SKILLCARD: COMPILING AND EVOLVING PROCEDURAL VISUAL SKILLS AS SINGLE IMAGES

Anonymous authors
Paper under double-blind review

ABSTRACT

Skills enable agents to tackle long-horizon tasks by capturing reusable procedural knowledge from human expertise and past interactions. However, most existing skills are textual and may omit visual cues needed to distinguish states and verify action outcomes. Recent approaches incorporate screenshots and keyframes as visual references alongside textual procedures. Yet preserving this evidence does not by itself make it actionable, as agents must still integrate separately presented instructions and visual evidence to determine when to act, what change to expect, and whether recovery is needed. As new experience reveals gaps in guidance, revisions must also keep the instructions and visual evidence consistent. The challenge is therefore not merely to retain visual information, but to organize it into reusable procedural knowledge that remains coherent as it evolves. To this end, we propose **SkillCard**, which represents each procedural visual skill as a standalone image while keeping its underlying source editable. Specifically, *compiling* organizes instructions and visual evidence into reusable cards, while *evolving* revises their sources based on agent experience and recompiles the cards. Together, these processes support skill reuse at inference and the co-evolution of skills and policies during training. Experiments on *Visual ALFWorld* and *Visual WebShop* show that SkillCard improves success rates by over 10% compared with base models.

1 INTRODUCTION

Multimodal agents are increasingly expected to complete long-horizon visual tasks in interactive environments (Shridhar et al., 2021; Wang et al., 2023b; Tan et al., 2025). These tasks require sustained planning, coordination across multiple stages, and the ability to adapt as the environment changes. Skills help agents meet these demands by capturing reusable procedural knowledge from human expertise and past interactions (Zhang et al., 2025; Wang et al., 2025; Jiang et al., 2026a; Li et al., 2026b). They can be constructed from expert instructions and demonstrations or distilled from an agent's own successful and failed attempts, turning experience into guidance that extends beyond a single episode. Such guidance describes how to pursue a goal, when a procedure applies, and how to assess progress, allowing agents to reuse established knowledge while carrying out new tasks. By reducing the need to rediscover recurring procedures, skills provide continuity across decisions and make accumulated experience available for task execution (Liang et al., 2026; Liu et al., 2026b).

However, most existing skills encode procedural knowledge primarily as textual instructions and workflows (Xia et al., 2026; Li et al., 2026c; Chen et al., 2026; Liu et al., 2026a). While text is compact and easy to revise, visual appearance, spatial relations, and subtle state differences can be difficult to describe concisely and unambiguously. Converting visual experience into words may therefore preserve the intended procedure while omitting cues needed to recognize when it applies. The same limitation affects outcome verification, as a description of the expected result may leave the agent unsure whether an action succeeded or produced a visually similar but incorrect state. For example, an instruction to "verify that the object was picked up" specifies what to check but may leave unclear which visible change would confirm success. As Figure 1 illustrates, a skill may specify the correct procedure yet provide insufficient visual guidance for recognizing when to act and verifying whether the intended outcome has been reached.

To preserve these cues, recent methods (Zhang et al., 2026; Jiang et al., 2026b) augment textual procedures with screenshots and keyframes, sometimes organizing them through explicit state de-

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.

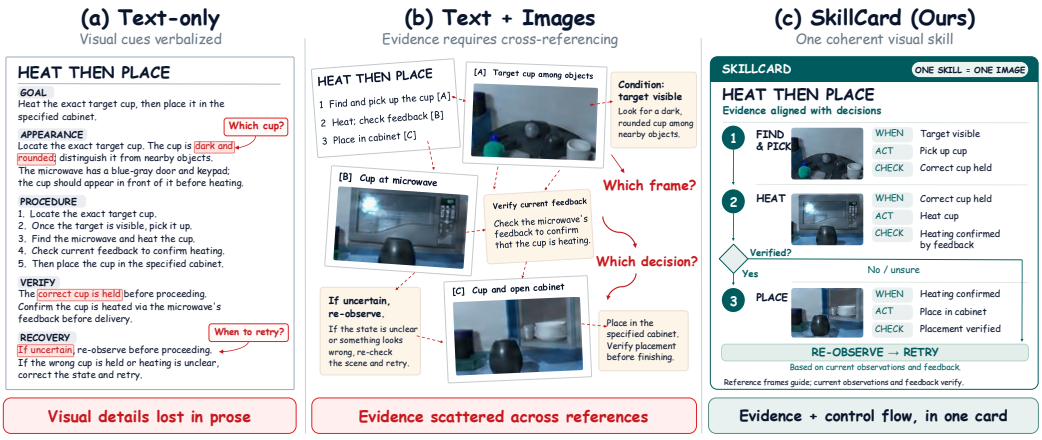


Figure 1: **SkillCard: procedural guidance and visual evidence in one image.** An illustrative comparison of three representations of the same skill. **(a) Text-only** describes visual cues and procedural logic in prose. **(b) Text + Images** retains reference images but requires cross-referencing between instructions and evidence. **(c) SkillCard** aligns evidence with conditions, actions, and checks, making execution order and recovery explicit within one image. Outcome verification relies on current observations and feedback.

scriptions and verification conditions. These methods retain valuable evidence, but two connected challenges remain: **(i) Coherent presentation.** Instructions, state descriptions, and reference images remain distinct components that agents must consult together during execution. Even when their associations are explicit, using the skill requires following these links to determine when to act, what change to expect, and whether recovery is needed. **(ii) Consistent revision.** The correspondence between instructions and evidence must be maintained as skills evolve through further experience, including during policy learning (Xia et al., 2026; Zhang & Qiu, 2026). New failures may reveal missing conditions or inadequate recovery guidance. Revising these instructions can also change which visual evidence is relevant and what it should demonstrate. Updating instructions without corresponding changes to the evidence risks leaving the skill inconsistent. The challenge is therefore to organize instructions and visual evidence into reusable procedural knowledge that remains coherent both during execution and through continued revision.

We view procedural visual skills as editable knowledge that can be compiled into a coherent image for agent use. Building on this perspective, we introduce **SkillCard**, a framework that represents each procedural visual skill as a standalone image through two complementary processes, *compiling* and *evolving*. Compiling arranges instructions and relevant visual evidence within a single card, using spatial grouping to connect action conditions, expected outcomes, and recovery guidance with their supporting cues, as illustrated in Figure 1. Evolving uses agent experience to revise instructions and their associated evidence in the source, then recompiles the affected cards so that their visual presentation reflects the updated procedure. Together, these processes support skill reuse at inference and the co-evolution of skills and policies during GRPO training, where interaction trajectories inform both skill revision and policy optimization. Experiments on *Visual ALFWorld* and *Visual WebShop* show that SkillCard improves success rates by over 10% compared with base models. These results suggest that organizing instructions and visual evidence into a single structured image makes procedural logic explicit and can help agents connect task requirements with relevant observations and actions.

2 Related Work

Experience derived procedural memory. Language agents increasingly externalize interaction experience as reusable knowledge. Reflexion (Shinn et al., 2023) stores verbal feedback, while ExpeL (Zhao et al., 2024), AutoGuide (Fu et al., 2024), and CLIN (Majumder et al., 2023) distill trajectories into transferable insights, context-aware guidelines, or continually refined abstractions. Agent Workflow Memory (Wang et al., 2025) induces reusable routines, and Memp (Fang et al., 2026) explicitly studies the construction, retrieval, correction, and deprecation of procedural memory. ReasoningBank (Ouyang et al., 2025) and EvolveR (Wu et al., 2026) further organize successful

and failed experience into closed-loop self-improvement. Moving from textual memory to explicit skills, Voyager (Wang et al., 2023a) accumulates executable code, while SkillRL (Xia et al., 2026) and SkillGraph (Li et al., 2026c) build evolving hierarchical or graph-structured skill banks coupled to policy learning. These systems establish experience-derived procedures as effective external memory, but their reusable artifacts remain primarily textual or code-based.

