoRR*, abs/2503.09516.

Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, Hongxuan Lu, Tianrui Qin, Chenghao Zhu, Yi Yao, Shuying Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, and 11 others. 2025a. Chain-of-agents: End-to-end

agent foundation models via multi-agent distillation and agentic RL. *CoRR*, abs/2508.13167.

Zhiyu Li, Shichao Song, Chenyang Xi, Hanyu Wang, Chen Tang, Simin Niu, Ding Chen, Jiawei Yang, Chunyu Li, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, and 20 others. 2025b. Memos: A memory OS for AI system. *CoRR*, abs/2507.03724.

Zijun Liu, Zhennan Wan, Peng Li, Ming Yan, Ji Zhang, Fei Huang, and Yang Liu. 2025. Scaling external knowledge input beyond context windows of llms via multi-agent collaboration. *CoRR*, abs/2505.21471.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 13851–13870. Association for Computational Linguistics.

MeituanLongCatTeam, Bayan, Bei Li, Bingye Lei, Bo Wang, Bolin Rong, Chao Wang, Chao Zhang, Chen Gao, Chen Zhang, Cheng Sun, Chengcheng Han, Chenguang Xi, Chi Zhang, Chong Peng, Chuan Qin, Chuyu Zhang, Cong Chen, Congkui Wang, and 163 others. 2025. Longcat-flash technical report. *Preprint*, arXiv:2509.01322.

MiniMax, Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, Enwei Jiao, Gengxin Li, Guojun Zhang, Haohai Sun, Houze Dong, Jiadai Zhu, Jiaqi Zhuang, Jiayuan Song, and 71 others. 2025. Minimax-01: Scaling foundation models with lightning attention. *CoRR*, abs/2501.08313.

Siddharth Nayak, Adelmo Morrison Orozco, Marina Ten Have, Jackson Zhang, Vittal Thirumalai, Darren Chen, Aditya Kapoor, Eric Robinson, Karthik Gopalakrishnan, James Harrison, Anuj Mahajan, Brian Ichter, and Hamsa Balakrishnan. 2024. Long-horizon planning for multi-agent robots in partially observable environments. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

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, Mexico City, Mexico, June 16-21, 2024*, pages 4226–4252. Association for Computational Linguistics.

Shuofei Qiao, Runnan Fang, Ningyu Zhang, Yuqi Zhu, Xiang Chen, Shumin Deng, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024. Agent planning with world knowledge model. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Matthew Renze and Erhan Guven. 2024. Self-reflection in LLM agents: Effects on problem-solving performance. *CoRR*, abs/2405.06682.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. 2021. Alfworld: Aligning text and embodied environments for interactive learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7584–7600. Association for Computational Linguistics.

Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. 2023. Adaplanner: Adaptive planning from feedback with language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Ruoyao Wang, Peter A. Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11279–11298. Association for Computational Linguistics.

Zhaowei Wang, Wenhao Yu, Xiyu Ren, Jipeng Zhang, Yu Zhao, Rohit Saxena, Liang Cheng, Ginny Y. Wong, Simon See, Pasquale Minervini, Yangqiu Song, and Mark Steedman. 2025a. Mmlongbench: Benchmarking long-context vision-language models effectively and thoroughly. *CoRR*, abs/2505.10610.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Got-

tlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. 2025b. RAGEN: understanding self-evolution in LLM agents via multi-turn reinforcement learning. *CoRR*, abs/2504.20073.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Xin Guo, Dingwen Yang, Chenyang Liao, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. 2025. Agentgym: Evaluating and training large language model-based agents across diverse environments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 27914–27961. Association for Computational Linguistics.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, Songyang Gao, Lu Chen, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, Xipeng Qiu, Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. 2024. Agentgym: Evolving large language model-based agents across diverse environments. *CoRR*, abs/2406.04151.

