PROCEDURAL EXPOSURE BIAS IN SKILL-AUGMENTED LANGUAGE MODEL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

External skills provide language-model agents with reusable procedural knowledge, yet how models use such knowledge after it is supplied remains poorly understood. In this paper, we identify **Procedural Exposure Bias (PEB)**, a systematic tendency of language models to increasingly favor an action as its associated procedural pattern is repeatedly exposed in a skill, even when the interaction state and the action's relevance remain unchanged. We find that PEB is strongly dependent on how procedural knowledge is represented: presenting the same procedural content visually substantially attenuates the exposure-induced shift in action preference compared with direct textual presentation. Further analysis shows that this attenuation primarily arises from visual encoding rather than structured visual layout. Although repeated procedural signals remain recoverable from Visual representations, they exert a weaker influence on downstream action prediction, revealing a distinction between signal accessibility and its use as action evidence. This representation-dependent effect also extends to executable action selection, where increasing procedural exposure more strongly favors the corresponding action under Text than Visual. Layer-wise analyses show that exposure-dependent action evidence becomes progressively more pronounced toward the model output, with stronger evidence under Text, while activation interventions reveal architecture-specific roles of internal components in transmitting or counteracting the resulting preference shift. Finally, complete-agent evaluations show that these representation differences persist during end-to-end interaction, with visual skills yielding higher overall task success than textual skills across the evaluated models. Together, these findings establish skill representation as an important factor governing how procedural knowledge is translated into agent decisions. All code and datasets will be released on GitHub.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated strong emergent capabilities (Wu & Lo, 2025; Dong et al., 2025) and are increasingly deployed as agents in interactive environments (Park & Kwon, 2026; Xi et al., 2025). In these settings, external skills provide reusable procedural knowledge, including execution steps, applicability conditions, and operational constraints (Jiang et al., 2026a; Xiong et al., 2026). A typical skill-augmented agent constructs or extracts skills into a reusable library, retrieves or selects relevant skills according to the task, and provides them to the model to guide planning and action prediction (Xia et al., 2026; Zhao et al., 2026; Fang et al., 2026; Wang et al., 2026b). Such procedural guidance equips agents with reusable experience and can substantially facilitate the completion of complex, long-horizon tasks.

Prior work has extensively explored skill construction, extraction, representation, retrieval, and selection for LLMs (Wang et al., 2026b; Fang et al., 2026). These efforts make procedural knowledge accessible to the LLM, yet accessibility does not necessarily guarantee effective utilization (Li et al., 2026a; Liu et al., 2026). Once a skill is provided, the LLM must still determine which procedures are applicable to the current interaction state and which actions should be executed next (Han et al., 2026; Jiang et al., 2026b). Importantly, a procedure can be relevant to the overall task while containing actions that are irrelevant to the current state. This raises a fundamental question: *does the way procedural actions are exposed within a skill systematically affect how an LLM chooses actions, even when their actual relevance remains unchanged?* Although previous studies have shown

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

that LLM predictions can be influenced by irrelevant or repeatedly presented context (Yang et al., 2025; Niu et al., 2025; Geng et al., 2025), how such exposure effects manifest in procedural skill utilization remains poorly understood, due to the entangled effects of skill content and the model's inherent behavioral biases.

In this paper, we uncover a systematic phenomenon that we term **Procedural Exposure Bias (PEB)**. PEB refers to the tendency of an LLM to increasingly favor an action as its corresponding procedural pattern is repeatedly exposed in a skill, even when the interaction state and the action's relevance remain unchanged. To isolate this effect, we conduct controlled experiments by manipulating the n-gram exposure frequency of action-related patterns while keeping the underlying task state and action relevance fixed. We observe that increasing the exposure of a procedural action consistently shifts the model's action preference toward that action, revealing that LLMs can be biased by the frequency with which procedural actions are presented rather than solely by their state-dependent applicability. These findings suggest that skill utilization is not merely a matter of retrieving relevant procedures: the internal representation of a skill can itself introduce systematic biases into downstream action selection. Motivated by this observation, we further explore whether changing the modality of skill representation can reduce such exposure-induced bias. Specifically, we represent procedural skills visually and provide them to Vision-Language Models (VLMs), replacing explicit token-level repetition with visual representations of the same procedural information. Unlike textual skills, where repeatedly appearing action phrases directly increase lexical exposure, visual encoding transforms the procedure into distributed representations over image patches. This representation provides a natural way to decouple procedural information from its surface-level repetition.

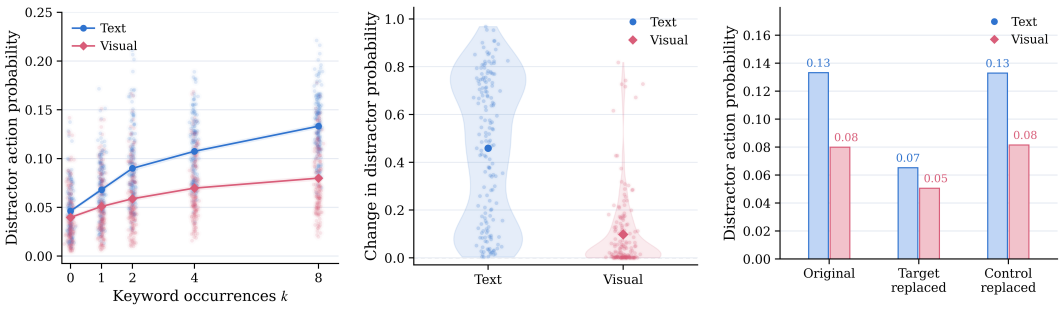
Empirically, we find that visualized skills substantially attenuate the preference shift induced by repeated procedural actions, revealing a strong dependence of PEB on skill representation. By disentangling visual encoding from spatial layout, we find that most of this attenuation already emerges when the same textual skill is rasterized into an image, while additional structural formatting contributes only marginally. Importantly, direct signal-recovery probes show that the repeated procedural information remains recoverable under visual encoding, indicating that the representation difference lies not simply in whether the signal is accessible, but in how strongly it is translated into downstream action evidence. Internal analyses further show that this exposure-dependent action evidence develops progressively across model depth, while matched component interventions reveal architecture-dependent roles in transmitting or counteracting the resulting preference shift. Finally, we connect these controlled effects to agent behavior. Across the three benchmarks, Visual achieves a higher macro-average success rate than Text for all four evaluated backbones: Qwen3.5-9B and Qwen3.5-27B (Qwen Team, 2026a), Gemma-4-31B-it (Gemma Team et al., 2026), and Kimi-K2.5 (Kimi Team et al., 2026). Controlled generation experiments further show that increasing procedural exposure can alter executable action selection, with a substantially stronger effect under Text than Visual.

2 Uncovering Procedural Exposure Bias in Skill-Augmented LLM Planning

In this section, we investigate whether repeated exposure to action-related procedural patterns in a supplied skill can influence action prediction independently of their relevance to the current decision. We first formalize *procedural exposure* and construct a controlled probe that varies exposure while keeping the decision context and action applicability fixed. We then examine how the model's action preference changes as exposure increases and determine whether the observed effect can be attributed to the repeated action-related signal.

2.1 Procedural Exposure Bias Probing

To probe procedural exposure bias, we focus on *action-bearing procedural elements*, such as "open the microwave" or "put the object into the microwave", which specify executable operations within a skill describing how a task should be performed. The same action-associated procedural pattern may occur multiple times within a supplied skill, potentially exposing the model to the same action-related signal repeatedly.



(a) Exposure-Dependent Distractor Preference.

(b) Paired Changes in Distractor Preference.

(c) Keyword-Replacement Controls.

Figure 1: Procedural Exposure Bias in Qwen3.5-9B Under Text Input and Image Rendering. Figure 1a reports distractor preference across exposure levels; Figure 1b, paired preference changes from $k = 0$ to $k = 8$; and Figure 1c, the original, target-replacement, and frequency-matched control conditions.

To operationalize procedural exposure, for decision step i and exposure level k, let $\mathcal{C}_i^{(k)}$ denote the supplied skill content, and let $w(a)$ denote the target keyword associated with action a for the exposure manipulation. We define the procedural exposure frequency of action a as

$$\mathcal{F}_i^{(k)}(a) = \mathrm{Count}\left(w(a), \mathcal{C}_i^{(k)}\right), \tag{1}$$

where $\mathrm{Count}(\cdot)$ measures the number of occurrences of the target keyword in the supplied skill. In our controlled probe, the target keyword is an object-related noun associated with the action. For example, for the action "open microwave," the target keyword is $w(a) =$ "microwave".

To isolate the effect of exposure from action relevance, we sample intermediate decision steps from successful ALFWorld trajectories. At each step, we fix the observation history and the correct action a_i^C from the reference trajectory, and introduce a distractor action a_i^D that is relevant to the overall task but inapplicable to the current state. We construct skill variants in which the distractor keyword $w(a_i^D)$ occurs $k \in \{0, 1, 2, 4, 8\}$ times. The remaining insertion positions are filled with content that does not contain the target keyword, while keeping the total word count fixed across exposure levels. Details of decision-state construction, distractor and keyword selection, and exposure-controlled skill construction are provided in Appendix A.2.1 and Appendix A.2.2.

We quantify model support by scoring the same fixed candidate actions rather than allowing the model to freely generate the next action. For a candidate action $a = (a_1, \ldots, a_{|a|})$, we compute its length-normalized token log-probability:

$$S_i(a; k) = \frac{1}{|a|} \sum_{t=1}^{|a|} \log p_\theta \left(a_t \mid x_i^{(k)}, a_{<t}\right), \tag{2}$$

where $|a|$ is the number of action tokens and $x_i^{(k)}$ denotes the complete input at exposure level k, including the fixed scoring prefix. Here, θ denotes model parameters, a_t is the t-th action token, and $a_{<t}$ denotes the preceding action tokens. A higher score indicates stronger model support for the candidate action under the same decision context. Relative action preference is obtained by normalizing candidate scores across the fixed set, including the distractor.

As a reference condition, we also render the same skill content as images and provide it through the model's visual encoder. Text and Visual use the same procedural content and exposure levels, differing only in how the supplied skill is represented. This comparison allows us to examine whether the relationship between procedural exposure and action prediction changes across representations. Further details of candidate-action scoring and image rendering are provided in Appendix A.2.3 and Appendix A.2.4.