Multimodal memory and procedural skill. Multimodal agents retain visual experience to support subsequent decisions. JARVIS-1 (Wang et al., 2023b) reuses multimodal Minecraft experience for planning, while Cradle (Tan et al., 2025) combines screenshot-based control with skill curation and memory. Embodied VideoAgent (Fan et al., 2025) maintains an object-centric scene memory updated after interactions, and HIMM (Li et al., 2026a) combines episodic and semantic memory to support exploration. Most closely related, VisualSkill (Jiang et al., 2026b) constructs hierarchical application-specific skills containing text and figures from documentation and UI exploration, while MMSkills (Zhang et al., 2026) converts public trajectories into state-conditioned packages of procedures, state cards, and keyframes. These works show that visual evidence can support skill reuse, but leave open how an acting policy can revise multimodal procedures from its own continuing successes and failures.

Retrieval and continual multimodal skill evolution. Multimodal agents (Xia et al., 2026; Lu et al., 2026; Shi et al., 2026) draw on both knowledge encoded in pretrained parameters and external knowledge retrieved into context, motivating co-evolution in which interaction experience refines both the policy and the reusable knowledge that guides its decisions. MuRAG (Chen et al., 2022) retrieves jointly from textual and visual memory, whereas MA-LMM (He et al., 2024) compresses streaming observations into a bounded visual memory. Such mechanisms motivate multimodal retrieval under context constraints, but do not connect retrieved evidence to the controlled evolution of action procedures. Conversely, existing procedural-memory methods (Fang et al., 2026; Xia et al., 2026; Li et al., 2026c) update text-centric knowledge without step-aligned visual evidence. AceSkill (Xiong et al., 2026) studies prioritized rollout allocation and clustered knowledge organization for self-evolving multimodal agents. SkillCard complements these directions by making the boundary between editable procedural knowledge and its actor-facing representation explicit, with a typed source compiled into one independently retrievable image and the corresponding cards regenerated through experience-driven source revisions during policy learning. The focus is therefore the compilation and continued evolution of single-image procedural skills under bounded context.

3 METHOD

SkillCard represents reusable procedural knowledge as standalone images that combine instructions with the visual evidence needed to apply and verify them. Its central design separates an editable *procedural source* from the *policy-facing card*. *Compiling* selects and organizes the source's instructions, keyframes, and procedural relations on a shared canvas. *Evolving* uses subsequent interaction experience to revise the source and recompile its cards. This separation allows each skill to be consumed as one image while its procedural logic, evidence, and layout remain explicitly editable. Figure 2 provides an overview of construction, execution, and revision.

3.1 PROBLEM SETTING

We consider an agent pursuing a goal g in a partially observed visual environment. Its decisions draw on knowledge encoded in model parameters and external guidance supplied through context. We focus on *skills*: reusable procedures that specify action conditions, execution steps, expected outcomes, and recovery instructions.

At step t, the agent receives an observation o_t comprising the current image and available environment feedback. Let h_t denote the goal, current observation, action constraints, and a bounded interaction history. Given a skill bank B, a retrieval function R supplies a skill context $\mathcal{S}_t$, which conditions the policy alongside h_t:

$$\mathcal{S}_t = R(h_t, B), \qquad a_t \sim \pi_\theta(\cdot \mid h_t, \mathcal{S}_t). \tag{1}$$

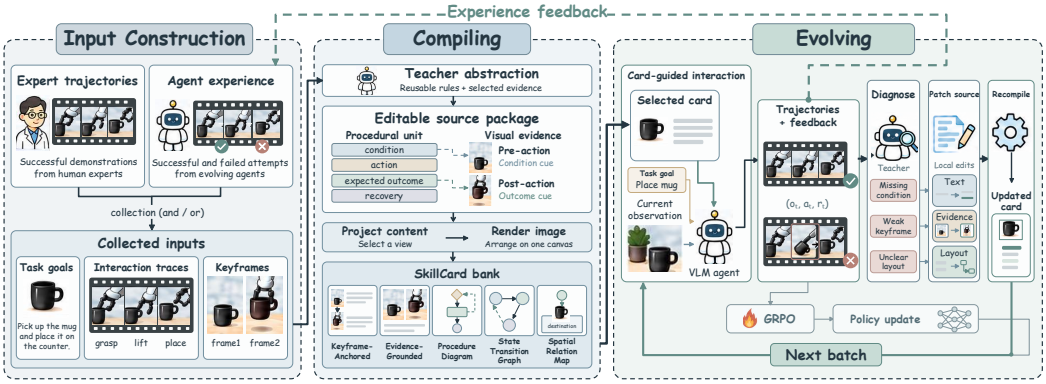


Figure 2: **SkillCard pipeline.** *Input construction* collects task goals, interaction traces, and keyframes from expert demonstrations and agent experience. *Compiling* abstracts these inputs into editable procedural sources and organizes their instructions and evidence into single-image SkillCard variants. *Evolving* uses card-guided trajectories and feedback to locally revise procedural logic, visual evidence, and layout. Recompilation supplies updated cards for subsequent rollouts; during reinforcement learning, the same trajectories also support GRPO policy updates.

Here θ captures parametric knowledge, whereas $\mathcal{S}_t$ supplies explicit procedural guidance that can change without modifying the policy. A trajectory is $\tau = ((o_t, a_t, r_t))_{t=1}^{T}$, where r_t is the environment reward, and the objective is to maximize expected task return $\mathbb{E}[\sum_{t=1}^{T} r_t]$.

The policy parameters θ and bank B can be updated independently or jointly. In SkillCard, retrieved guidance is delivered as card images, while the underlying sources support construction, retrieval, and revision. We study both reuse with frozen policies and banks, and joint policy–skill adaptation during reinforcement learning.

3.2 SkillCard Source Package

A single-image representation brings instructions and visual evidence together, but revising the image directly would make their correspondence difficult to maintain. Changing an action condition, for example, may require revising its wording, replacing a reference frame, and adjusting the connection to a recovery step. We therefore retain an editable *source package* that explicitly records the procedure, its evidence, and their organization. The delivered image is compiled from this package rather than serving as the editable knowledge itself.

For skill i, the source is

$$S_i = (Q_i, E_i, L_i, M_i), \qquad q_{ij} = (c_{ij}, a_{ij}, s_{ij}^{+}, \rho_{ij}, e_{ij}), \tag{2}$$

where Q_i contains procedural units and E_i contains their visual evidence. Each unit q_{ij} associates an applicability condition c_{ij}, action instruction a_{ij}, expected outcome s_{ij}^{+}, and recovery guidance ρ_{ij} with evidence references $e_{ij} \subseteq E_i$. An action specifies how to proceed, but does not by itself establish when the step is appropriate or whether it has succeeded. The condition and expected outcome supply these decision criteria, while recovery guidance describes how to continue when the expected result is not observed. Building on state-conditioned multimodal procedures in MM-Skills (Zhang et al., 2026), we retain these associations as explicit source components that can be revised together.

Instructions use task roles, such as a target object or destination, to describe reusable operations rather than a fixed episode. These roles refer to entities in the current task; historical object identities and locations are not assumed to describe the live scene. The metadata M_i records skill identity, applicability, and provenance to support retrieval and trace the source to its originating experience.

The evidence collection E_i stores selected keyframes with captions and trajectory/frame references. Captions identify the cue to inspect, while procedural links specify why it matters. A pre-action frame can illustrate an applicable condition, whereas a post-action frame can illustrate an expected result. This temporal distinction records whether a frame supports a precondition or an expected

result, rather than treating the two as interchangeable. Historical frames remain references: the current observation and available feedback determine whether a condition or outcome holds in the present episode.

Finally, L_i specifies grouping, ordering, and connections among instructions and evidence. Ordered panels express progression; conditional connections expose branches and recovery paths; spatial grouping associates a rule with its supporting cue. Keeping layout in the source allows revisions to address presentation as well as content.