Baixuan Xu, Chunyang Li, Weiqi Wang, Wei Fan, Tianshi Zheng, Haochen Shi, Tao Fan, Yangqiu Song, and Qiang Yang. 2025a. Towards multi-agent reasoning systems for collaborative expertise delegation: An exploratory design study. *CoRR*, abs/2505.07313.

Baixuan Xu, Weiqi Wang, Haochen Shi, Wenxuan Ding, Huihao Jing, Tianqing Fang, Jiaxin Bai, Xin Liu, Changlong Yu, Zheng Li, Chen Luo, Qingyu Yin, Bing Yin, Long Chen, and Yangqiu Song. 2024. MIND: multimodal shopping intention distillation from large vision-language models for e-commerce purchase understanding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 7800–7815. Association for Computational Linguistics.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025b. A-MEM: agentic memory for LLM agents. *CoRR*, abs/2502.12110.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. 2025. Memagent: Reshaping long-context LLM with multi-conv rl-based memory agent. *CoRR*, abs/2507.02259.

Siyu Yuan, Zehui Chen, Zhiheng Xi, Junjie Ye, Zhengyin Du, and Jiecao Chen. 2025. Agent-r: Training language model agents to reflect via iterative self-training. *CoRR*, abs/2501.11425.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ınftybench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15262–15277. Association for Computational Linguistics.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. 2025. RLVMR: reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. *CoRR*, abs/2507.22844.

Jun Zhao, Can Zu, Hao Xu, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang.

2024. Longagent: Scaling language models to 128k context through multi-agent collaboration. *CoRR*, abs/2402.11550.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. 2025. MEM1: learning to synergize memory and reasoning for efficient long-horizon agents. *CoRR*, abs/2506.15841.

Appendix

A Action Schema Lists for Different Environments

A.1 AlfWorld

Action Schema List for ALFWorld

```
Alfworld_Template = [
``close [OBJ]'',
``cool [OBJ] with [OBJ]'',
``heat [OBJ] with [OBJ]'',
``examine [OBJ]'',
``go to [OBJ]'',
``inventory'',
``look'',
``open [OBJ]'',
``put [OBJ] in/on [OBJ]'',
``take [OBJ] from [OBJ]'',
``clean [OBJ] with [OBJ]'',
]
```

A.2 SciWorld

Action Schema List for SciWorld

```
Sciworld_Template = [
``open [OBJ]'',
``close [OBJ]'',
``activate [OBJ]'',
``deactivate [OBJ]'',
``connect [OBJ] to [OBJ]'',
``disconnect [OBJ]'',
``use [OBJ]'',
``use [OBJ] on [OBJ]'',
``look around'',
``look at [OBJ]'',
``look in [OBJ]'',
``read [OBJ]'',
``move [OBJ] to [OBJ]'',
``pick up [OBJ]'',
``put down [OBJ]'',
``pour [OBJ] into [OBJ]'',
``dunk [OBJ] into [OBJ]'',
``mix [OBJ]'',
``go to [OBJ]'',
``eat [OBJ]'',
``flush [OBJ]'',
``focus on [OBJ]'',
``wait'',
``wait1'',
``task'',
``inventory''
]
```

B Trajectory Analysis Prompt

B.1 Action Analysis for Repetitive Actions

This analysis is conducted based on the analysis result of Kimi-K2, a powerful LLMs with 1T parameters effectively trained for agentic tasks.