2.2 Procedural Exposure is Associated with Action Prediction

To characterize how procedural exposure influences action prediction independently of state relevance, we apply the controlled probe to Qwen3.5-9B across 165 ALFWorld decision steps. Figure 1

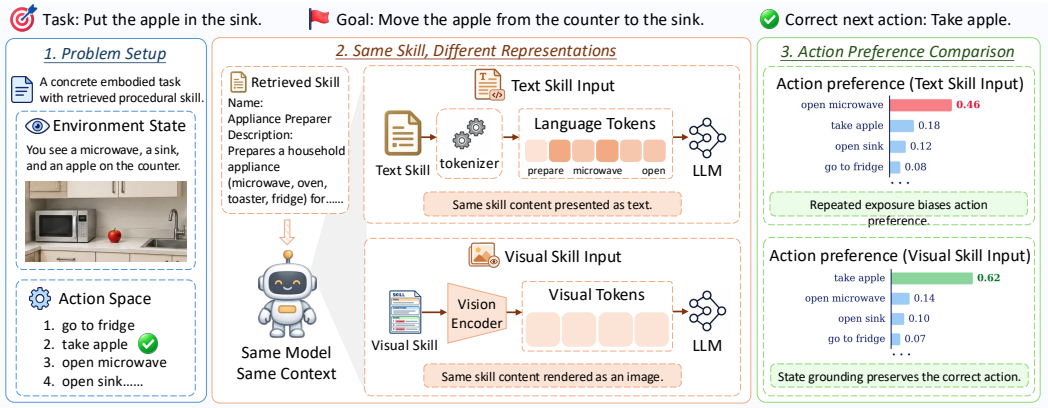


Figure 2: Overview of the Controlled Comparison Between Text and Visual Skill Representations.

summarizes this analysis along three dimensions: its dependence on exposure frequency, its consistency across decision states, and its specificity to the repeated action-related keyword.

As illustrated in Figure 1a, increasing the frequency of the distractor keyword systematically elevates the preference of the model for the corresponding distractor under direct text input, even while the decision state and the actual applicability of the distractor remain strictly unchanged. We formally define this exposure-dependent increase in support for a state-inapplicable action as **Procedural Exposure Bias (PEB)**. Furthermore, the results demonstrate that this increase is substantially attenuated when the skill is presented as an image, indicating that the magnitude of PEB is highly sensitive to the representational modality of the procedural content.

We further investigate this representation discrepancy at the granularity of individual decision states in Figure 1b. By comparing the zero-exposure ($k = 0$) and high-exposure ($k = 8$) conditions, we observe that direct text input induces substantial positive shifts in distractor preference across the entire state distribution. Conversely, the corresponding changes under image input remain tightly concentrated near zero. This distribution confirms that the dampened exposure effect under visual rendering is a widespread structural behavior broadly shared across decision states, rather than an artifact driven by a small subset of atypical examples.

To test whether the observed preference shift specifically depends on the repeated target keyword, we compare the original condition with two matched replacement controls. As shown in Figure 1c, replacing the target keyword substantially reduces distractor preference, whereas replacing a frequency-matched non-target word produces little change. This contrast indicates that the preference shift is specifically associated with repeated exposure to the target action-related keyword rather than lexical replacement in general. The reduction induced by target replacement is also smaller under Visual than under Text, consistent with the weaker exposure sensitivity observed under visual encoding. Detailed configurations of the replacement controls are provided in Appendix A.2.5. Additional results for Qwen3.5-27B and Gemma-4-31B-it are provided in Appendix A.2.6.

3 Methodology

Building on the empirical observation in Section 2.2 that visual representation attenuates PEB, we investigate how repeated procedural content enters action prediction and how the same content is processed through different representation pathways. We first formulate procedural exposure in autoregressive action prediction and characterize how repeated contextual evidence can contribute through Transformer attention aggregation. We then examine how the same procedural content is represented under Text and Visual input, yielding different pathways through which repeated evidence reaches action prediction. Figure 2 summarizes this controlled comparison.

3.1 Procedural Exposure in Autoregressive Action Prediction

To characterize how procedural exposure influences action prediction, we consider the autoregressive generation process, motivated by recent evidence that repeated contextual information can sys-

tematically alter language model reasoning (Yu et al., 2025). For a complete input $x_i^{(k)}$ at decision step i and exposure level k, the probability of a candidate action $a = (a_1, \ldots, a_{|a|})$ factorizes as:

$$p_\theta\left(a \mid x_i^{(k)}\right) = \prod_{t=1}^{|a|} p_\theta\left(a_t \mid x_i^{(k)}, a_{<t}\right). \tag{3}$$

Under the controlled setting established in Section 2.1, the interaction history and candidate actions remain unchanged across exposure levels. Varying k therefore changes only the frequency with which the target procedural pattern appears in the context, while keeping the current interaction state and the relevance of each candidate action fixed.

Repeated procedural information can enter Transformer (Vaswani et al., 2017) computation through attention aggregation. At a generation position t, the attention output aggregates information from preceding context positions as:

$$\mathbf{o}_t = \sum_{j=1}^{N} \alpha_{t,j} \mathbf{v}_j, \tag{4}$$

where N is the number of visible context positions when predicting a_t, $\alpha_{t,j}$ is the attention weight assigned to position j, $\mathbf{v}_j$ is its value vector, and $\mathbf{o}_t$ is the resulting attention output.

For the action-associated keyword under manipulation, increasing the exposure level k introduces additional occurrences of the same procedural pattern into the model context. Each occurrence occupies a distinct context position whose representation can contribute to the attention aggregation used for subsequent action prediction. Thus, increasing k increases the number of context positions carrying related procedural evidence, while leaving the interaction state and the applicability of the corresponding action unchanged. Importantly, this effect does not require any individual occurrence to receive a larger attention weight: the exposure manipulation changes the number of potential contributors to the aggregation rather than directly increasing the weight of the action occurrence.

3.2 Reducing Procedural Exposure Bias through Visual Skill Encoding

To investigate why identical procedural exposure can produce different effects under different representations, we examine how the supplied skill is encoded before contributing to action prediction under Text and Visual input. We keep the procedural content and exposure frequency consistent across the two settings, while varying how the skill is encoded and delivered to the language model.

For the *Text*-based encoding, the skill is directly tokenized and inserted into the language model context. Repeating an action-associated keyword therefore introduces additional occurrences of the corresponding lexical pattern at distinct token positions. These positions directly participate in the Transformer computation described in Eq. 4, allowing repeated procedural evidence to contribute to action prediction through the language context. For the *Visual*-based encoding, the same procedural content is first rendered into an image. Following the standard Vision Transformer formulation (Dosovitskiy et al., 2021), the rendered image is divided into local patches, converted into patch-level representations, and processed by the model's vision encoder before being passed to the language model. Consequently, repeated procedural content is no longer introduced as repeated lexical token occurrences in the language context. Instead, each rendered occurrence is encoded through the image patches covering its spatial region and transformed by the visual encoding pathway before contributing to downstream action prediction. The two settings thus preserve the same procedural content and exposure frequency while routing the corresponding procedural evidence through different intermediate representations before reaching the language model.

4 Experimental Methodology

We evaluate skill representation through controlled procedural-exposure experiments and complete-agent interaction. This section summarizes the shared experimental setup; detailed constructions, scoring procedures, prompts, and additional implementation details are provided in Appendix A.

Benchmarks and Models. We evaluate on ALFWorld (Shridhar et al., 2021), ScienceWorld (Wang et al., 2022), and WebShop (Yao et al., 2022). ALFWorld is additionally used for the controlled

Table 1: Main results on ALFWorld, ScienceWorld, and WebShop under the default skill setting. **Score** denotes the average task score on WebShop, **Succ.** denotes the success rate (%), and **Steps** denotes the average number of interaction steps. **Overall** reports the macro-average success rate across the three benchmarks. Higher scores and success rates are better (↑). For interaction steps, shorter trajectories are preferred when task success is comparable (↓). The best result for each metric within each backbone is highlighted in **bold**.

Setting	Overall	ALFWorld		ScienceWorld		WebShop		
	Macro Succ. ↑	Succ. ↑	Steps ↓	Succ. ↑	Steps ↓	Score ↑	Succ. ↑	Steps ↓
Qwen3.5-9B								
No Skill	16.21	15.67	27.59	18.96	17.04	38.52	14.00	13.52
Text Skill	40.70	66.42	18.28	42.18	**15.36**	39.53	13.50	12.85
Visual Skill	**44.06**	**70.15**	**17.75**	**45.02**	15.95	**40.77**	**17.00**	**12.79**
Qwen3.5-27B								
No Skill	41.33	58.96	19.01	45.02	15.83	49.27	20.00	12.10
Text Skill	53.06	78.36	14.87	57.82	14.28	**53.93**	**23.00**	11.96
Visual Skill	**53.94**	**80.60**	**14.66**	**59.72**	**13.80**	51.05	21.50	**9.45**
Gemma-4-31B								
No Skill	41.02	62.69	18.66	40.76	16.23	49.66	19.60	**9.47**
Text Skill	57.24	**90.30**	12.01	58.29	14.99	52.66	23.12	9.92
Visual Skill	**58.21**	**90.30**	**11.78**	**59.72**	**14.60**	**52.69**	**24.62**	9.60
Kimi-K2.5								
No Skill	49.73	78.36	16.76	48.82	16.54	50.64	22.00	15.11
Text Skill	56.34	90.30	12.75	50.71	**15.95**	54.88	28.00	**13.35**
Visual Skill	**58.64**	**91.79**	**12.41**	**52.13**	16.06	**56.71**	**32.00**	13.43

decision-state experiments. The complete-agent evaluation covers Qwen3.5-9B, Qwen3.5-27B, Gemma-4-31B-it, and Kimi-K2.5. Detailed benchmark descriptions and evaluation metrics are provided in Appendix A.1.1.

Skill Representations. We compare *Text* and *Visual* representations of the same procedural content. Text supplies skills directly as language tokens, whereas Visual renders the same content as structured skill cards and processes them through the model's native visual pathway. Skill content and selection are matched across representations. Representation-specific details and the additional Rasterized condition are provided in Appendix A.2.4 and Appendix A.3.1.

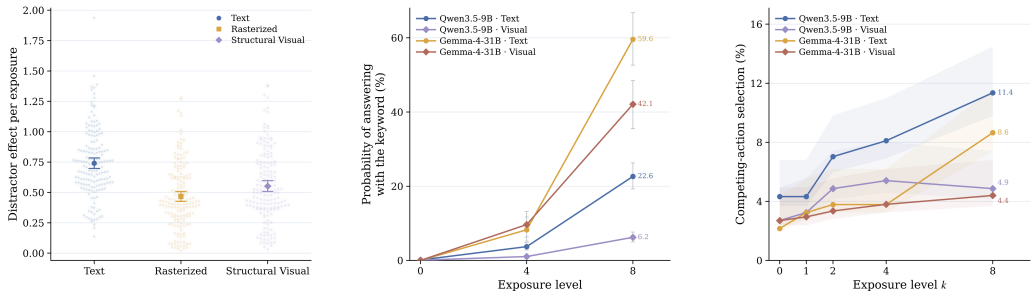
Complete-Agent Evaluation. We compare No Skill, Text Skill, and Visual Skill under the same skill-selection setting across ALFWorld, ScienceWorld, and WebShop. Text and Visual use identical ranked skills and procedural content, differing only in representation. We report task success rate as the primary metric, together with interaction steps where applicable. Further evaluation details, skill-ranking prompts, and task-level results are provided in Appendix A.1.