3.3 Compiling: From Source to a Single Image

Compilation translates the source's selected content and procedural relations into a shared visual canvas. To construct the initial source, a teacher model T_ϕ abstracts training trajectories D_i into reusable instructions and supporting keyframes under source budgets b. For a representation variant r, a projection P_r selects content and a renderer C_r arranges it into one image:

$$S_i = T_\phi(D_i; b), \qquad K_{i,r} = C_r\big(P_r(S_i); \omega_r\big). \tag{3}$$

The projection determines which source information is retained, while the renderer uses layout information L_i and configuration ω_r to determine placement and scale. Source construction and subsequent revision use training-task experience, excluding evaluation tasks.

Compilation organizes information both within and across procedural units. Within a unit, the relevant instruction, caption, and keyframe are grouped so that a perceptual cue appears alongside the rule that explains when to act or what to verify. Across units, ordering and labeled connections express progression, alternative actions, and recovery paths. Thus, the image carries both the content of individual steps and selected relations among them, rather than only placing reference images next to an otherwise separate procedure. For example, a verification rule can appear beside its post-action frame, while a labeled branch links a failed check to the recovery instruction. The arrangement exposes both the evidence's role and the next procedural decision.

We explore five organizations that allocate the canvas differently. *Keyframe-Anchored* pairs instructions with reference frames, emphasizing the connection between a procedural rule and its supporting visual cue. *Evidence-Grounded* allocates more space to reference observations, with nearby text explaining their relevance. These views prioritize perceptual detail for recognizing conditions and interpreting outcomes. *Procedure Diagram* instead foregrounds execution order, conditional decisions, and recovery. *State Transition Graph* places states at nodes and actions along their connections, highlighting expected changes. *Spatial Relation Map* organizes the procedure around relations among task entities.

These variants trade off visual detail and procedural structure within a finite canvas. Because projections may retain different instructions, frames, or relations, a shared source does not imply identical actor-visible content. Source budgets constrain the available material, while resolution affects its legibility. We compare the resulting representations in Section 4.4 and examine resolution separately in Section 4.4.

Once compiled, cards are stored in bank B, indexed by source and variant. Our default *Source Template Retrieval* matches the task context against source fields such as task type and applicability conditions, then selects up to k source–variant pairs:

$$\mathcal{A}_t = R_{\text{template}}(h_t, B; k), \qquad \mathcal{S}_t = [K_{i,r}]_{(i,r)\in\mathcal{A}_t}, \qquad |\mathcal{A}_t| \le k. \tag{4}$$

The budget k counts delivered card images rather than source records. Each selected card enters the policy context alongside the task instruction, live observation, available feedback, and bounded history. Its procedural text is rendered within the image, not additionally supplied as native text. The policy uses the retrieved guidance to interpret the current context and select an action according to Equation 1. These interactions subsequently provide experience for source revision.

3.4 Evolving: Local Patches and Recompilation

Compiled cards provide reusable guidance, while subsequent interaction reveals where that guidance is incomplete or difficult to apply. Trajectories may expose missing action conditions, ambiguous visual cues, or a layout that obscures a recovery path. *Evolving* uses this experience to revise the

source and recompile the affected cards. We implement this process through *Patchify*, which targets the relevant source components while retaining unrelated content, aiming to limit abrupt changes in the guidance supplied to the policy.

The teacher examines card-guided trajectories and environment feedback, identifies guidance requiring revision, and proposes a source patch:

$$\delta_i = (\delta_i^{\text{logic}}, \delta_i^{\text{vis}}, \delta_i^{\text{layout}}), \quad S_i' = \text{Patch}(S_i, \delta_i), \quad K_{i,r}' = C_r\big(P_r(S_i'); \omega_r\big). \tag{5}$$

The logic edit δ_i^{logic} revises conditions, action ordering, expected outcomes, or recovery rules. The visual edit δ_i^{vis} reselects or replaces keyframes and updates their captions and procedural associations. The layout edit δ_i^{layout} adjusts grouping, ordering, and connections to express the revised procedure. All three operate on editable source components rather than directly changing the delivered image's pixels.

These edits are coupled through the associations in the source. Clarifying an expected outcome may require a more informative post-action frame, a revised caption identifying the relevant change, and a clearer connection to the preceding action. Patchify therefore targets affected units and their dependencies rather than treating text, evidence, and layout as unrelated edits. Recompilation then regenerates the corresponding variants from the revised package. Locality constrains the scope of a revision, not its behavioral effect; it does not guarantee improved task performance.

Skill revision can operate with fixed policy parameters, using rollouts on training tasks to refine the bank. During reinforcement learning, the same newly collected trajectories support both source revision and policy optimization. To control when revised guidance enters this process, we hold the bank fixed within each rollout group and its associated policy update. Changes are incorporated at designated evolution rounds and used in subsequent rollouts. These two controls act at different levels: patch scope limits source changes, while the evolution schedule determines when the policy encounters them. At round n,

$$\begin{aligned} D_n &\sim \text{Rollout}(\pi_{\theta_n}, B_n), \\ \theta_{n+1} &= \text{GRPO}(\theta_n; D_n), \\ B_{n+1} &= \text{Recompile}\big(\text{Patchify}(B_n, D_n)\big). \end{aligned} \tag{6}$$

Here bank-level Patchify operates on the sources underlying B_n, and D_n contains the interaction records for that round. GRPO uses environment rewards to update the policy, while Patchify uses trajectories and feedback to revise external procedural guidance. The bank used to collect a rollout remains unchanged during its associated policy update; the revised bank conditions later interaction. The implementation does not perform additional candidate-policy replay or use a behavioral accept/reject gate. Instead, subsequent training rollouts provide feedback on revised cards and experience for further evolution.

3.5 Analysis and Discussion

SkillCard can be viewed as task-oriented information compression: compilation selects reusable instructions and diagnostic evidence from interaction experience, then organizes them within a bounded canvas. The source budget b limits retained material, rendering configuration ω_r controls its visual presentation, and retrieval budget k limits the number of delivered cards. These controls expose a trade-off between context cost and useful information. A compact card can reduce redundant descriptions and observations, but a single image is not inherently inexpensive: token consumption depends on resolution, image preprocessing, and how often cards are supplied. Conversely, excessive compression can remove a critical cue or make an instruction unreadable. Section 4.4 examines this trade-off empirically.

The representation combines complementary ways of expressing procedural knowledge. Text states conditions, actions, and recovery rules; reference images retain appearance and spatial detail; layout exposes order, branching, and instruction–evidence associations. Spatial grouping and labeled connections make selected correspondences visible, placing a rule alongside the evidence for applying or verifying it. The policy must still determine how these references relate to its live observation, but their procedural roles are expressed within the card rather than left entirely to cross-referencing.

Table 1: Experiments on *Visual ALFWorld* with Qwen3-VL-8B under frozen policy and reinforcement learning settings across both in-domain and out-of-domain sets.