System Prompt For Action Analysis

```
ACTION_ANALYSIS_SYS_PROMPT = You are an expert
↪  trajectory analyst.
Your task is to identify repetitive/meaningless
↪  actions in an agent's interaction history.

**Input:**
- A trajectory dictionary mapping integer step indices
↪  to interaction strings.
Each interaction string has the format:
  "Human: <observation>\n Agent: <thought and action>"
```

```
**Definition of Repetitive Action:**
- **Meaningless repeated actions** that don't advance the task
- Must meet these criteria:
  1. The action is identical or functionally equivalent to a previous action
  2. The action occurs after the environment has provided negative feedback
  3. The action doesn't yield new information or state change
  4. The action doesn't contribute to task progression
- Examples:
  - Repeatedly opening the same empty drawer after being told it's empty
  - Moving between the same locations without purpose
  - Checking inventory multiple times without new items
  - Retrying an action immediately after failure feedback

**Task:**
- Analyze the trajectory and identify step indices for repetitive actions
- Output a JSON dictionary with exactly one key:
  - "repetitive_action": sorted list of step indices with repetitive actions

**Output Rules:**
- Only include step indices where the action clearly meets the criteria
- Sort indices in ascending order
- If no instances are found, return an empty list
- Output must be valid JSON with no additional text

**Output Schema:**
{
  "repetitive_action": [<int>, <int>, ...]
}

**Analysis Guidelines:**
1. Focus on patterns:
   - Look for repeated action patterns, not necessarily consecutive
   - Consider actions that are functionally equivalent (e.g., "open cabinet 1" and "open cabinet1")
   - Pay attention to environment feedback (e.g., "nothing happens", "still empty")

2. Be practical:
   - Include actions that show clear lack of progress
   - Ignore repetitions that are part of necessary sequences
   - Consider the context of the task

3. Key indicators:
   - Environment feedback indicating no change
   - Agent thoughts showing frustration or confusion
   - Actions that could be optimized (e.g., moving between same locations)

**Example:**
Input:
{
  "0": "Human: Kitchen with cabinet\n Agent: Open cabinet",
  "1": "Human: Cabinet is empty\n Agent: Close cabinet",
  "2": "Human: Cabinet closed\n Agent: Open cabinet",
  "3": "Human: Cabinet still empty\n Agent: Open cabinet",
  "4": "Human: Cabinet remains empty\n Agent: Look for key",
  "5": "Human: Key on counter\n Agent: Take key",
  "6": "Human: Key taken\n Agent: Open cabinet with key",
  "7": "Human: Cabinet opens\n Agent: Take microwave",
  "8": "Human: Microwave taken\n Agent: Put microwave in fridge",
  "9": "Human: Microwave doesn't fit\n Agent: Put microwave in fridge",
  "10": "Human: Still doesn't fit\n Agent: Try putting microwave in fridge"
}

Output:
{
  "repetitive_action": [2, 3, 9, 10]
}

Explanation:
- Step 2: Repetitive (reopening cabinet after knowing it's empty)
- Step 3: Repetitive (consecutive identical action after known empty)
- Step 9: Repetitive (retrying after failure feedback)
- Step 10: Repetitive (functionally equivalent retry after failure)
- Step 6: Not repetitive (new context with key)
```

System Prompt For Action Analysis

```
ACTION_ANALYSIS_USER_PROMPT = Analyze the following trajectory
and identify invalid actions and repetitive actions.

Trajectory:
{trajectory}

Return JSON output exactly as specified.
```

B.2 Action Analysis for Invalid Actions

The analysis for invalid actions is conducted via on rule-based filtering. Specifically, we count invalid action frequency by detecting the signature response from the environment API which indicates invalid. For example, "nothing happens." for API response in ALFWorld and "No known action matches that input."

C Evaluation Environments and Experiment Protocols

Our experiments are conducted on a suite of four environments, selected to represent a progression of increasing complexity. The environments are modified from AgentGym (Xi et al., 2025).

Textcraft (Prasad et al., 2024): TextCraft is a text-only simulation environment designed to evaluate an agent's capacity for compositional reasoning and planning. Agents are tasked with crafting a specific target item by issuing text commands, with tasks requiring a variable number of steps to complete. The action space is restricted to three commands: craft, get, and inventory, forcing the agent to learn the hierarchical structure of crafting recipes.