5 Results and Analysis

5.1 Visual Skills Improve Overall Agent Performance

We first examine whether the representation difference observed in controlled decision states persists in complete agent interaction. To test whether this effect generalizes across model scales, architectures, and deployment settings, we evaluate Qwen3.5-9B, Qwen3.5-27B, Gemma-4-31B-it, and Kimi-K2.5 on ALFWorld (Shridhar et al., 2021), ScienceWorld (Wang et al., 2022), and WebShop (Yao et al., 2022).

Each model is evaluated under three configurations: *No Skill*, *Text Skill*, and *Visual Skill*. The No Skill baseline requires the agent to interact using only its internal parametric knowledge. To isolate the effect of representation from retrieval variation, Text and Visual use the same ranked skill subset. Following SkillNet (Liang et al., 2026), Qwen3.8-Max (Qwen Team, 2026b) ranks the shared skill



(a) Encoding Pathway Intervention.

(b) Repeated Signal Accessibility.

(c) Executable Action Selection.

Figure 3: Representation-Dependent Effects of Procedural Exposure. Figure 3a compares exposure sensitivity under Text, Rasterized, and Structural Visual representations. Figure 3b measures recoverability of the repeated target signal under Text and Visual encoding. Figure 3c reports the greedy selection rate of an executable competing action as procedural exposure increases.

library for each task instance based on its metadata and goal. The selected procedural content is then provided either directly as language tokens or through its rendered visual counterpart.

As shown in Table 1, providing procedural skills consistently improves agent performance over the No Skill baseline. More importantly, Visual achieves a higher macro-average success rate than Text across all four evaluated backbones. Although the magnitude of the gain varies across models and environments, the representation gap remains consistent at the aggregate level, showing that the difference observed in controlled decision states persists throughout complete agent trajectories. At the same time, this end-to-end evaluation alone does not establish reduced procedural exposure bias as the sole cause of the improvement. We therefore next examine how the representation difference arises under controlled exposure. Benchmark descriptions, evaluation metrics, ranking prompts, and task-level results are provided in Appendix A.1.

5.2 Why Visual Skills Reduce Procedural Exposure Bias

We next investigate why procedural exposure exerts a weaker influence under Visual than under Text. We first isolate the contribution of visual encoding from structured layout and test whether repeated procedural signals remain recoverable after encoding. We then examine how the resulting exposure-dependent evidence develops across model depth and finally intervene on internal components to identify how different architectures contribute to the final preference shift.

Visual Encoding Attenuates Procedural Exposure Bias. We first separate visual encoding from spatial layout using three representations of identical procedural content. *Text* provides the skill directly as language tokens. *Rasterized* renders the same wording and ordering as a plain monospaced image, changing the encoding pathway without introducing additional spatial structure. *Structural Visual* further presents the content using the structured skill-card layout adopted in our main experiments. This comparison allows us to distinguish the effect of visual encoding itself from the additional effect of layout.

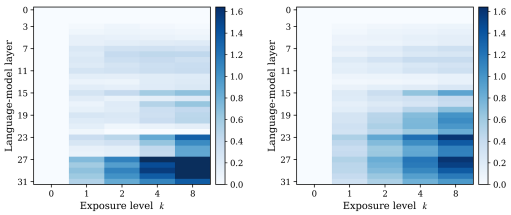
In Qwen3.5-9B, most of the reduction in exposure sensitivity occurs when moving from Text to Rasterized input, whereas Structural Visual produces only a small additional change (Figure 3a). This pattern indicates that the attenuation of PEB is driven primarily by the visual encoding pathway rather than by structured layout alone. Qwen3.5-27B and Gemma-4-31B-it show the same qualitative pattern, with full results reported in Appendix A.3.1.

We next ask whether weaker exposure sensitivity under Visual can be explained simply by reduced access to the repeated procedural signal. To separate signal accessibility from its downstream use in action prediction, we replace the action-prediction objective with a direct recovery probe. The model is prompted to identify the most frequent object at a fixed decoding position for $k \in \{0, 4, 8\}$. As shown in Figure 3b, the target-word probability increases with repetition under both Text and Visual for Qwen3.5-9B and Gemma-4-31B-it. The repeated signal therefore remains recoverable after visual encoding, even though its influence on action prediction is attenuated. This argues against a

simple account in which Visual reduces PEB merely by discarding the repeated information. Additional readout controls and Qwen3.5-27B results are provided in Appendix A.3.2.

Finally, we test whether the representation-dependent preference shift extends beyond candidate scoring to the action actually generated by the agent. We replace the state-inapplicable distractor with an admissible, task-relevant competing action and measure its greedy selection rate under the same exposure manipulation. Increasing exposure raises the probability that the competing action is selected, with a substantially stronger effect under Text than Visual (Figure 3c). The representation-dependent exposure effect therefore extends from latent action preference to executable action selection. The full action-selection protocol is detailed in Appendix A.3.3, while paired rollouts evaluating the downstream interaction cost of these actions are provided in Appendix A.3.4.

Exposure-Dependent Action Evidence Develops Across Model Depth. The preceding results show that Visual changes how repeated procedural signals affect final action prediction. We next trace where this representation-dependent difference emerges inside the model. At each layer, we apply the final normalization and language-model head to the intermediate representations at the candidate-action positions, obtaining a layer-wise readout of exposure-dependent action evidence.

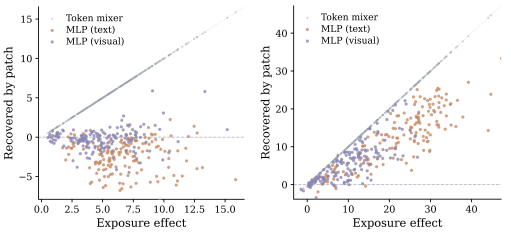


(a) Text. (b) Visual.

Figure 4: Layer-Wise Readout of Exposure-Dependent Action Evidence in Qwen3.5-9B. Values show the mean change in length-normalized distractor action log-probability relative to the $k = 0$ baseline.

As shown in Figure 4, exposure-dependent evidence is weak in the early layers and becomes increasingly decodable through the middle and later layers. The two representations then diverge: Text exhibits a stronger accumulation of exposure-driven evidence in deeper layers, whereas Visual substantially attenuates this late-stage increase. This pattern suggests that the representation gap is not fixed at the input embedding but develops progressively through model computation. Additional layer-wise results across model backbones are provided in Appendix A.3.5.

Component Interventions Reveal Architecture-Specific Roles. The layer-wise readout identifies where exposure-dependent evidence becomes decodable, but it does not establish which internal components contribute causally to the final preference shift. We therefore perform a targeted activation intervention by pairing zero-exposure ($k = 0$) and high-exposure ($k = 8$) runs. During the $k = 0$ pass, we replace the target component output at the distractor positions with the corresponding cached activation from the $k = 8$ run. This directly tests whether injecting the high-exposure component state recovers the exposure-induced increase in distractor support. A positive recovery indicates that the patched component transmits the exposure effect, whereas a negative shift indicates that the component counteracts it. We use the cross-position mixing pathway as a full-recovery reference and focus on the MLP pathway as the primary diagnostic intervention.



(a) Qwen3.5-9B. (b) Gemma-4-31B-it.

Figure 5: Component Intervention Under Controlled Procedural Exposure. The diagonal denotes full recovery of the exposure-induced change, and the horizontal dashed line denotes zero recovery.

The intervention reveals clear architecture-specific differences. In Qwen3.5-9B, patching high-exposure MLP outputs shifts the distractor score against the original exposure effect under both representations (Figure 5a), indicating that the MLP pathway counteracts the exposure-induced preference shift. In Gemma-4-31B-it, the same intervention instead partially recovers the exposure-induced increase (Figure 5b), indicating a transmitting role. Qwen3.5-27B reproduces the Qwen3.5-9B pattern. These results show that similar macroscopic exposure biases can emerge

through different internal computational pathways across model families. Full intervention details are provided in Appendix A.3.6.

6 Related Work

Recent work on skill-augmented language-model agents has studied the broader lifecycle of reusable procedural knowledge, including skill acquisition, organization, retrieval, and deployment (Zhou et al., 2026). Skill acquisition and evolution methods construct, refine, and reuse procedural knowledge from prior experience (Wang et al., 2026a), while recent benchmarks examine whether access to such skills leads to reliable improvements in downstream agent performance (Zhong et al., 2026). As skill libraries continue to grow, retrieval-based approaches increasingly focus on selecting task-relevant procedures from large collections (Zheng et al., 2026; Wang et al., 2026d), and injection-oriented pipelines further determine which retrieved information is ultimately supplied to the model (Li et al., 2026b). Together, these studies have substantially improved how procedural knowledge is acquired and made available to language-model agents, while also highlighting the importance of how supplied skills are ultimately utilized during state-conditioned decision making.

Language models are also known to be sensitive to statistical regularities and repeated patterns in contextual input (Cheng et al., 2025). In in-context learning, variations in demonstration composition and placement can induce majority-label and positional biases, causing predictions to reflect contextual statistics or presentation patterns rather than the intended input–label relation (Gupta et al., 2023; Cobbina & Zhou, 2025). Shortcut-learning studies show that models can rely on repeated lexical patterns or surface cues rather than the intended semantic relation (Doan et al., 2025; Tasawong et al., 2025). Explicit repetition can also systematically alter language-model reasoning behavior (Yu et al., 2025). These effects arise in a broader linguistic setting where token and word frequencies are highly non-uniform, as classically characterized by Zipf's law (Piantadosi, 2014; Dębowski, 2026). Related exposure effects have also been studied in recommender systems, where frequently exposed or popular items can receive preference disproportionate to their relevance (He et al., 2025; Carnovalini et al., 2025). These lines of work collectively demonstrate that frequency, repetition, and contextual statistics can systematically shape model preference, although their effect within supplied procedural knowledge remains less explored.