Method	Publication	In-Domain							Out-of-Domain						
		Pick	Look	Clean	Heat	Cool	Pick2	**Average**	Pick	Look	Clean	Heat	Cool	Pick2	**Average**
Qwen3-VL-8B: Frozen Policy															
Qwen3-VL-8B (Bai et al., 2025)	-	54.3	7.7	14.8	12.5	8.0	20.8	23.6	33.3	0.0	22.6	0.0	9.5	5.9	13.4
Text Skill	-	60.0	15.4	14.8	25.0	16.0	8.3	26.4	37.5	5.6	22.6	8.7	14.3	5.9	17.2
ExpeL (Zhao et al., 2024)	AAAI 24	48.6	15.4	7.4	18.8	16.0	8.3	21.4	33.3	0.0	25.8	4.3	14.3	5.9	15.7
Memp (Fang et al., 2026)	ACL-F 26	48.6	7.7	22.2	12.5	16.0	16.7	24.3	33.3	11.1	22.6	0.0	9.5	17.6	16.4
XSkill (Jiang et al., 2026a)	ICML 26	57.1	30.8	18.5	31.2	20.0	12.5	30.0	41.7	16.7	25.8	17.4	19.0	23.5	24.6
MMSkills (Zhang et al., 2026)	arXiv 26	51.4	23.1	22.2	12.5	20.0	8.3	25.7	33.3	22.2	25.8	13.0	23.8	11.8	22.4
VisualSkill (Jiang et al., 2026b)	arXiv 26	54.3	23.1	25.9	18.8	16.0	12.5	27.9	37.5	27.8	19.4	13.0	19.0	23.5	23.1
SkillCard	This Paper	60.0	23.1	33.3	37.5	24.0	16.7	**35.0**	50.0	33.3	25.8	13.0	23.8	17.6	**27.6**
Qwen3-VL-8B: Reinforcement Learning															
GRPO (Shao et al., 2024)	-	54.3	53.8	40.7	31.2	32.0	16.7	38.6	50.0	27.8	38.7	13.0	28.6	29.4	32.1
EvolveR (Wu et al., 2026)	ICML 26	48.6	46.2	37.0	25.0	28.0	33.3	37.1	50.0	27.8	41.9	26.1	23.8	17.6	32.8
AgentOCR (Feng et al., 2026)	ACL 26	57.1	46.2	40.7	37.5	36.0	25.0	41.4	58.3	33.3	45.2	30.4	28.6	23.5	38.1
SkillRL (Xia et al., 2026)	ICLR-W 26	60.0	46.2	51.9	43.8	44.0	37.5	48.6	62.5	38.9	54.8	26.1	23.8	35.3	41.8
SkillCard	This Paper	68.6	53.8	63.0	43.8	48.0	45.8	**55.7**	75.0	38.9	61.3	30.4	28.6	29.4	**46.3**

Meanwhile, keeping the source editable allows subsequent experience to revise these roles and associations before recompilation. This separates how procedural knowledge is maintained from how it is presented, supporting continued revision without abandoning the single-image interface.

Table 2: Experiments on *Visual WebShop* and *Visual ALFWorld* with commercial models DeepSeek-V4 and GPT-5.5. Mean reward score and success rate (%) are reported for *Visual WebShop* and task-weighted success rate (%) is reported for *Visual ALFWorld*.

(a) *Visual WebShop*

Method	Publication	Reward Score	Success Rate
DeepSeek-V4: Frozen Policy			
Direct Inference	-	0.285	26.6
Text Skill	-	0.294	28.2
XSkill (Jiang et al., 2026a)	ICML 2026	0.299	28.8
MMSkills (Zhang et al., 2026)	arXiv 2026	0.287	27.6
VisualSkill (Jiang et al., 2026b)	arXiv 2026	0.279	26.8
SkillCard	This Paper	**0.322**	**30.6**
GPT-5.5: Frozen Policy			
Direct Inference	-	0.487	47.0
Text Skill	-	0.491	48.4
XSkill (Jiang et al., 2026a)	ICML 2026	0.546	51.4
MMSkills (Zhang et al., 2026)	arXiv 2026	0.523	50.6
VisualSkill (Jiang et al., 2026b)	arXiv 2026	0.509	49.8
SkillCard	This Paper	**0.565**	**53.6**

(b) *Visual ALFWorld*

Method	Publication	In-Domain	Out-of-Domain
DeepSeek-V4: Frozen Policy			
Direct Inference	-	39.3	44.0
Text Skill	-	42.4	47.0
XSkill (Jiang et al., 2026a)	ICML 2026	46.4	49.3
MMSkills (Zhang et al., 2026)	arXiv 2026	47.2	50.0
VisualSkill (Jiang et al., 2026b)	arXiv 2026	44.4	46.2
SkillCard	This Paper	**51.4**	**54.5**
GPT-5.5: Frozen Policy			
Direct Inference	-	50.8	47.7
Text Skill	-	55.0	51.4
XSkill (Jiang et al., 2026a)	ICML 2026	57.8	55.2
MMSkills (Zhang et al., 2026)	arXiv 2026	59.4	53.7
VisualSkill (Jiang et al., 2026b)	arXiv 2026	56.0	52.2
SkillCard	This Paper	**62.1**	**57.5**

4 Experiments

4.1 Datasets

To evaluate the effectiveness of visual skills in long-horizon tasks, we use *Visual ALFWorld* (Shridhar et al., 2021) and *Visual WebShop* (Yao et al., 2022), modified versions of the original textual benchmarks with image observations for visual and GUI interaction. Both support local interaction for frozen policy evaluation and reinforcement learning. Our experiments cover Qwen3-VL-8B (Bai et al., 2025), DeepSeek-V4 (DeepSeek-AI, 2026), and GPT-5.5 (OpenAI, 2026).

Visual ALFWorld. We use *ALFWorld* (Shridhar et al., 2021) tasks executed in AI2-THOR and adapt the agent interface to provide first-person image observations. The six task families cover object placement, examination under light, cleaning, heating, cooling, and two-object placement. The dataset contains 3,553 training tasks, with 140 in-domain (ID) and 134 out-of-domain (OOD) evaluation tasks. A task succeeds only when its complete goal is satisfied within 50 high-level language actions. At each step, the agent receives the current first-person image observation, task description, admissible actions, and the previous four steps of action history.

Table 3: Ablation on SkillCard representations.

Representation	*Visual ALFWorld* (In-Domain)		*Visual ALFWorld* (Out-of-Domain)		*Visual WebShop*	
	DeepSeek-V4	GPT-5.5	DeepSeek-V4	GPT-5.5	DeepSeek-V4	GPT-5.5
Baseline: Direct Inference	39.3	50.8	44.0	47.7	26.6	47.0
Keyframe Anchored	47.9	58.6	54.5	55.2	29.4	51.2
Evidence Grounded	49.3	53.6	49.3	53.7	28.8	52.0
Procedure Diagram	47.9	58.6	52.2	53.7	29.8	53.4
State Transition Graph	50.7	56.4	51.5	54.5	30.2	52.4
Spatial Relation Map	47.9	53.6	53.7	56.7	28.6	51.8

Table 4: Ablation on skill retrieval.

Retrieval	*Visual ALFWorld* (In-Domain)		*Visual ALFWorld* (Out-of-Domain)		*Visual WebShop*	
	DeepSeek-V4	GPT-5.5	DeepSeek-V4	GPT-5.5	DeepSeek-V4	GPT-5.5
Card Embedding Retrieval	47.9	55.7	50.7	53.0	28.2	51.2
Source Embedding Retrieval	50.0	57.9	52.2	54.5	29.8	52.4
Source Template Retrieval	51.4	62.1	54.5	57.5	30.6	53.6

Visual WebShop. We adapt *WebShop* (Yao et al., 2022) into a screenshot-based GUI environment by rendering its webpages with Chromium while preserving the product, goal, and reward logic. Our setup uses the 1,000-product catalog and 6,910 synthetic shopping goals, comprising 6,410 training tasks and 500 test tasks. Tasks require product search, page navigation, and verification of shopping constraints through symbolic search and click actions. Success requires purchasing a product that satisfies all shopping requirements within 15 actions, and we also report mean task score to capture partial satisfaction. At each step, the agent receives the current webpage screenshot, shopping instruction, available search and click actions, and the previous four steps of action history.