Webshop (Yao et al., 2022): WebShop is an interactive, simulated e-commerce environment designed to evaluate an agent's ability to act on compositional, natural language instructions. Given a detailed product description, the agent must navigate the website, search for products, and select an item that satisfies as many specified attributes

as possible. The agent's action space consists of searching with free-form text and clicking buttons on the webpage. A final reward is calculated based on the degree of match between the purchased item and the user's initial instruction

ALFWorld (Shridhar et al., 2021): ALFWorld is a simulated household environment designed to benchmark an agent's ability to execute high-level tasks. Success in this environment requires the agent to perform multi-step reasoning, navigate and explore various rooms, and interact with objects. The action space supports a range of commands, including object manipulation and navigation. The environment provides deterministic feedback on the outcome of each action based on a predefined logical framework that governs the world state.

SciWorld (Wang et al., 2022): ScienceWorld is an interactive text-based benchmark created to evaluate an agent's scientific reasoning capabilities. The environment features a diverse set of 30 task types derived from a standard elementary science curriculum, such as correctly using measurement instruments or conducting simple mechanics experiments. Following each action executed by the agent, the environment simulator provides an updated observation, describing the effects and changes within the simulated world.

Experiment Protocols. The prompts for interacting with environment API are fully aligned with AgentGym (Xi et al., 2025) with minimal modification from self-deployed models to model API calling.

D Model selection

Our study conducts analyses across two distinct classes of models: general-purpose large language models and specialized large reasoning models. For the LLM category, we select a suite of state-of-the-art models representing a spectrum of scales and architectures: Qwen2.5-7B (Yang et al., 2024), DeepSeek-V3 (DeepSeek-AI et al., 2024), Kimi-K2 (Bai et al., 2025), GPT-4.1, Claude and so on. This selection is curated to capture performance trends across models with varying parameter counts. For the LRM category, we chose DeepSeek-R1 (DeepSeek-AI et al., 2025) and Longcat (MeituanLongCatTeam et al., 2025), as they are prominent examples of recent models specifically engineered for complex tasks.

Full list of selected models are provided below.

Large Language Models without Long Reasoning Capability

Opensourced: Qwen2.5-7B (Yang et al., 2024)
Qwen3-235B-A22B-NonThinking (Yang et al., 2025)
Llama4-Scout
DeepSeek-V3 (DeepSeek-AI et al., 2024)
Kimi-K2 (Bai et al., 2025)
Minimax-01 (MiniMax et al., 2025)
Proprietary
GPT-4.1-mini
GPT-4.1
Gemini-2.0-flash (Anil et al., 2023)

Large Reasoning Models

Opensourced: DeepSeek-R1 (DeepSeek-AI et al., 2025)
Seed-OSS-36B
Qwen3-235B-A22B-Thinking
LongCat (MeituanLongCatTeam et al., 2025)
Proprietary
GPT5-mini
Gemini-2.5-Flash
Claude-4.0-Sonnet

E Comprehensive Failure Actions Analysis

F Thinking vs. Non-Thinking Comparison

Table 3 reports thinking (T) vs. non-thinking (NT) results under PwA (Action) and PwS (Schema) on ALFWorld and SciWorld for DeepSeek and Qwen3. The per-column $\Delta_T = T - NT$ shows that enabling long-reasoning typically improves task performance for both paradigms, but the per-column $\Delta_p = \text{PwS} - \text{PwA}$ shows that the PwS – PwA gap on ALFWorld remains large under thinking and does not systematically close, supporting the claim in Section 6 that long-reasoning is not the decisive factor for mastering PwS.

Model	Mode	ALFWorld			SciWorld		
		NT	T	Δ_T	NT	T	Δ_T
DPSK	PwA	50.0	65.0	+15.0	22.0	48.4	+26.4
	PwS	7.0	14.0	+7.0	25.5	46.9	+21.4
	Δ_p	-43.0	-51.0	–	+3.5	-1.5	–
Qwen3	PwA	58.0	47.0	-11.0	43.3	47.7	+4.4
	PwS	5.0	23.0	+18.0	62.2	67.3	+5.1
	Δ_p	-53.0	-24.0	–	+18.9	+19.6	–

Table 3: Thinking (T) vs. Non-Thinking (NT) results under PwA (Action) and PwS (Schema) on ALFWorld and SciWorld. $\Delta_T = T - NT$. $\Delta_p = \text{PwS} - \text{PwA}$ (computed per column).