The representation through which information is supplied can further affect how that information is processed and utilized (van Sprang et al., 2026). Prior work has shown that textual content can be rendered into visual form while preserving its underlying information (Li et al., 2025), while visually rendered inputs are processed through a distinct visual encoding pathway before being integrated into language-model computation (Wang et al., 2026c). Visual representations have also been extensively studied in agents operating over structured interfaces and multimodal observations (Liu et al., 2025; Xu et al., 2024; Hasegawa et al., 2025). In parallel, mechanistic studies have developed circuit-level accounts of Transformer computation and identified recurring mechanisms associated with in-context learning (Elhage et al., 2021; Olsson et al., 2022). More recent work has extended these analyses to layer-wise dynamics and broader patterns of information flow within Transformer models (Atad et al., 2026). Building on these research directions, we study how repeated procedural information within an already supplied skill influences action prediction, and how this effect changes when the same procedural content is represented through textual or visual encoding pathways.

7 Conclusion

This paper investigates how procedural exposure affects skill utilization in language-model agents. We identify and formalize **Procedural Exposure Bias (PEB)**, showing that repeated exposure to an action-associated procedural pattern can systematically shift action preference even when the interaction state and action relevance remain unchanged. We further show that this effect is strongly representation-dependent: visual encoding substantially attenuates PEB while keeping the repeated procedural signal recoverable, and internal analyses reveal architecture-dependent pathways through which exposure-related action evidence develops. Finally, controlled generation and complete-agent evaluations show that these representation differences extend to executable decisions and persist across ALFWorld, ScienceWorld, and WebShop. These findings identify skill representation as an important factor in how language-model agents utilize externally supplied procedural knowledge.

AI USE STATEMENT

Generative AI tools were used to assist with language polishing and improving the clarity of the manuscript. All research ideas, experimental design, implementation, analysis, and conclusions were developed and verified by the authors, who take full responsibility for the content of the paper.

REFERENCES

Ido Andrew Atad, Itamar Zimerman, Shahar Katz, and Lior Wolf. TensorLens: End-to-end transformer analysis via high-order attention tensors. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3452–3468, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.156. URL https://aclanthology.org/2026.acl-long.156/.

Filippo Carnovalini, Antonio Rodà, and Geraint A Wiggins. Popularity bias in recommender systems: The search for fairness in the long tail. *Information*, 16(2):151, 2025.

Liang Cheng, Zhaowei Wang, and Mark Steedman. Llms are frequency pattern learners in natural language inference. *arXiv preprint arXiv:2505.21011*, 2025.

Kwesi Adu Cobbina and Tianyi Zhou. Where to show demos in your prompt: A positional bias of in-context learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 29548–29581, 2025.

Łukasz Dębowski. From zipf's law to neural scaling through heaps' law and hilberg's hypothesis. *Journal of Machine Learning Research*, 27(194):1–35, 2026.

Nhi Hoai Doan, Tatsuya Hiraoka, and Kentaro Inui. Understanding and controlling repetition neurons and induction heads in in-context learning. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 2854–2876, 2025.

Zhichen Dong, Zhanhui Zhou, Zhixuan Liu, Chao Yang, and Chaochao Lu. Emergent response planning in llms. *arXiv preprint arXiv:2502.06258*, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Memp: Exploring agent procedural memory. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17490–17502, 2026.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 Technical Report. *arXiv preprint arXiv:2607.02770*, 2026.

Jiayi Geng, Howard Chen, Ryan Liu, Manoel Horta Ribeiro, Robb Willer, Graham Neubig, and Thomas L Griffiths. Accumulating context changes the beliefs of language models. *arXiv preprint arXiv:2511.01805*, 2025.

Karan Gupta, Sumegh Roychowdhury, Siva Rajesh Kasa, Santhosh Kumar Kasa, Anish Bhanushali, Nikhil Pattisapu, and Prasanna Srinivasa Murthy. How robust are llms to in-context majority label bias? *arXiv preprint arXiv:2312.16549*, 2023.

Jinyi Han, Yuanjian Xu, Ying Liao, Xinyi Wang, Zishang Jiang, Zixiang Di, Fanyang Lu, Zhichao Hu, and Yanghua Xiao. Skill-use: Can llms actually use skills in agentic harnesses? *arXiv preprint arXiv:2608.04828*, 2026.

Kimihiro Hasegawa, Wiradee Imrattanatrai, Masaki Asada, Ken Fukuda, and Teruko Mitamura. Tama: Tool-augmented multimodal agent for procedural activity understanding. *arXiv preprint arXiv:2510.00161*, 2025.

Ming He, Zihao Zhang, Han Zhang, and Chang Liu. Relieving popularity bias in recommender systems via user group-level augmentation. *Applied Soft Computing*, 169:112610, 2025.

Yanna Jiang, Delong Li, Haiyu Deng, Baihe Ma, Xu Wang, Qin Wang, and Guangsheng Yu. Sok: Agentic skills–beyond tool use in llm agents. *arXiv preprint arXiv:2602.20867*, 2026a.

Zhiyuan Jiang, Fangrui Huang, Hanwen Xing, Xander Wu, Yipeng Gao, Rui Cao, Mengdi Wang, Shilong Liu, and Yijiang Li. Demystifying agent skills: Why they work-until they don't. *arXiv preprint arXiv:2608.14036*, 2026b.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Ziwei Chai, Y Charles, HS Che, Cheng Chen, et al. Kimi K2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, et al. Skillsbench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026a.

Yanchao Li, Wanhao Liu, Ben Gao, Jiaqing Xie, Zhehong Ai, Na Zou, Yuqiang Li, and Tianfan Fu. Skillsinjector: Dynamic skill context construction for llm agents. *arXiv preprint arXiv:2605.29794*, 2026b.

Yanhong Li, Zixuan Lan, and Jiawei Zhou. Text or pixels? evaluating efficiency and understanding of llms with visual text inputs. In *EMNLP (Findings)*, pp. 10564–10578, 2025.

Yuan Liang, Ruobin Zhong, Haoming Xu, Chen Jiang, Yi Zhong, Runnan Fang, Jia-Chen Gu, Shumin Deng, Yunzhi Yao, Mengru Wang, et al. Skillnet: Create, evaluate, and connect ai skills. *arXiv preprint arXiv:2603.04448*, 2026.

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Song XiXuan, Yifan Xu, Shudan Zhang, Hanyu Lai, Jiadai Sun, Xinyue Yang, et al. Visualagentbench: Towards large multimodal models as visual foundation agents. In *International Conference on Learning Representations*, volume 2025, pp. 95650–95707, 2025.

Yujian Liu, Jiabao Ji, Li An, Tommi Jaakkola, Yang Zhang, and Shiyu Chang. How well do agentic skills work in the wild: Benchmarking llm skill usage in realistic settings. *arXiv preprint arXiv:2604.04323*, 2026.

Jingcheng Niu, Xingdi Yuan, Tong Wang, Hamidreza Saghir, and Amir H Abdi. Llama see, llama do: A mechanistic perspective on contextual entrainment and distraction in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16218–16239, 2025.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Sangeun Park and Minhae Kwon. Multi^2: Hierarchical multi-agent decision-making with llm-based agents in interactive environments. *arXiv preprint arXiv:2606.03698*, 2026.

Steven T Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130, 2014. URL https://link.springer.com/article/10.3758/s13423-014-0585-6.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. 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, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Panuthep Tasawong, Napat Laosaengpha, Wuttikorn Ponwitayarat, Sitiporn Lim, Potsawee Manakul, Samuel Cahyawijaya, Can Udomcharoenchaikit, Peerat Limkonchotiwat, Ekapol Chuangsuwanich, and Sarana Nutanong. Shortcut learning in safety: The impact of keyword bias in safeguards. In *Proceedings of the The First Workshop on LLM Security (LLMSEC)*, pp. 189–197, 2025.

Angela van Sprang, Laurens Samson, Ana Lucic, Erman Acar, Sennay Ghebreab, and Yuki M. Asano. Same content, different answers: Cross-modal inconsistency in mllms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8781–8790, June 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Chenxi Wang, Zhuoyun Yu, Xin Xie, Wuguannan Yao, Runnan Fang, Shuofei Qiao, Kexin Cao, Guozhou Zheng, Xiang Qi, Peng Zhang, et al. Skillx: Automatically constructing skill knowledge bases for agents. *arXiv preprint arXiv:2604.04804*, 2026a.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. Reinforcement learning for self-improving agent with skill library. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1529–1550, 2026b.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022.

Yifan Wang, Shiyu Li, Peiming Li, Xiaochen Yang, Zheng Wei, and Yang Tang. Render-of-thought: Rendering textual chain-of-thought as images for visual latent reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45236–45253, 2026c.

Zifei Wang, Wei Wen, Qiang Ji, Ruizhi Qiao, and Xing Sun. Skill is not document: A query-conditional benchmark and two-stage retriever for llm agent skill routing. *arXiv preprint arXiv:2606.03565*, 2026d.

Tung-Yu Wu and Melody Lo. U-shaped and inverted-u scaling behind emergent abilities of large language models. In *International Conference on Learning Representations*, volume 2025, pp. 99426–99458, 2025.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Xin Guo, Dingwen Yang, Chenyang Liao, Wei He, et al. 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)*, pp. 27914–27961, 2025.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

Yuqi Xiong, Chunyi Peng, Zhipeng Xu, Zhenghao Liu, Zulong Chen, Yukun Yan, Shuo Wang, Yu Gu, and Ge Yu. Lang2act: Fine-grained visual reasoning through self-emergent linguistic toolchains. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 8375–8399, 2026.

Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, and Caiming Xiong. Aguvis: Unified pure vision agents for autonomous gui interaction. *arXiv preprint arXiv:2412.04454*, 2024.

Minglai Yang, Ethan Huang, Liang Zhang, Mihai Surdeanu, William Yang Wang, and Liangming Pan. How is llm reasoning distracted by irrelevant context? an analysis using a controlled benchmark. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13340–13358, 2025.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *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*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html.

Sangwon Yu, Ik-hwan Kim, Jongyoon Song, Saehyung Lee, Junsung Park, and Sungroh Yoon. Unleashing multi-hop reasoning potential in large language models through repetition of misordered context. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6435–6455, 2025.

Xinyu Zhao, Zhen Tan, Vaishnav Tadiparthi, Nakul Agarwal, Kwonjoon Lee, Ehsan Moradi Pari, Hossein Nourkhiz Mahjoub, and Tianlong Chen. Generative skill composition for llm agents. *arXiv preprint arXiv:2606.32025*, 2026.

YanZhao Zheng, ZhenTao Zhang, Chao Ma, YuanQiang Yu, JiHuai Zhu, Yong Wu, Tianze Xu, Baohua Dong, Hangcheng Zhu, Ruohui Huang, et al. Skillrouter: Skill routing for llm agents at scale. *arXiv preprint arXiv:2603.22455*, 2026.