4.2 Experiments with Local Models

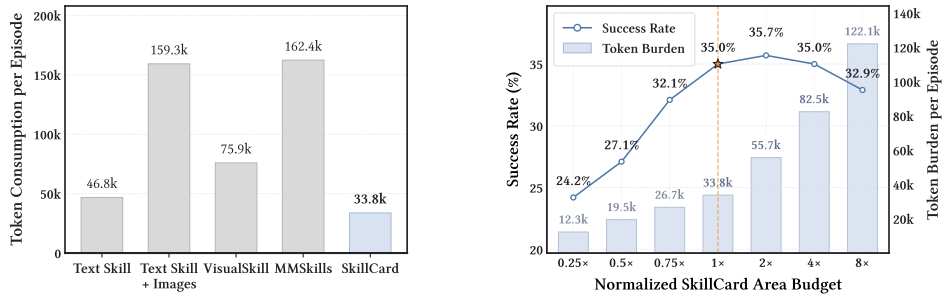
Table 1 presents the results of SkillCard with Qwen3-VL-8B on *Visual ALFWorld*, under both frozen policy evaluation and policy optimization with GRPO. SkillCard improves success rates on long-horizon visual tasks by more than **10%** on both ID and OOD splits, relative to the backbone without skills and standard GRPO in the respective settings. It also surpasses the evaluated textual and multimodal skills, attributed to its organization of descriptive instructions, procedural logic, and key visual cues from training trajectories into a compact image. By explicitly linking instructions to visual evidence, the resulting representation provides a structured reference for action selection and interpretation of visual feedback, helping the agent complete tasks within a limited action budget.

4.3 Experiments with Commercial Models

In Table 2, we report results for commercial models DeepSeek-V4 (DeepSeek-AI, 2026) and GPT-5.5 (OpenAI, 2026), accessed through APIs on *Visual WebShop* and *Visual ALFWorld*. Without updating model parameters, SkillCard improves success rates over direct inference in both environments. These results verify the generalizability of SkillCard across large-scale model backbones and interaction tasks. The gains are attributed to the joint organization of instructions and visual evidence, which makes the correspondence between procedural rules and observable cues more explicit and can help agents apply reusable knowledge to the current task state.

4.4 Ablation Study

Ablation on representations. Table 3 compares five SkillCard representations under frozen API policies and fixed skill banks, varying their emphasis on visual evidence, procedural flow, state changes, and spatial relations, as well as their layouts. All five outperform direct inference across the reported settings, suggesting that multiple forms of information organization support skill reuse. On *Visual WebShop*, explicit procedural flow or state changes show promise across both models, consistent with the sequential nature of shopping interactions. On *Visual ALFWorld*, linking instructions to reference observations and explicitly encoding entity relations appear beneficial, particularly on out-of-domain tasks. These trends motivate combining visual grounding and relational structure to suit task demands.



(a) Token cost across skills.

(b) Ablation on card resolution.

Figure 3: Efficiency analysis on the *Visual ALFWorld* in-domain set with Qwen3-VL-8B.

Ablation on skill retrieval. We try three strategies to retrieve skills relevant to the task input, as shown in Table 4. Source Template Retrieval matches this input against skill fields such as task type and applicability conditions through template rules. Source Embedding Retrieval and Card Embedding Retrieval use Qwen3-VL-Embedding-2B to match the task input against selected textual source fields and rendered card images, respectively, ranking skills by embedding similarity. Source Template Retrieval yields higher task success in our settings, while both embedding approaches offer limited matching effects. For embedding-based retrieval, lengthy skill content may dilute task-relevant cues, while computing embeddings adds overhead. These observations support Source Template Retrieval as our default retrieval strategy.

Skill token cost analysis. We compare per-episode token consumption attributable to skill use on *Visual ALFWorld* ID with Qwen3-VL-8B, as shown in Figure 3a, excluding skill construction and policy training. Text Skill + Images incurs substantial cost because trajectories contain many images, which impose a large token burden without careful selection. MMSkills adds overhead through its roll-in stage where the model first reads the full multimodal package to produce a branch version tailored to the current task. SkillCard selects relevant visual evidence and compiles it with procedural guidance into a compact image, yielding the lowest token consumption among the compared methods.

Ablation on card resolution. Figure 3b scales rendered card area while keeping card content, retrieval, and live-observation resolution fixed. Success rate initially improves with resolution, consistent with small text becoming more legible to the model. At higher resolutions, performance plateaus and can decline as token consumption continues to grow, possibly because the same information is spread across more visual tokens. This trend suggests a trade-off between preserving readable visual detail and limiting context cost. We therefore select a default area budget of approximately 600k pixels ($1\times$), which balances task performance and token consumption.

5 Conclusion

In this work, we propose SkillCard, which treats multimodal skill packages as editable sources that can be compiled into single-image visual skills. Our experiments demonstrate that a single compiled image can provide effective, reusable guidance for multimodal agents. The central insight is that visual composition makes a skill's logic explicit by bringing instructions, visual evidence, and their relationships together in a compact view. This organization provides a human-readable representation and can help agents connect task requirements with relevant observations and actions. These findings support compilation into structured visual representations as a promising approach to communicating reusable knowledge to multimodal agents.

Reproducibility Statement

The main text and appendices describe the method, experimental settings, and implementation details to support reproducibility.

Ethics Statement

All experiments are conducted in simulated environments, without physical deployment or real purchases. Real-world applications would require further evaluation of safety and reliability.

AI Use Statement

AI tools assisted with manuscript organization, language editing, and consistency checks. The authors are responsible for the experimental results, claims, and references.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL Technical Report, 2025. URL https://arxiv.org/abs/2511.21631.

Tianyi Chen, Yinheng Li, Michael Solodko, Sen Wang, Nan Jiang, Tingyuan Cui, Junheng Hao, Jongwoo Ko, Sara Abdali, Leon Xu, Suzhen Zheng, Hao Fan, Pashmina Cameron, Justin Wagle, and Kazuhito Koishida. CUA-Skill: Develop skills for computer using agent. *arXiv preprint arXiv:2601.21123*, 2026. doi: 10.48550/arXiv.2601.21123. URL https://arxiv.org/abs/2601.21123.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. MuRAG: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5558–5570. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.375. URL https://aclanthology.org/2022.emnlp-main.375/.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Yue Fan, Xiaojian Ma, Rongpeng Su, Jun Guo, Rujie Wu, Xi Chen, and Qing Li. Embodied videoagent: Persistent memory from egocentric videos and embodied sensors enables dynamic scene understanding. *arXiv preprint arXiv:2501.00358*, 2025. URL https://arxiv.org/abs/2501.00358.

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. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.866. URL https://aclanthology.org/2026.findings-acl.866/.

Lang Feng, Fuchao Yang, Feng Chen, Xin Cheng, Haiyang Xu, Zhenglin Wan, Ming Yan, and Bo An. AgentOCR: Reimagining agent history via optical self-compression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5067–5086. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.230. URL https://aclanthology.org/2026.acl-long.230/.

Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae, and Honglak Lee. AutoGuide: Automated generation and selection of context-aware guidelines for large language model agents. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d8efbb5dd415974eb095c3f06bff1f48-Abstract-Conference.html.

Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. MA-LMM: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13504–13514, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/He_MA-LMM_Memory-Augmented_Large_Multimodal_Model_for_Long-Term_Video_Understanding_CVPR_2024_paper.html.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R. Fung. XSkill: Continual learning from experience and skills in multimodal agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=AjP1yvCyoG.

Ziyan Jiang, Li An, Yujian Liu, Jiabao Ji, Qiucheng Wu, Jacob Andreas, Yang Zhang, and Shiyu Chang. VISUALSKILL: Multimodal skills for computer-use agents. *arXiv preprint arXiv:2606.18448*, 2026b. URL https://arxiv.org/abs/2606.18448.

Ji Li, Jing Xia, Mingyi Li, and Shiyan Hu. Improving MLLMs in embodied exploration and question answering with human-inspired memory modeling. *arXiv preprint arXiv:2602.15513*, 2026a. URL https://arxiv.org/abs/2602.15513.

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*, 2026b. doi: 10.48550/arXiv.2602.12670. URL https://arxiv.org/abs/2602.12670.

Xiaoyuan Li, Moxin Li, Keqin Bao, Yubo Ma, Wenjie Wang, Dayiheng Liu, and Fuli Feng. SkillGraph: Skill-augmented reinforcement learning for agents via evolving skill graphs. *arXiv preprint arXiv:2605.12039*, 2026c. URL https://arxiv.org/abs/2605.12039.