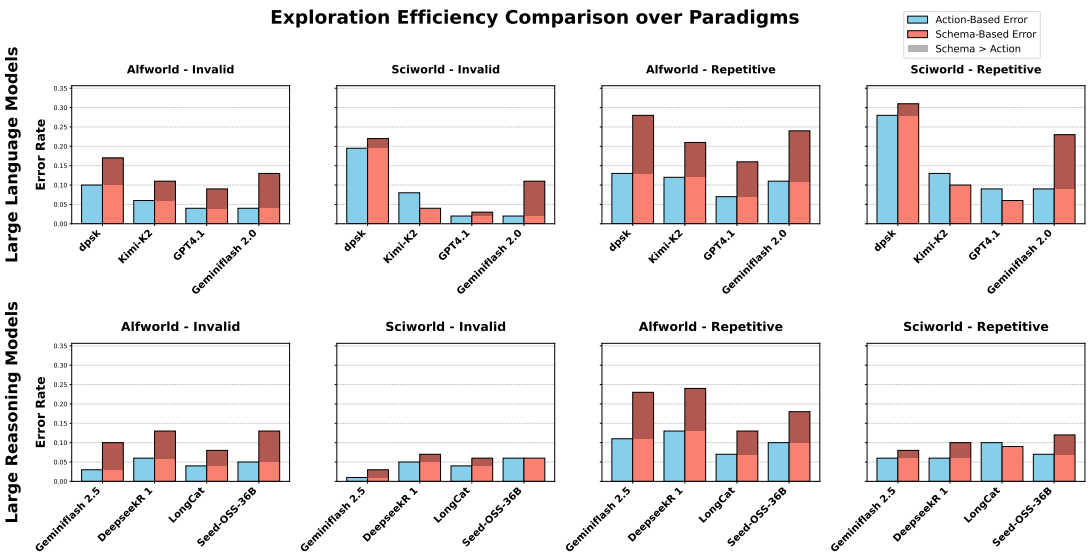

Figure 5: Comparison of agent behaviors in ALFWorld and SciWorld, two environments situated on either side of the observed complexity inflection point.

Environment	Model	Prompt in Paper	Alternative Prompt
ALFWorld	DeepSeek-V3	7.0	5.0
	Llama-4-Scout	12.0	9.0
	Kimi-K2	45.0	51.0
	DeepSeek-R1	14.0	9.0
SciWorld	DeepSeek-V3	25.5	28.4
	Llama-4-Scout	41.4	32.2
	Kimi-K2	52.9	49.2
	DeepSeek-R1	46.9	43.6

Table 4: Comparison under ALFWorld and SciWorld between different schema design.

G Robustness under Schema Variation

H Noisy Action List for Cognitive Load Attack

In this section, we demonstrate some schema-aligned simplified examples on the distractors to be injected into ALFWorld action list.