Shanshan Zhong, Yi Lu, Jingjie Ning, Yibing Wan, Lihan Feng, Yuyi Ao, Leonardo FR Ribeiro, Markus Dreyer, Sean Ammirati, and Chenyan Xiong. Skilllearnbench: Benchmarking continual learning methods for agent skill generation on real-world tasks. *arXiv preprint arXiv:2604.20087*, 2026.

Yingli Zhou, Wang Shu, Yaodong Su, Wenchuan Du, Yixiang Fang, and Xuemin Lin. A comprehensive survey on agent skills: Taxonomy, techniques, and applications. *arXiv preprint arXiv:2605.07358*, 2026.

A Appendix

This appendix provides details of the controlled procedural exposure experiments, the representation-dependent analyses, and the complete-agent evaluation. The following sections describe the experimental constructions, scoring procedures, supplementary results, and visual skill examples.

A.1 Details of the Complete-Agent Evaluation

This section supplements Section 5.1 with benchmark descriptions, evaluation metrics, the ALFWorld skill-ranking prompt, task-level results, and rendered skill examples. The evaluation covers Qwen3.5-9B, Qwen3.5-27B, Gemma-4-31B-it, and Kimi-K2.5 under No Skill, Text Skill, and Visual Skill conditions.

A.1.1 Benchmarks and Evaluation Metrics

The main evaluation reports task success rates across ALFWorld, ScienceWorld, and WebShop. Success rate denotes the proportion of evaluated tasks completed successfully. For each model and skill condition, the macro-average in Table 1 is the arithmetic mean of the three environment-level success rates, giving each environment equal weight. Task-level interaction steps are additionally reported for ALFWorld and ScienceWorld.

ALFWorld. ALFWorld (Shridhar et al., 2021) aligns text-based household environments with embodied tasks. Agents follow natural-language goals by locating objects, interacting with appliances and receptacles, and changing object states. Its six task categories are Pick & Place, Examine in Light, Clean & Place, Heat & Place, Cool & Place, and Pick Two & Place. Task success rate measures completion of the requested household goal. This task-level measure is distinct from partial goal-condition completion. Our task-category success rates and interaction steps are reported in Appendix A.1.3.

ScienceWorld. ScienceWorld (Wang et al., 2022) is an interactive text environment for performing elementary science tasks. The original benchmark contains 30 tasks across 10 topics, including changes in states of matter, electrical conductivity, and biological processes. Agents must carry out procedures in the environment rather than only answer factual questions. The original benchmark provides normalized scores based on required goals and optional subgoals, allowing partial progress to receive credit. Such scores differ from task success rates: the success rates reported in this paper concern completed tasks rather than average partial-progress scores. Task-level results are provided in Appendix A.1.3.

WebShop. WebShop (Yao et al., 2022) is a simulated shopping environment in which agents search for products, inspect product information, select options, and make a purchase according to a natural-language instruction. Its reward evaluates the purchased product against requested attributes, options, price, and product type. The original benchmark distinguishes task score, defined as 100 times mean reward, from success rate, defined as the proportion of episodes receiving reward 1. The WebShop results in Table 1 use success rate, consistent with the main evaluation.

A.1.2 Skill Ranking Prompt

As described in Section 5.1, Qwen3.8-Max ranks the shared skill library for each task instance using its type, name, and goal. Text and Visual use the same ranked skills and procedural content, supplied either as language tokens or as rendered skill cards. Table 2 presents the ALFWorld ranking prompt, whose placeholders contain task information and the available skill names and descriptions.

A.1.3 Task-Level Performance under the Default Skill Setting

The following tables break down the complete-agent results by task category for ALFWorld and ScienceWorld, reporting success rates and interaction steps.

Table 2: Prompt used to rank the shared skill library for each ALFWorld task instance. The placeholders are instantiated with the task type, task name, task goal, and the names and descriptions of all available skills.

Task Specific Skill Ranking Prompt

```
You are an expert in embodied AI agent task decomposition for the
ALFWorld household environment.

Task Instance

Task type: {task_type}
Task name: {task_name}
Goal: {query}

Available Skills ({num_skills} total)

{skills_text}

Instruction

Rank ALL {num_skills} skills by their relevance to completing this
specific task instance: "{query}".

Assign each skill a relevance score from 1 (not relevant at all) to
10 (essential/critical for this task).

Consider which skills would be directly invoked during execution,
which are supportive or auxiliary, and which are irrelevant.

Return a JSON array sorted from MOST relevant (highest score) to
LEAST relevant (lowest score).

Output Format

[
{"skill_name": "xxx", "relevance_score": 10, "reason": "brief
reason"},
{"skill_name": "xxx", "relevance_score": 8, "reason": "brief
reason"},
...
]

Return ONLY the JSON array, with no markdown fences or additional
text.
```

ALFWorld. Tables 3 and 4 report success rates and interaction steps across the six ALFWorld task categories. The task-level breakdown shows that the performance difference between Text and Visual is not uniform across categories. Visual's relative advantage varies with the task category, indicating that the aggregate result reflects different contributions from individual task types. The interaction-step results provide a complementary view of the number of steps used under the two representations.

ScienceWorld. Tables 5 and 6 report the corresponding task-level results for ScienceWorld. The subtasks contributing to the aggregate Text–Visual performance difference vary across model backbones. Thus, the task-level pattern is model-dependent, even when the aggregate comparison favors the same representation. The success-rate and interaction-step breakdowns show how the representation differences vary across individual scientific tasks.

A.2 Details of the Controlled Exposure Experiments

This section supplements Section 2. It describes decision-state selection, distractor construction, exposure-controlled skill variants, candidate-action scoring, the Visual condition, and keyword-replacement controls.

A.2.1 Decision States, Distractors, and Procedural Keywords

We extract 165 intermediate decision steps from successful ALFWorld trajectories. Each trajectory is replayed in the environment simulator to reconstruct the observation history and admissible action

Table 3: Success rate (%) across different ALFWorld task types under the default skill setting. **Overall** denotes the overall success rate. Higher is better (↑). The best result for each task type within each backbone is highlighted in **bold**.

Setting	Overall ↑	Pick ↑	Clean ↑	Heat ↑	Cool ↑	Pick2 ↑	Examine ↑
Qwen3.5-9B							
No Skill	15.67	33.33	6.45	4.35	4.76	11.76	38.89
Text Skill	66.42	66.67	**65.62**	**57.14**	**77.27**	56.25	**73.68**
Visual Skill	**70.15**	**91.67**	64.29	50.00	**77.27**	**88.24**	52.63
Qwen3.5-27B							
No Skill	58.96	87.50	80.65	4.35	14.29	82.35	**83.33**
Text Skill	78.36	**91.67**	80.65	**65.22**	**95.24**	70.59	61.11
Visual Skill	**80.60**	**91.67**	**93.55**	56.52	**95.24**	**94.12**	44.44
Gemma4-31B							
No Skill	62.69	95.83	70.97	0.00	76.19	64.71	66.67
Text Skill	**90.30**	**100.00**	93.55	**78.26**	**95.24**	**94.12**	77.78
Visual Skill	**90.30**	**100.00**	**96.77**	69.57	90.48	88.24	**94.44**
Kimi-K2.5							
No Skill	78.36	95.83	87.10	30.43	76.19	82.35	**100.00**
Text Skill	90.30	95.83	90.32	**65.22**	**95.24**	**100.00**	**100.00**
Visual Skill	**91.79**	**100.00**	**93.55**	**65.22**	**95.24**	**100.00**	**100.00**

Table 4: Average number of interaction steps across different ALFWorld task types under the default skill setting. **Overall** denotes the average number of interaction steps across all tasks. Lower is better (↓). The best result for each task type within each backbone is highlighted in **bold**.

Setting	Overall ↓	Pick ↓	Clean ↓	Heat ↓	Cool ↓	Pick2 ↓	Examine ↓
Qwen3.5-9B							
No Skill	27.59	25.04	29.45	29.78	29.62	27.65	22.56
Text Skill	18.28	17.88	**18.69**	**21.05**	18.18	20.81	**13.00**
Visual Skill	**17.75**	**9.96**	19.25	21.83	**17.91**	**16.94**	20.74
Qwen3.5-27B							
No Skill	19.01	11.21	15.32	29.17	27.48	17.47	**14.39**
Text Skill	14.87	**10.33**	15.87	**16.48**	13.67	17.12	16.39
Visual Skill	**14.66**	11.67	**13.16**	19.87	**11.76**	**13.18**	19.39
Gemma4-31B							
No Skill	18.66	9.38	17.55	30.00	15.81	22.76	17.89
Text Skill	12.01	9.79	**11.84**	17.13	10.19	13.71	**9.28**
Visual Skill	**11.78**	**8.04**	12.81	**13.96**	**9.14**	**12.71**	14.39
Kimi-K2.5							
No Skill	16.76	9.75	16.35	26.26	17.29	20.24	10.78
Text Skill	12.75	**8.50**	**13.13**	22.26	10.67	14.29	**6.61**
Visual Skill	**12.41**	**8.50**	14.71	**17.83**	**10.29**	**12.65**	9.00

set at the selected step. Procedural guidance supplied during the source run is removed from the reconstructed history before constructing the probe.

For each decision step i, let a_i^C denote the action executed in the reference trajectory. We introduce a distractor action a_i^D that is relevant to the overall task and refers to an existing environment entity but is excluded from the current admissible action set. The distractor is therefore task-relevant but state-inapplicable. Its verb is required to differ from that of the reference action.

Each distractor is associated with a target keyword $w(a_i^D)$. Consistent with Section 2.1, the target is an object-related noun associated with the action; for example, `microwave` for `open microwave`. Occurrences are counted using whole-word matching, independently of the model's tokenization. The target keyword must occur in the original supplied skill content and must not ap-

Table 5: Success rate (%) across different ScienceWorld task types under the default skill setting. **Overall** denotes the overall success rate. Higher is better (↑). The best result for each task type within each backbone is highlighted in **bold**.