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. doi: 10.48550/arXiv.2603.04448. URL https://arxiv.org/abs/2603.04448.

Jiateng Liu, Zhenhailong Wang, Rushi Wang, Bingxuan Li, Jeonghwan Kim, Aditi Tiwari, Pengfei Yu, Denghui Zhang, and Heng Ji. OSExpert: Computer-use agents learning professional skills via exploration. *arXiv preprint arXiv:2603.07978*, 2026a. doi: 10.48550/arXiv.2603.07978. URL https://arxiv.org/abs/2603.07978.

Xingyan Liu, Xiyue Luo, Linyu Li, Ganghong Huang, Jianfeng Liu, and Honglin Qiao. SkillForge: Forging domain-specific, self-evolving agent skills in cloud technical support. *arXiv preprint arXiv:2604.08618*, 2026b. doi: 10.48550/arXiv.2604.08618. URL https://arxiv.org/abs/2604.08618.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Skill0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*, 2026. doi: 10.48550/arXiv.2604.02268. URL https://arxiv.org/abs/2604.02268.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. CLIN: A continually learning language agent for rapid task adaptation and generalization. *arXiv preprint arXiv:2310.10134*, 2023. URL https://arxiv.org/abs/2310.10134.

OpenAI. GPT-5.5 System Card, April 2026. URL https://openai.com/index/gpt-5-5-system-card/.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T. Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. ReasoningBank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*, 2025. URL https://arxiv.org/abs/2509.25140.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Yaorui Shi, Yuxin Chen, Zhengxi Lu, Yuchun Miao, Shugui Liu, Qi Gu, Xunliang Cai, Xiang Wang, and An Zhang. Skill1: Unified evolution of skill-augmented agents via reinforcement learning. *arXiv preprint arXiv:2605.06130*, 2026. URL https://arxiv.org/abs/2605.06130.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *The Ninth International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0IOX0YcCdTn.

Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Ziluo Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, Ruyi An, Molei Qin, Chuqiao Zong, Longtao Zheng, Yujie Wu, Xiaoqiang Chai, Yifei Bi, Tianbao Xie, Pengjie Gu, Xiyun Li, Ceyao Zhang, Long Tian, Chaojie Wang, Xinrun Wang, Börje F. Karlsson, Bo An, Shuicheng Yan, and Zongqing Lu. Cradle: Empowering foundation agents towards general computer control. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 58658–58725. PMLR, 2025. URL https://proceedings.mlr.press/v267/tan25h.html.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023a. URL https://arxiv.org/abs/2305.16291.

Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, Xiaojian Ma, and Yitao Liang. JARVIS-1: Open-world multi-task agents with memory-augmented multimodal language models. *arXiv preprint arXiv:2311.05997*, 2023b. URL https://arxiv.org/abs/2311.05997.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911. PMLR, 2025. URL https://proceedings.mlr.press/v267/wang25bx.html.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, and Botian Shi. From interactions to principles: Experience-driven self-distillation for evolving LLM agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=BamDfCP5R3.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. SkillRL: Evolving agents via recursive skill-augmented reinforcement learning. In *ICLR 2026 Workshop on Memory for LLM-Based Agentic Systems (MemAgents)*, 2026. URL https://github.com/aiming-lab/SkillRL.

Feng Xiong, Zengbin Wang, Yong Wang, Xuecai Hu, Jinghan He, Liang Lin, Yuan Liu, and Xiangxiang Chu. Ace-Skill: Bootstrapping multimodal agents with prioritized and clustered evolution, 2026. URL https://arxiv.org/abs/2605.08887.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2207.01206.

Barry Zhang, Keith Lazuka, and Mahesh Murag. Equipping agents for the real world with Agent Skills. Anthropic Engineering, October 2025. URL https://www.anthropic.com/engineering/equipping-agents-for-the-real-world-with-agent-skills.

Kangning Zhang, Shuai Shao, Qingyao Li, Jianghao Lin, Lingyue Fu, Shijian Wang, Wenxiang Jiao, Yuan Lu, Weiwen Liu, Weinan Zhang, and Yong Yu. MMSkills: Towards multimodal skills for general visual agents. *arXiv preprint arXiv:2605.13527*, 2026. URL https://arxiv.org/abs/2605.13527.

Ru Zhang and Weijie Qiu. SPyCE: Skill-policy co-evolution for multimodal agents, 2026. URL https://arxiv.org/abs/2607.13854.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19632–19642, 2024. doi: 10.1609/aaai.v38i17.29936. URL https://ojs.aaai.org/index.php/AAAI/article/view/29936.

A Experimental Details

This appendix describes the benchmark tasks, dataset splits, training settings, and evaluation protocols. Skill banks are constructed separately from each environment's training pool, excluding evaluation tasks from source construction and revision.

A.1 Visual ALFWorld Tasks and Observations

Visual ALFWorld (Shridhar et al., 2021) consists of household tasks in simulated indoor scenes from *ALFRED*, presented through first-person images. Given a natural-language instruction, the agent navigates the scene and interacts with objects to satisfy the goal. We retain solvable tasks from six task families and exclude variants involving sliced objects or movable receptacles. The dataset contains 3,553 training tasks and 274 evaluation tasks, with 140 from the original seen validation split and 134 from the unseen validation split.

The train/seen/unseen counts by task family are 790/35/24 for Pick & Place (Pick), 308/13/18 for Examine in Light (Look), 650/27/31 for Clean & Place (Clean), 459/16/23 for Heat & Place (Heat), 533/25/21 for Cool & Place (Cool), and 813/24/17 for Pick Two & Place (Pick2). At each step, the agent receives the current image, task instruction, available actions, and four steps of action history. An oracle controller localizes objects and executes high-level actions such as navigation, pickup, and placement. Episodes end when the goal is fully satisfied or after 50 actions. Figure 4(a) illustrates the observation and action interface.

A.2 Visual WebShop Tasks and Observations

Visual WebShop (Yao et al., 2022) consists of online shopping tasks in which the agent searches for products, compares candidates, selects options, and completes a purchase that satisfies a natural-language request. We use a catalog of 1,000 products across beauty, electronics, garden/home, fashion, and grocery categories. The dataset contains 6,910 automatically generated goals specifying product requirements, attributes, option combinations, and price limits. The first 500 tasks form the fixed test set, and the remaining 6,410 form the training pool. The training pool incorporates the original validation tasks, with no separate validation set. Expert trajectories cover all 6,410 training tasks, including 6,392 successful and 18 failed trajectories.

At each step, the agent receives a screenshot of the current page, the task instruction, available actions, and four steps of action history. Textual action guidance identifies visible result titles, their row and column positions, and the corresponding clickable targets. Product details such as prices, descriptions, and features are available through the screenshot, without additional textual summaries or hidden page information. The agent uses search and click actions to browse, select options, and purchase a product. API evaluation uses 1280×1280 screenshots and a limit of 15 actions per episode. Figure 4(b) provides an illustrative interaction example; its differences from the evaluation interface are noted in the caption.

A.3 Models and Training Settings

Qwen3-VL-8B is used for frozen policy evaluation and GRPO (Shao et al., 2024) training on *Visual ALFWorld*, while DeepSeek-V4 and GPT-5.5 are evaluated through APIs on both environments. For frozen policy evaluation, skill sources are constructed from rollouts on training tasks, while model parameters and skill banks remain fixed throughout evaluation. GRPO training runs for 50 global steps with 32 prompts per batch and 8 rollouts per prompt. During SkillCard training, the initial bank provides procedural guidance, and new rollouts support both policy updates and skill source revision through Patchify, followed by recompilation. The bank remains fixed within each rollout group and through the associated policy update. The terminal reward is 10 for task success and 0 otherwise, with no additional process rewards. Source Template Retrieval is the default retrieval method. Individual variants and alternative retrievers are evaluated separately in Tables 3 and 4, respectively. Goal wording follows a deterministic template. Comparisons with textual memory, multimodal skills, and RL baselines share the observation interface, action interface, and history format within each setting. The history length is set to 4 in both environments. Tasks must be

completed within 50 environment actions in *Visual ALFWorld* and 15 environment actions in *Visual WebShop*.