Invalid OBJ Noisy Actions

```
ALFWorld_Distractors = [
    "open portal",
    "open mirror",
    "open book",
    "open map",
    "open pyramid",
    "open record",
    "open portrait",
    "open vehicle",
    "open council",
    "open grove",
    "open ship",
    "open library",
    "close portal",
    "close mirror",
    "close book",
    "close map",
    "close pyramid",
    "close record",
    "close portrait",
    "close vehicle",
    "close council",
    "examine feather",
    "examine crystal",
    "examine mane",
    "examine stone",
    "examine sensor",
    "examine rune",
    "examine balloon",
    "examine charm",
    "examine star",
    "examine wing",
    "examine mirror",
    "examine book",
    "examine map",
    "examine record",
    "examine portrait",
    "examine pyramid",
    "examine vehicle",
    "examine council",
    "examine grove",
    "examine library",
    "examine ship",
    "examine algorithm",

    # go to
    "go to grove",
    "go to portal",
    "go to plane",
    "go to realm",
    "go to world",
    "go to border",
    "go to library",
    "go to garden",
    "go to council",
    "go to ocean",

    # put [OBJ] in/on [OBJ]
    "put feather in pyramid",
    "put crystal in library",
    "put book on portrait",
    "put map on council",
    "put charm in vehicle",
    "put mirror on pyramid",
    "put stone in grove",
    "put mane on portrait",
    "put star on garden",
    "put wing on council",
    "put record in library",
    "put form on portrait",
    "put song in library",
```

```
    "put algorithm in library",
    "put tree in garden",

    # take [OBJ] from [OBJ]
    "take book from library",
    "take map from council",
    "take feather from pyramid",
    "take crystal from garden",
    "take charm from vehicle",
    "take mirror from portrait",
    "take stone from grove",
    "take wing from council",
    "take record from library",
    "take star from world",

    # clean [OBJ] with [OBJ]
    "clean mirror with feather",
    "clean pyramid with feather",
    "clean portrait with mane",
    "clean book with charm",
    "clean record with feather",
    "clean council with cloth",
    "clean library with cloth",
    "clean vehicle with feather",
    "clean grove with cloth",

    # heat [OBJ] with [OBJ]
    "heat book with incense",
    "heat crystal with incense",
    "heat stone with incense",
    "heat mirror with incense",
    "heat record with incense",
    "heat pyramid with incense",

    # cool [OBJ] with [OBJ]
    "cool crystal with wave",
    "cool book with radiation",
    "cool mirror with wave",
    "cool stone with radiation",
    "cool pyramid with wave",
    "cool portrait with radiation",

    # extra plain variants
    "open council",
    "close council",
    "examine message",
    "examine pattern",
    "examine aura",
    "examine vision",
    "examine orb",
    "examine fabric",
    "examine machine",
    "examine tree",
    "examine path",
    "examine portrait",
    "examine riddle",
    "examine math",
    "examine wave",
    "go to territory",
    "go to library",
    "go to realm",
    "put book in library",
    "put map in library",
    "put song in library",
    "put form in library",
    "put algorithm on council",
    "put feather on portrait",
    "put crystal on council",
    "take charm from portrait",
    "take balloon from garden",
    "take stone from council",
    "take vision from library",
    "take song from library",
    "clean portrait with feather",
    "clean book with mane",
    "clean pyramid with charm",
]
```

I Cognitive-Load Stress Test Results

J Long Horizon Task Formulation

Formally, an agent in an interactive environment could be modeled as decision-making under partial observation. Following existing works (Yuan et al., 2025; Song et al., 2024), these tasks could be formulated as **Partially Observable Markov Decision Process (POMDP)** which could be expressed in form of $(\mathcal{U}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{R})$. To be detailed, the $\mathcal{U}$ stands for the task description and the relevant requirements. $\mathcal{S}$ is the state space, $\mathcal{A}$ represents the action space of the agent, and $\mathcal{O}$ is the observation space. $\mathcal{T}$ stands for the transition function where $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ and $\mathcal{T}$ is determined by the interactive environment. $\mathcal{R}$ represents the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ which indicates the final reward of the agents movements. Since our experiments are natural language based, $\mathcal{U}, \mathcal{S}, \mathcal{A}$ and $\mathcal{O}$ are all in natural language form. At each timestamp t, the trajectory history is denoted as:

$$\tau_t = (u, a_1, o_1, \ldots, a_t, o_t) \sim \pi_\theta(\tau_t|u) \quad (1)$$

where $a_i \in \mathcal{A}, o_i \in \mathcal{O}$ stand for the action and the observation after the action a_t is executed at timestamp t. The probability of agent with parameter θ generating τ would be:

$$\pi_\theta(\tau|u) = \prod_{j=1}^{n} \pi_\theta(a_j|i, a_1, o_1, \ldots, o_{j-1}) \quad (2)$$

where n is the trajectory length. Finally, a final reward r is computed, with 1 indicating success of the task, 0 stands for the failure of the task and other values for partial completion of the task.