Setting	Overall ↑	State ↑	Elec. ↑	Class. ↑	Growth ↑	Chem. ↑	Life ↑	Stages ↑	Meas. ↑
Qwen3.5-9B									
No Skill	18.96	0.00	36.67	7.50	5.00	11.54	**70.00**	0.00	5.00
Text Skill	42.18	**8.33**	20.00	**95.00**	**40.00**	19.23	56.67	11.11	**55.00**
Visual Skill	**45.02**	5.56	**46.67**	92.50	25.00	**30.77**	60.00	**22.22**	45.00
Qwen3.5-27B									
No Skill	45.02	19.44	30.00	70.00	35.00	19.23	**73.33**	**22.22**	**75.00**
Text Skill	57.82	**33.33**	50.00	**100.00**	**45.00**	42.31	70.00	**22.22**	60.00
Visual Skill	**59.72**	27.78	**56.67**	**100.00**	**45.00**	**53.85**	**73.33**	**22.22**	60.00
Gemma-4-31B									
No Skill	40.76	**41.67**	46.67	50.00	0.00	7.69	63.33	22.22	**70.00**
Text Skill	58.29	36.11	33.33	**100.00**	**40.00**	**57.69**	**73.33**	**44.44**	55.00
Visual Skill	**59.72**	**41.67**	**56.67**	92.50	**40.00**	50.00	**73.33**	**44.44**	50.00
Kimi-K2.5									
No Skill	48.82	**36.11**	30.00	77.50	25.00	30.77	73.33	11.11	**70.00**
Text Skill	50.71	22.22	**40.00**	**95.00**	35.00	**50.00**	60.00	0.00	55.00
Visual Skill	**52.13**	27.78	30.00	87.50	**45.00**	34.62	**76.67**	**44.44**	55.00

Table 6: Average number of interaction steps across different ScienceWorld task types under the default skill setting. **Overall** denotes the average number of interaction steps across all tasks. Lower is better (↓). The best result for each task type within each backbone is highlighted in **bold**.

Setting	Overall ↓	State ↓	Elec. ↓	Class. ↓	Growth ↓	Chem. ↓	Life ↓	Stages ↓	Meas. ↓
Qwen3.5-9B									
No Skill	17.04	21.62	**22.57**	12.25	22.80	25.36	4.97	21.02	**9.80**
Text Skill	**15.36**	**20.22**	22.67	**7.53**	**19.00**	24.77	4.37	7.67	15.40
Visual Skill	15.95	20.23	24.60	9.38	22.45	**23.02**	**3.67**	**7.56**	14.95
Qwen3.5-27B									
No Skill	15.83	**13.78**	26.27	9.85	20.85	**20.62**	5.97	18.67	18.10
Text Skill	14.28	17.30	**20.57**	**7.00**	19.25	26.05	**3.30**	9.44	12.35
Visual Skill	**13.80**	17.62	20.80	7.35	**18.90**	23.28	**3.30**	**5.67**	**11.25**
Gemma-4-31B									
No Skill	16.23	**15.15**	25.93	10.90	24.70	**19.01**	6.53	18.89	**15.50**
Text Skill	14.99	18.73	23.93	**7.55**	20.45	22.31	**3.33**	6.56	16.05
Visual Skill	**14.60**	17.62	**23.53**	8.07	**19.35**	20.96	**3.33**	**6.47**	16.30
Kimi-K2.5									
No Skill	16.54	**17.47**	27.47	**8.45**	24.30	**20.77**	5.50	18.33	17.15
Text Skill	**15.95**	21.61	**19.60**	9.07	23.20	23.27	**3.37**	11.44	18.15
Visual Skill	16.06	20.56	20.13	9.75	**23.05**	26.27	3.80	**10.11**	**15.30**

pear in the task query or the reference action. Keyword frequency serves as the operational measure of action-associated exposure in these probes.

A.2.2 Exposure-Controlled Skill Construction

We construct skill variants with target exposure levels $k \in \{0, 1, 2, 4, 8\}$ while keeping the interaction state and action applicability fixed. Exposure is counted within the supplied skill, rather than across the full interaction history.

For the zero-exposure baseline ($k = 0$), native occurrences of the target keyword in the manipulated skill sections are replaced with contextually appropriate alternatives. Fixed insertion slots are distributed across the skill document. At exposure level k, exactly k slots contain target-bearing

procedural phrases, and the remaining slots contain filler phrases without the target keyword. Target-bearing and filler phrases are matched in word count. Content outside the manipulation zones remains unchanged, and automated checks verify that the total whole-word occurrence count in the supplied skill equals k.

The total word count is held constant across exposure levels for each decision state. This word-count constraint does not imply equal token counts under every model's tokenizer.

A.2.3 Action Scoring and Distractor Preference

Let $\mathcal{A}_i^{\text{adm}}$ denote the admissible action set at decision step i. The candidate set is

$$\mathcal{A}_i = \mathcal{A}_i^{\text{adm}} \cup \{a_i^D\}. \tag{5}$$

This set is fixed across exposure levels and skill representations for each state. Including the inadmissible distractor permits its model support to be measured without executing it in the environment.

Candidates are scored using teacher forcing with a fixed scoring prefix. We use the length-normalized token log-probability in Equation 2. To make the representation explicit, we write this score as $S_i(a; k, r)$, where r denotes the skill representation and $x_i^{(k,r)}$ denotes the corresponding complete input. Candidate scores are normalized as

$$P_i(a; k, r) = \frac{\exp(S_i(a; k, r))}{\sum_{a' \in \mathcal{A}_i} \exp(S_i(a'; k, r))}. \tag{6}$$

Distractor preference is $P_i(a_i^D; k, r)$. This quantity is a normalized preference over the fixed candidate set, not the probability that unrestricted generation produces the distractor action.

Figure 1a reports mean distractor preference across exposure levels. Figure 1b reports the within-state endpoint change

$$\Delta P_{i,r}^D = P_i(a_i^D; 8, r) - P_i(a_i^D; 0, r). \tag{7}$$

The distribution of these paired changes characterizes variation in the exposure effect across decision states.

A.2.4 Visual Reference Condition

Each exposure-controlled Text skill is paired with a Visual counterpart containing the same procedural wording. The interaction history, exposure level, and candidate actions are matched across the two conditions.

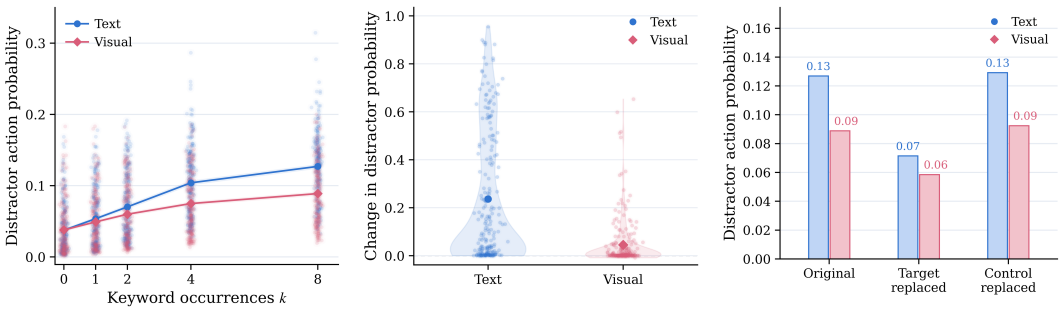
Text supplies the skill as language tokens. Visual renders it as a structured skill card and supplies the resulting image through the model's native visual encoding pathway, without adding auxiliary modules. Unless otherwise specified, Visual refers to this structured skill-card format. Because this comparison changes both the encoding pathway and the presentation of the skill, Appendix A.3.1 additionally evaluates plain rasterized text without the skill-card structure.

A.2.5 Keyword-Replacement Control

At $k = 8$, we compare three conditions to examine whether distractor preference depends on the target keyword rather than lexical replacement in general.

The *original* condition retains the exposure-controlled skill. The *target-replacement* condition replaces all occurrences of the distractor keyword with a neutral alternative while preserving word count and syntactic structure. The *frequency-matched control* condition replaces an unrelated word with the same occurrence frequency while leaving the target keyword intact. Control words are selected to avoid semantic overlap with the distractor action, reference action, task query, and operational verbs.

Figure 1c uses the normalized preference metric defined above. The original and target-replacement conditions include all 165 decision states. A qualifying frequency-matched control word is available for 128 states, and the frequency-matched control statistics use this subset. These sample counts differ; comparisons of aggregate values across the full set and the subset are therefore not fully paired.

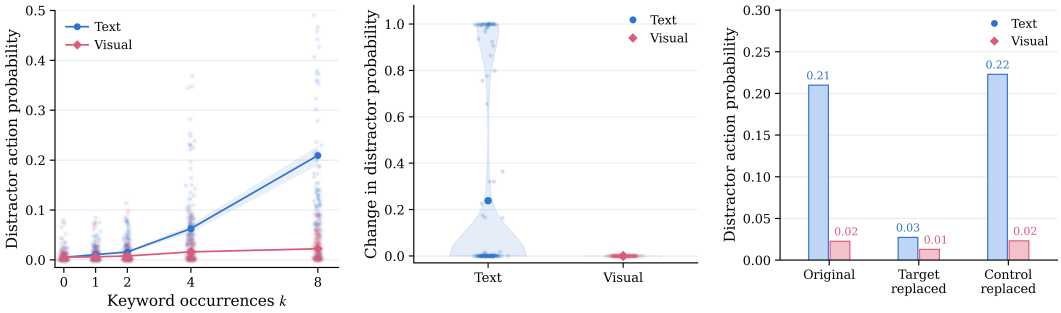


(a) Exposure-Dependent Distractor Preference.

(b) Paired Changes in Distractor Preference.

(c) Keyword-Replacement Controls.

Figure 6: Procedural Exposure Bias in Qwen3.5-27B Under Text and Visual Skill Representations. Figure 6a: Distractor Preference Across Exposure Levels. Figure 6b: Paired Preference Changes From $k = 0$ to $k = 8$. Figure 6c: Keyword-Replacement Controls.



(a) Exposure-Dependent Distractor Preference.

(b) Paired Changes in Distractor Preference.

(c) Keyword-Replacement Controls.

Figure 7: Procedural Exposure Bias in Gemma-4-31B-it Under Text and Visual Skill Representations. Figure 7a: Distractor Preference Across Exposure Levels. Figure 7b: Paired Preference Changes From $k = 0$ to $k = 8$. Figure 7c: Keyword-Replacement Controls.

A.2.6 Additional Models

Figures 6 and 7 report the same controlled probe for Qwen3.5-27B and Gemma-4-31B-it, respectively. Each figure includes exposure-dependent distractor preference, paired endpoint changes, and keyword-replacement controls.