A.4 Repeated Evaluation and Result Reporting

Randomness in the evaluation environments and stochastic sampling by the models can produce different interaction trajectories and outcomes under the same experimental configuration. We therefore evaluate each configuration three times with different random seeds and report the highest result in the main text following SkillRL (Xia et al., 2026). To characterize this variability, we use the standard deviation across the three evaluations, as defined in Appendix A.6 and reported for SkillCard in Table 5.

A.5 Evaluation Metrics

For N evaluation tasks and an action limit H, strict success in percent is

$$\mathrm{SR@}H = \frac{100}{N} \sum_{j=1}^{N} \mathbf{1}[\text{task } j \text{ succeeds within } H \text{ actions}], \tag{7}$$

where $H = 50$ for *Visual ALFWorld* and $H = 15$ for *Visual WebShop*. *Visual ALFWorld* success requires complete goal satisfaction, so partial goal completion does not count as success. The aggregate is task-weighted across each evaluation split. *Visual WebShop* success requires executing `buy now` and attaining a final task score of 1.0. Mean task score, reported separately on $[0, 1]$, includes partial credit for matching product type, attributes, options, and price constraints. All gains stated as pp are absolute differences between success rates. The reported RL comparisons assess the complete trained systems, and the current results do not include an otherwise identical fixed-bank SkillCard training control or a local-patch versus full-rewrite ablation.

A.6 Evaluation Variability

For three evaluation results x_1, x_2, x_3 under the same configuration, we quantify variability using the sample standard deviation

$$s = \sqrt{\frac{1}{2} \sum_{r=1}^{3} (x_r - \bar{x})^2}, \qquad \bar{x} = \frac{1}{3} \sum_{r=1}^{3} x_r. \tag{8}$$

For success rates expressed in percent, the standard deviation is in percentage points. Table 5 reports the standard deviations for Qwen3-VL-8B on *Visual ALFWorld* in panel (a), and for the API models on *Visual ALFWorld* and *Visual WebShop* in panels (b) and (c). These settings correspond to the SkillCard results in Tables 1 and 2.

Table 5: Standard deviation of SkillCard success rates across three evaluations with different random seeds, in percentage points. API models use frozen policies.

(a) Qwen3-VL-8B *Visual ALFWorld*

Setting	ID	OOD
Frozen Policy	±1.1	±0.8
GRPO	±1.8	±2.1

(b) API Models *Visual ALFWorld*

Model	ID	OOD
DeepSeek-V4	±0.4	±0.7
GPT-5.5	±0.4	±0.5

(c) API Models *Visual WebShop*

Model	Success Rate
DeepSeek-V4	±0.9
GPT-5.5	±1.0

B Limitations and Future Work

Skill-based agents depend on the coverage and relevance of accumulated experience, and transferring procedural knowledge across tasks and environments remains an open challenge. As skill

libraries grow, retrieving applicable knowledge and composing multiple skills also become more demanding. Future work could explore broader experience collection, hierarchical skill organization, and composition guided by task context to support more flexible reuse.

Continual skill learning further requires balancing adaptation to new experience with the consistency of existing knowledge. For multimodal skills, this includes maintaining alignment between procedural rules and visual evidence under limited context budgets. Promising directions include feedback-guided validation, adaptive allocation of visual detail, and joint optimization of skill selection and policy learning.

C Dataset Interaction Examples

Figure 4 illustrates the interaction interface for both environments. At each step, the agent receives system instructions, the task, a current image, recent action history, and executable actions. It returns a brief rationale and one action. Executing that action updates the observation and feedback used in the next decision.

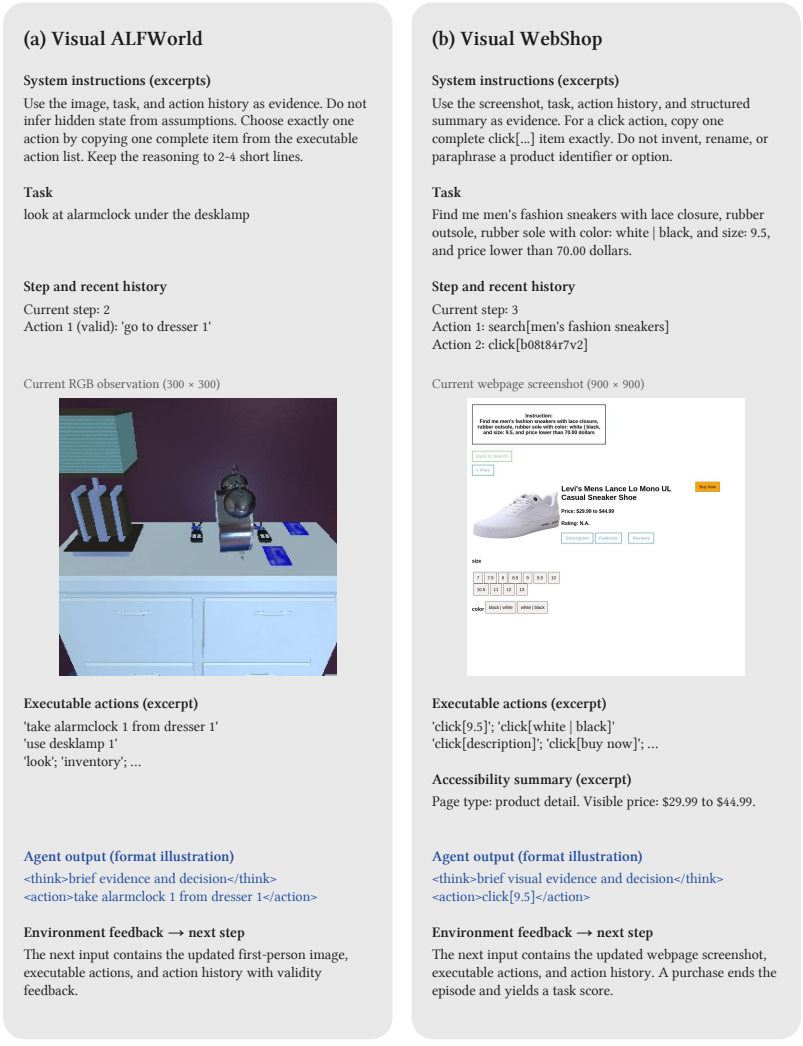
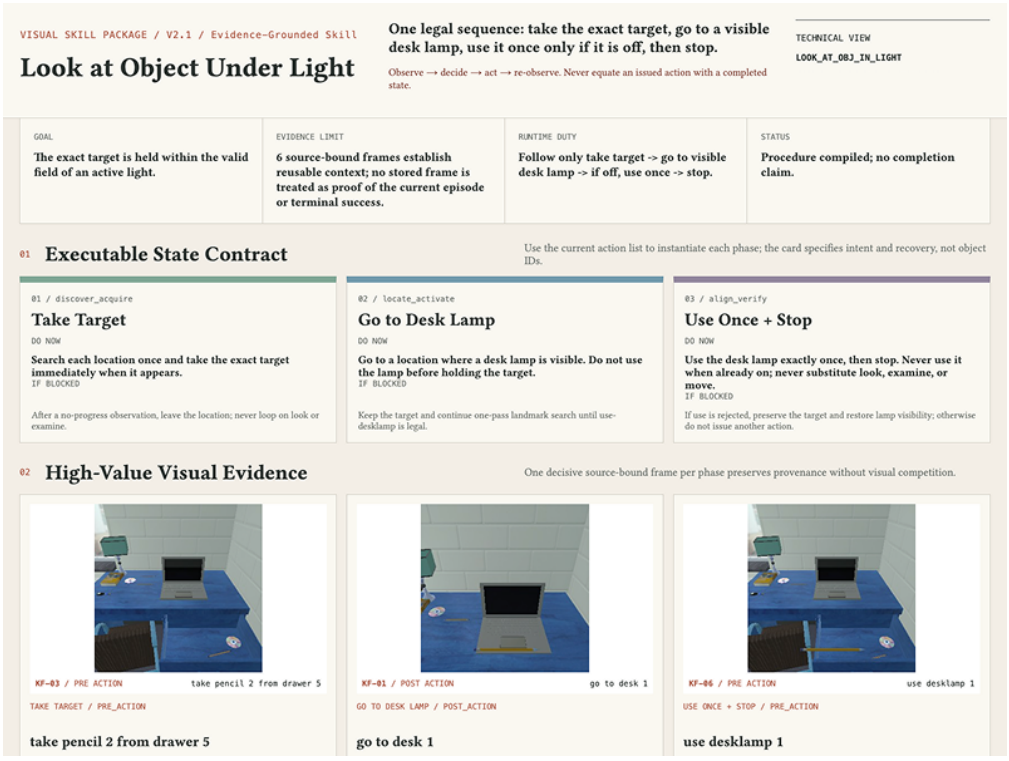
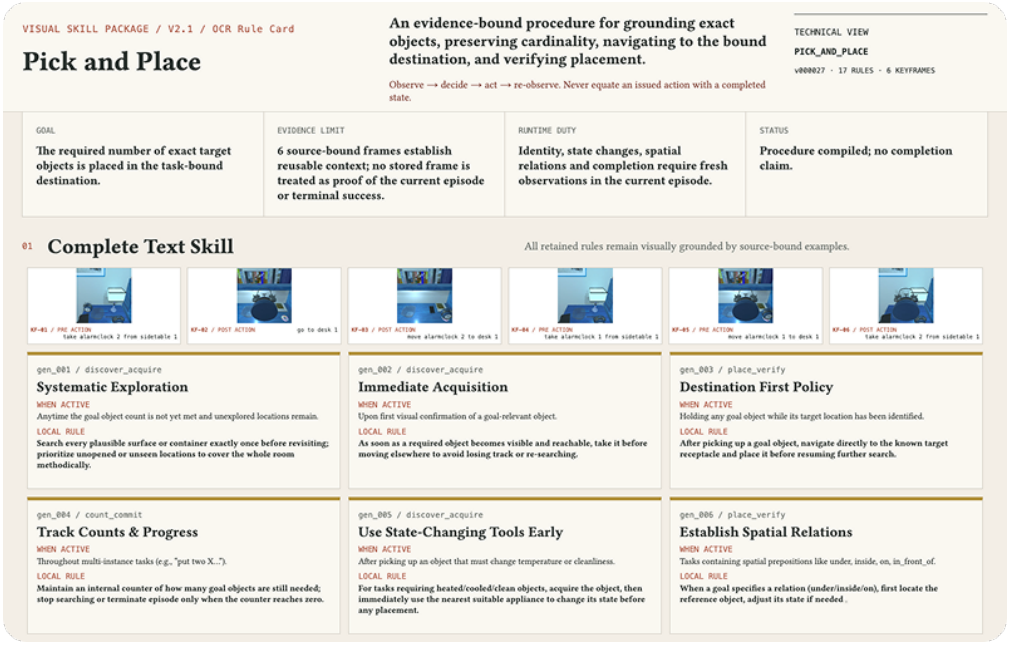