K The Use of Large Language Models

In this paper, the LLMs serves as a writing assistant to help polish the content.

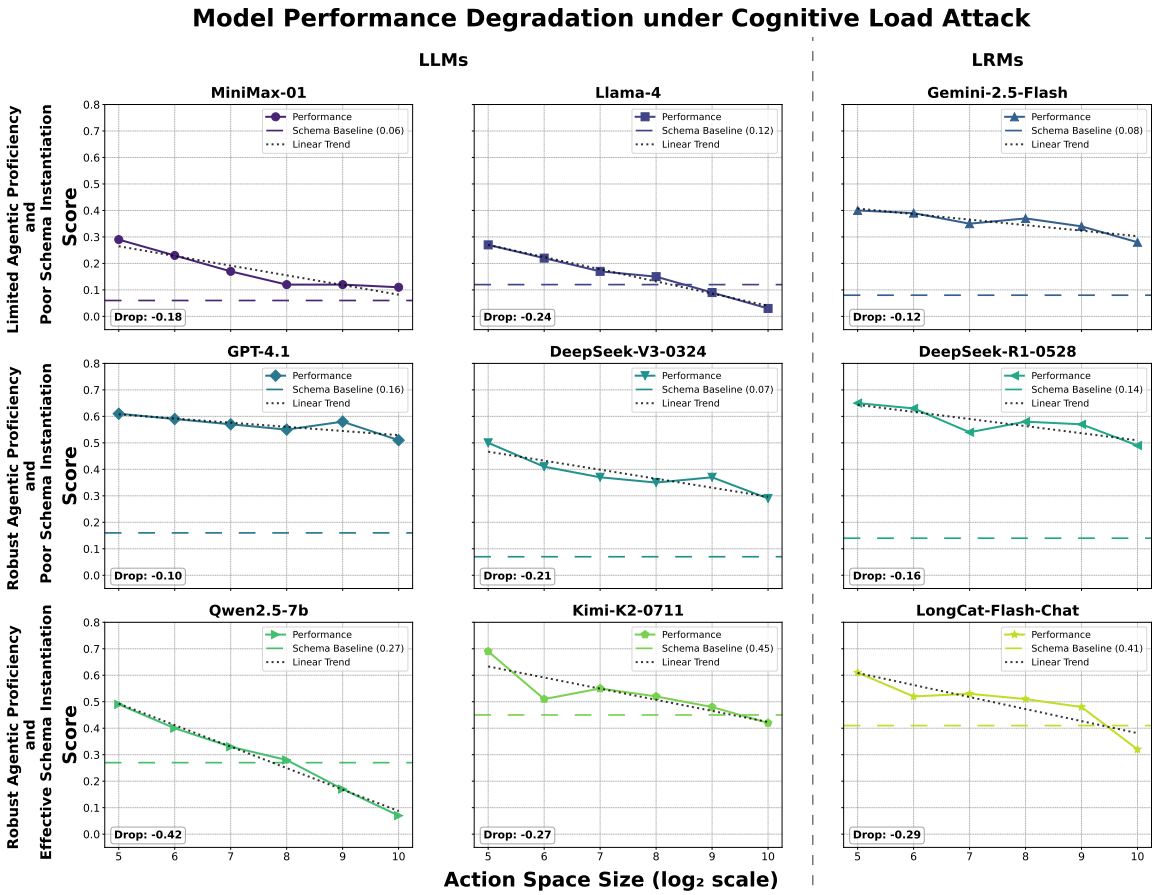

Figure 6: Per-model PwA success rate versus distractor-injected action-list size k ($\log_2$ scale) across the cognitive-load stress test on ALFWorld. Each panel shows one model; dashed lines mark the PwS baseline. The crossover k^* of each model is the smallest k where PwA falls below its PwS baseline.