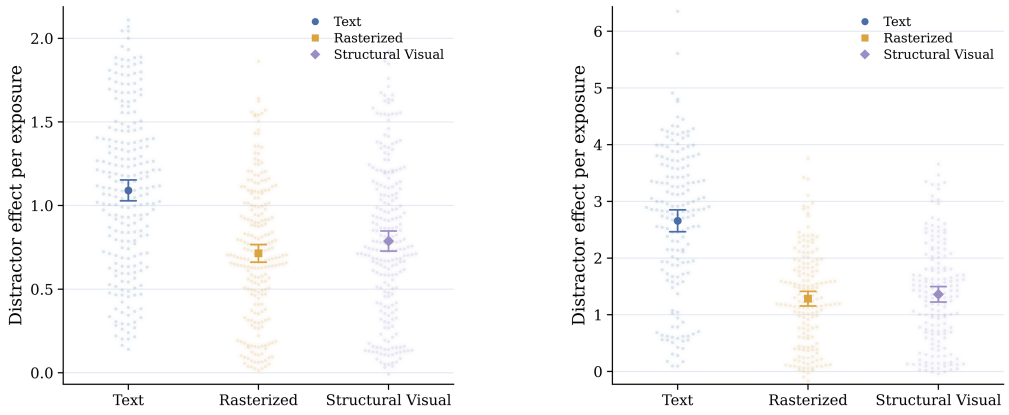
A.3 Details of the Representation-Dependent Exposure Experiments

This section supplements Section 5.2. It describes the rasterized representation comparison, signal readout, executable action selection, paired rollouts, layer-wise readout, and component interventions. Unless otherwise specified, these analyses use the 165 decision states and exposure-controlled variants described in Appendix A.2. The executable-action experiment uses a separate set of states, described below.

A.3.1 Rasterized Representation Intervention

This experiment examines whether the attenuation observed under Visual input also occurs without the additional structure of the skill-card layout.

Skill Representations. We compare three representations of the same procedural content. *Text* supplies the skill as language tokens. *Rasterized* renders the same wording and ordering as a plain monospaced image without additional structural formatting. *Structural Visual* uses the structured



(a) Qwen3.5-27B.

(b) Gemma-4-31B-it.

Figure 8: Additional Results for the Rasterized Representation Intervention. Each panel reports state-level exposure sensitivity under Text, Rasterized, and Structural Visual representations. Central markers and whiskers denote the mean and 95% confidence interval.

skill-card layout adopted in the main experiments. Comparing Text with Rasterized assesses the representation change without additional skill-card structure. Comparing Rasterized with Structural Visual assesses the additional effect of that layout.

Exposure Sensitivity. For this analysis, we use the summed action-token log-probability, denoted by $\widetilde{S}$ to distinguish it from the length-normalized score S used in the candidate-preference probe:

$$\widetilde{S}_i(a; k, r) = \sum_{t=1}^{|a|} \log p_\theta \left(a_t \mid x_i^{(k,r)}, a_{<t} \right). \tag{8}$$

For each state and representation, exposure sensitivity is estimated using the linear fit

$$\widetilde{S}_i(a_i^D; k, r) = \alpha_{i,r} + \beta_{i,r} k + \epsilon_{i,k,r}, \tag{9}$$

where $\beta_{i,r}$ summarizes the change in distractor score with exposure over $k \in \{0, 1, 2, 4, 8\}$.

For Qwen3.5-9B, mean exposure sensitivity decreases from 0.741 under Text to 0.467 under Rasterized input, and increases to 0.552 under Structural Visual input. Gemma-4-31B-it shows the same ordering, with corresponding values of 2.657, 1.283, and 1.361. Substantial attenuation is thus already observed under plain image rendering, without the structured skill-card layout. Figure 8 reports the results for Qwen3.5-27B and Gemma-4-31B-it.

A.3.2 Repeated Signal Readout

This probe examines whether repeated procedural information remains recoverable under visual encoding by replacing action prediction with a direct signal-recovery task.

Setup and Scoring. We use variants in which the target object word occurs $k \in \{0, 4, 8\}$ times and prompt the model to output the most frequent object in the skill. The target word is scored at a fixed answer position. Let $x_{i,\text{read}}^{(k,r)}$ denote the input containing the skill and the readout prompt, and let the target word w tokenize into $(u_1, \ldots, u_m)$. Its sequence log-probability is

$$R_i(w; k, r) = \sum_{j=1}^{m} \log p_\theta \left(u_j \mid x_{i,\text{read}}^{(k,r)}, u_{<j} \right). \tag{10}$$

The corresponding target-token-sequence probability is

$$P_i^{\text{read}}(w; k, r) = \exp(R_i(w; k, r)). \tag{11}$$

This metric measures support for the target word at the designated answer position; it is not an exact-match accuracy measure for the most-frequent-object task.

Table 7: Paired Rollout Results under the Text Skill Representation. The Text condition is evaluated at $k \in \{0, 4, 8\}$. Success rates and mean continuation steps are computed over all 185 states for each branch and exposure level. ΔL denotes the competing-minus-correct continuation-step difference among pairs for which both branches successfully complete the task. Confidence intervals use 4,000 game-clustered bootstrap resamples, and the final column reports the exact McNemar test for paired rollout success. The Pooled row aggregates the both-success pairs across the three exposure levels.

	Success Rate (%)		Mean Steps		Both-Success Analysis		
k	Correct	Competing	Correct	Competing	Pairs	ΔL [95% CI]	McNemar p
0	30.27	28.11	7.60	9.09	25	$+2.52$ [1.48, 3.97]	0.694
4	40.54	30.27	8.09	8.95	33	$+2.97$ [1.69, 4.75]	0.025
8	34.05	26.49	8.06	8.44	28	$+1.36$ [0.48, 2.14]	0.081
Pooled	N/A	N/A	N/A	N/A	86	$+2.31$ [1.61, 3.21]	N/A

Readout Results and Control. Figure 3b reports results for Qwen3.5-9B and Gemma-4-31B-it. Target-word probability increases with exposure under both Text and Visual, indicating that the repeated signal remains recoverable under visual encoding. This does not establish equal accessibility across representations or rule out partial degradation of the signal.

An unrelated frequency-matched control word is scored using the same procedure at $k \in \{4, 8\}$. The control probabilities remain approximately unchanged in the reported comparison, providing a check against a general increase in output confidence.

A.3.3 Executable Action Selection

This experiment tests whether exposure-dependent preference shifts extend to freely generated actions when the competing action is admissible in the current environment state.

Decision States and Competing Actions. We construct a separate set of 185 decision states from 64 successful ALFWorld trajectories. For each state, a_i^C denotes the reference-trajectory action, and a_i^{comp} denotes a task-relevant competing action selected from the current admissible action set. The competing action is executable but differs from the reference action.

Exposure-Controlled Generation. The context contains five ranked skills. We manipulate the keyword associated with the competing action over $k \in \{0, 1, 2, 4, 8\}$ using fixed insertion slots, while preserving content outside those slots and matching word counts across variants. The model freely generates the next action using greedy decoding under Text or Visual input.

For representation r, the competing-action selection rate is

$$R_r(k) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[\hat{a}_{i,r}^{(k)} = a_i^{\text{comp}}\right], \tag{12}$$

where $N = 185$ and $\hat{a}_{i,r}^{(k)}$ is the generated action. Figure 3c reports the selection-rate results for Qwen3.5-9B and Gemma-4-31B-it.

Aggregation and Statistical Analysis. We summarize the aggregate relationship between exposure and selection rate using

$$R_r(k) = \alpha_r + \beta_r k + \epsilon_{r,k}. \tag{13}$$

The reported aggregate fits yield a Text slope of 0.0091 ($R^2 = 0.95$) and a Visual slope of 0.0026 ($R^2 = 0.48$). The paired slope difference is 0.0065, with a bootstrap 95% confidence interval of $[0.0011, 0.0120]$. These fits summarize the aggregate selection-rate trend and do not imply monotonic switching at every decision state.

Table 8: Paired Rollout Results under the Visual Skill Representation. The Visual condition is evaluated at $k \in \{0, 4, 8\}$ using the structured skill-card representation. Success rates and mean continuation steps are computed over all 185 states for each branch and exposure level. ΔL denotes the competing-minus-correct continuation-step difference among pairs for which both branches successfully complete the task. Confidence intervals use 4,000 game-clustered bootstrap resamples, and the final column reports the exact McNemar test for paired rollout success. The Pooled row aggregates the both-success pairs across the three exposure levels.

	Success Rate (%)		Mean Steps		Both-Success Analysis		
k	Correct	Competing	Correct	Competing	Pairs	ΔL [95% CI]	McNemar p
0	25.41	31.35	16.32	17.06	24	+2.17 [0.13, 4.27]	0.185
4	41.08	31.35	15.08	16.97	36	+3.44 [1.75, 5.41]	0.030
8	39.46	31.89	16.31	16.39	38	+1.63 [0.10, 3.27]	0.081
Pooled	N/A	N/A	N/A	N/A	98	+2.43 [1.26, 3.90]	N/A

A.3.4 Paired Rollout Evaluation

This evaluation examines the downstream consequences of executing the competing action rather than the reference action from a matched initial state.

Paired Continuations. For each evaluated state, we create two environment branches. The *reference* branch first executes a_i^C, while the *competing* branch first executes a_i^{comp}. Both branches then continue with unrestricted greedy action generation under the same skill representation and exposure condition within the pair.

Rollouts terminate upon task success, an unparsable action, or exhaustion of the state-dependent budget

$$B_i = \min\big(40,\ B_i^{\text{expert}} + 20\big)\,, \tag{14}$$

where B_i^{expert} is the reference-trajectory step count used to set the budget for state i.

Conditional Step Difference. For pairs in which both branches succeed, we compute

$$\Delta L_{i,r}(k) = L_{i,r,k}^{\text{comp}} - L_{i,r,k}^{C}, \tag{15}$$

where $L_{i,r,k}^{\text{comp}}$ and $L_{i,r,k}^{C}$ are the recorded rollout step counts for the competing and reference branches. A positive value indicates additional steps in the competing branch. This metric is conditional on both branches succeeding and does not capture differences in failure rates.

Tables 7 and 8 report the paired outcomes under Text and Visual, respectively. The paired-rollout comparison assesses the consequences of the initial action choice; it does not by itself estimate the effect of exposure on complete-agent performance.

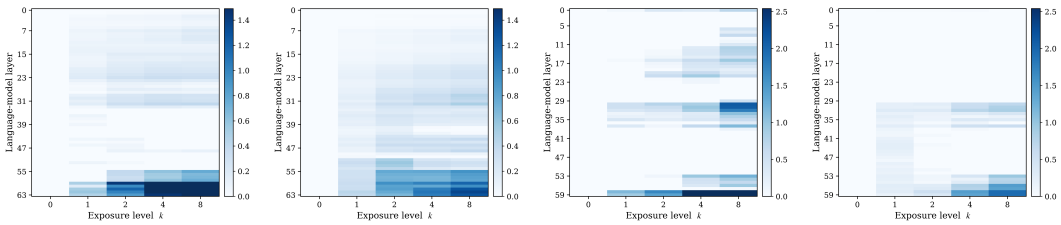
A.3.5 Layer-Wise Readout of Exposure-Dependent Action Evidence

We use intermediate-layer readouts to examine how exposure-related action evidence varies across model depth.