Figure 4: Per-step interfaces for (a) *Visual ALFWorld* and (b) *Visual WebShop*. Instructions and action lists are excerpted; blue responses illustrate output syntax, not recorded generations. Feedback describes the next input. The WebShop example uses a 900×900 screenshot and a summary containing price, whereas evaluation uses 1280×1280 screenshots and minimal grounding (Appendix A.2).

D SKILLCARD EXAMPLES

Figure 5 presents four SkillCard examples. The first card guides object examination under light, from acquiring the target to locating a desk lamp and activating it if needed. Each stage pairs action and recovery instructions with reference frames. The second card organizes object placement around exploration, acquisition, navigation to the destination, and placement verification. Its rules also track the required object count and address state changes and spatial constraints when these are part of the task.





The third card describes online shopping as a sequence of parsing requirements, searching, inspecting a candidate, selecting options, and verifying the purchase. Reference screenshots accompany the stages, while explicit conditions specify when the agent can advance. The fourth card presents a complementary view of this process. It links task constraints to candidate evidence and the configured product, alongside a state diagram that specifies the conditions for search, inspection, configuration, and purchase. Across these examples, stored images provide procedural references, while current observations and feedback determine whether an action has achieved its intended outcome.

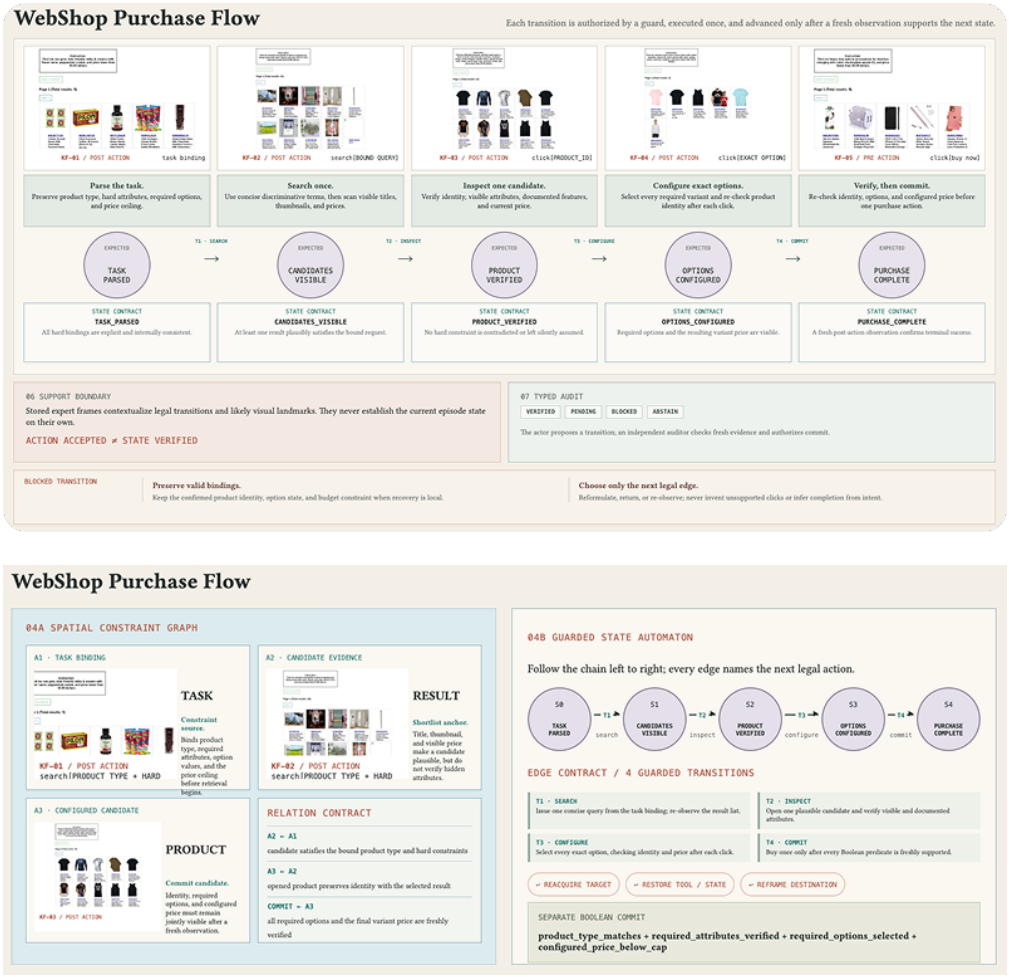


Figure 5: SkillCard examples for visual interaction and online shopping.