Experimental Setup. We perform teacher-forced forward passes for the reference and distractor actions. Let $q_{i,k,r,t}$ denote the input position immediately preceding action token a_t in the corresponding sequence. At language-model layer l, the hidden state at this prediction position is passed through the model's final normalization and language-model head. The layer-wise action score is

$$S_{i,l}(a; k, r) = \frac{1}{|a|} \sum_{t=1}^{|a|} \log \left[\text{softmax}\left(W_{\text{LM}} \text{Norm}\left(h^{(l)}_{i,k,r,q_{i,k,r,t}} \right) \right) \right]_{a_t}. \tag{16}$$

Here, W_{LM} denotes the language-model output projection, and Norm denotes the final normalization. These scores are intermediate readouts under teacher forcing rather than probabilities from independent generation at each layer.



(a) Qwen3.5-27B, Text. (b) Qwen3.5-27B, Visual. (c) Gemma-4-31B-it, Text. (d) Gemma-4-31B-it, Visual.

Figure 9: Additional Layer-Wise Readouts of Exposure-Dependent Action Evidence. Each heatmap reports the mean change in distractor action evidence relative to the corresponding $k = 0$ condition across model depth.

Exposure-Dependent Readout. The mean change in distractor evidence relative to zero exposure is

$$H_r(l,k) = \frac{1}{n}\sum_{i=1}^{n}\left[S_{i,l}(a_i^D;k,r) - S_{i,l}(a_i^D;0,r)\right], \tag{17}$$

where $n = 165$. The readouts show stronger exposure-dependent distractor evidence under Text in the later layers. Figure 9 reports additional results for Qwen3.5-27B and Gemma-4-31B-it. For Qwen3.5-27B, the measured evidence remains weak through much of the network and increases primarily in the final layers. Weak early-layer readouts do not establish the absence of exposure-related information in those representations.

A.3.6 Matched Component Intervention

We use matched activation interventions to test how selected component outputs affect the exposure-related change in distractor score.

Intervention Protocol. For each state and representation, we cache component activations from a high-exposure donor pass ($k = 8$). During the matched zero-exposure host pass ($k = 0$), the target component outputs at the action-sequence positions are replaced with the cached donor activations.

Using the action-score notation from Appendix A.2.3, the original exposure effect is

$$G_{i,r}^D = S_i(a_i^D;8,r) - S_i(a_i^D;0,r). \tag{18}$$

For a patched component c, the recovered score change is

$$R_{i,r,c}^D = S_i^{\mathrm{patch}(c)}(a_i^D;0,r) - S_i(a_i^D;0,r), \tag{19}$$

where $S_i^{\mathrm{patch}(c)}$ denotes the score from the patched host pass. Aggregate recovery is expressed as

$$\mathrm{Recovery}_{r,c} = 100\,\frac{\mathbb{E}_i[R_{i,r,c}^D]}{\mathbb{E}_i[G_{i,r}^D]}. \tag{20}$$

This ratio is defined when the mean original effect is nonzero and is not constrained to the interval $[0, 100]$. In the state-level scatter plots, the horizontal coordinate is $G_{i,r}^D$ and the vertical coordinate is $R_{i,r,c}^D$. The diagonal indicates full recovery of the original score change, and the horizontal zero line indicates no recovered score change.

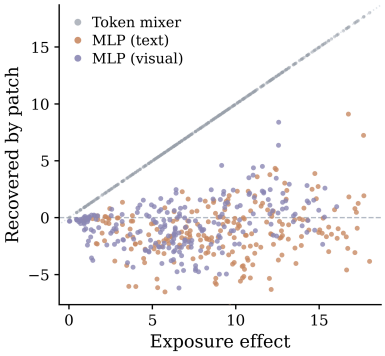


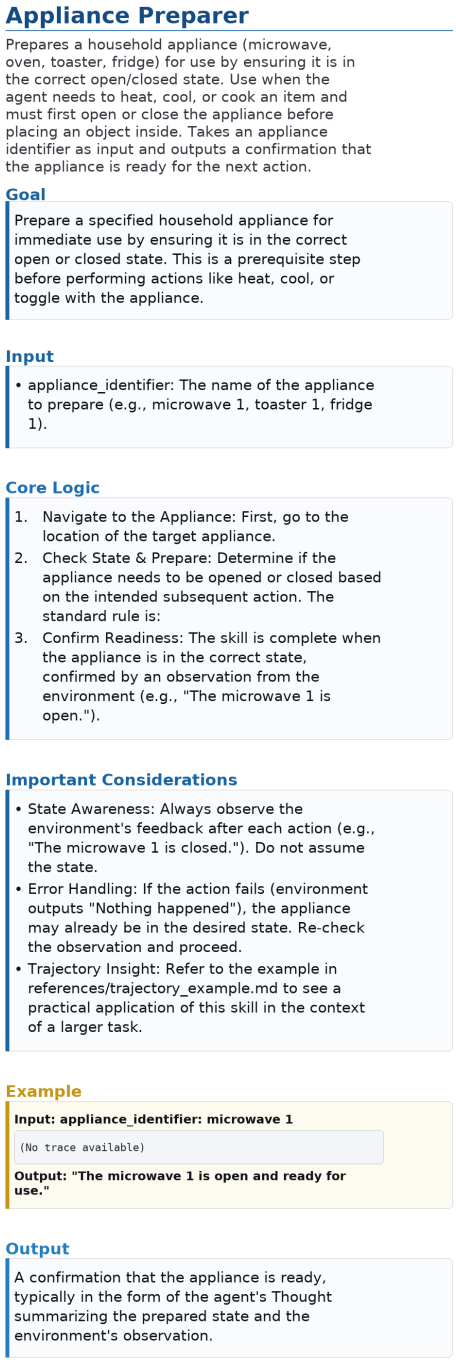
Figure 10: Additional Qwen3.5-27B Results for the Matched Component Intervention. Each point denotes one ALFWorld decision state. The diagonal denotes full recovery, while the horizontal dashed line denotes zero recovered score change.

Component Comparisons. We use the cross-position mixing intervention as a full-recovery reference and the MLP intervention as the primary diagnostic comparison. In Qwen3.5-9B, transplanting high-exposure MLP outputs produces a score change opposite to the original exposure effect

under both representations. In Gemma-4-31B-it, the same intervention produces a change in the same direction as the original effect. These findings describe the effects of the specified activation transplants. Additional Qwen3.5-27B results show the same qualitative MLP intervention pattern as Qwen3.5-9B and are presented in Figure 10.

A.3.7 Examples of Rendered Visual Skills

Figures 11 and 12 illustrate the structured visual format used to present procedural content. The ALFWorld example organizes an appliance-related skill into functional regions, while the WebShop example presents product-search procedures and output constraints.



Appliance Preparer

Prepares a household appliance (microwave, oven, toaster, fridge) for use by ensuring it is in the correct open/closed state. Use when the agent needs to heat, cool, or cook an item and must first open or close the appliance before placing an object inside. Takes an appliance identifier as input and outputs a confirmation that the appliance is ready for the next action.

Goal

Prepare a specified household appliance for immediate use by ensuring it is in the correct open or closed state. This is a prerequisite step before performing actions like heat, cool, or toggle with the appliance.

Input

- appliance_identifier: The name of the appliance to prepare (e.g., microwave 1, toaster 1, fridge 1).

Core Logic

1. Navigate to the Appliance: First, go to the location of the target appliance.
2. Check State & Prepare: Determine if the appliance needs to be opened or closed based on the intended subsequent action. The standard rule is:
3. Confirm Readiness: The skill is complete when the appliance is in the correct state, confirmed by an observation from the environment (e.g., "The microwave 1 is open.").

Important Considerations

- State Awareness: Always observe the environment's feedback after each action (e.g., "The microwave 1 is closed."). Do not assume the state.
- Error Handling: If the action fails (environment outputs "Nothing happened"), the appliance may already be in the desired state. Re-check the observation and proceed.
- Trajectory Insight: Refer to the example in references/trajectory_example.md to see a practical application of this skill in the context of a larger task.

Example

Input: appliance_identifier: microwave 1

```
(No trace available)
```

Output: "The microwave 1 is open and ready for use."

Output

A confirmation that the appliance is ready, typically in the form of the agent's Thought summarizing the prepared state and the environment's observation.

Figure 11: Case Study of a Rendered Visual Skill from ALFWorld. The *Appliance Preparer* skill visually segregates procedural content into explicit functional regions, mapping operational conditions, execution steps, and expected outputs into distinct spatial blocks.

Search Formulator

This skill generates effective search keywords based on parsed product criteria. It is triggered after query parsing, when the agent needs to perform an initial product search on an e-commerce platform. The skill takes structured attributes (e.g., 'women's size 5 patent-beige high heel') and produces a concise, platform-appropriate search string designed to return relevant results, balancing specificity with recall.

Skill: Webshop Search Formulator

Purpose

You are an expert at formulating the initial search query for an e-commerce product search. Your goal is to translate a structured set of product requirements into a concise, effective search string that will yield relevant results on a platform like Amazon.

Core Workflow

1. **Input:** You receive a parsed query containing key product attributes (e.g., category, type, size, color, material, price limit).
2. **Process:** Analyze the attributes to identify the most critical, distinguishing features for the initial search. Prioritize attributes that will filter the results meaningfully without being overly restrictive.
3. **Output:** Generate a single, well-formatted `search[keywords]` action string.

Key Principles for Search Formulation

- **Balance Specificity & Recall:** Start with a moderately specific query. Including 2-3 core attributes (e.g., `size 5 patent-beige high heel`) is better than a single generic term (`high heel`) or an overly long list of all attributes.
- **Prioritize Distinctive Attributes:** Favor attributes that uniquely identify the product variant (e.g., "patent-beige", "size 5") over very common ones (e.g., "women's") in the initial search.
- **Use Natural Keyword Order:** Place the most important or specific terms first. Mimic how a user might type the query.
- **Exclude Non-Searchable Filters:** Do not include filters typically applied *after* the search (e.g., price ranges like `< $90`) in the initial keyword string. These are for later refinement.
- **Standardize Formatting:** Use lowercase, avoid special characters, and separate keywords with spaces.

Example from Trajectory

Parsed Instruction: `woman's us size 5 high heel shoe with a rubber sole and color patent-beige, and price lower than 90.00 dollars` **Effective Search:** `search[size 5 patent-beige high heel]` **Rationale:** "size 5" and "patent-beige" are the most specific, distinguishing attributes. "high heel" defines the product type. "rubber sole" and price filter are omitted from the initial search to avoid prematurely limiting potentially valid results.

Action Format

Your final output must be a single action in the exact format:

Figure 12: Case Study of a Rendered Visual Skill from WebShop. The *Search Formulator* skill presents product-parsing procedures and output-format constraints in a structured visual layout.