A Survey on Agent Skills: Externalized Procedural Knowledge in Language Models

Anonymous ACL submission

Abstract

Large language model (LLM) agents increasingly struggle with reliability, scalability, and execution stability in long-horizon tasks such as software engineering, web automation, and desktop interaction. Recent research addresses these limitations through *agentic skills*: reusable procedural capabilities that externalize execution knowledge into modular, executable, and portable artifacts. This survey presents a unified systems-level review of the emerging agentic skills ecosystem. We formalize skills as externalized procedural abstractions bridging high-level cognitive planning with deterministic execution environments, and organize existing research into a lifecycle-oriented taxonomy spanning discovery, authoring, retrieval, orchestration, execution, adaptation, evaluation, governance, and deprecation. We further analyze security risks including prompt injection, malicious skill packages, and privilege escalation, while outlining open challenges in continual learning, cross-platform portability, and standardized skill ecosystems. This survey positions agentic skills as a foundational abstraction for building scalable, reusable, and reliable autonomous language agents.

1 Introduction

1.1 Motivation: The Reliability and Reusability Bottlenecks in Language Agents

Large language model (LLM) agents representing stateful, autonomous entities have emerged as a popular approach for automating complex, open-ended workflows (Luo et al., 2025). When evaluated on complex, long-horizon software engineering benchmarks, desktop operating system environments, and multi-modal task suites, contemporary agent architectures exhibit a critical performance-limiting reliability cliff (Jimenez et al., 2024). Systemic tracing of these failures indicates that agents are highly vulnerable to early-stage execution and path resolution errors that cascade and propagate downstream (Wang et al., 2026g; Liu et al., 2026f). This error propagation creates an irreversible no-recovery bottleneck, where agents enter endless reasoning loops or exceed platform recursion limits without attempting correct solutions (Pushkin and Abbe, 2026).

This lack of execution stability is compounded by state drift, wherein natural language representations contaminate the context window over time, and models exhibit sycophantic behaviors (Malmqvist, 2024). Traditional mitigation strategies relying on empirical prompt engineering are fragile and inflate inference costs (Chacko et al., 2026). In contrast, externalized executable skills are a useful emerging abstraction separating high-level cognitive planning from deterministic, modular, and portable execution routines (Liu et al., 2024). Recent comprehensive surveys of harness engineering further underscore the necessity of a unified protocol for standardizing these memories and capabilities (Zhou et al., 2026a; Xing et al., 2026).

1.2 What Is an Agentic Skill? Formalizing the Abstraction

To establish a rigorous theoretical foundation, an externalized executable skill is formalized as a six-tuple:

$$s = (\mathcal{A}, \mathcal{I}, \mathcal{C}, \mathcal{T}, \pi, \mathcal{E}) \quad (1)$$

where $\mathcal{A}$ represents the activation or trigger condition, $\mathcal{I}$ represents procedural instructions, $\mathcal{C}$ represents applicability context or constraints, $\mathcal{T}$ represents accessible tools and resources, π represents the execution policy, and $\mathcal{E}$ represents expected outcomes or effects. Unlike atomic tools or stateless APIs, which represent simple functional primitives, a skill is a stateful, procedural routine coordinating multiple tool invocations under task-specific

Table 1: Comparison of computational abstractions.

Abstraction	Execution Semantics	Statefulness	Compositionality	Persistence	Invocation Style	Runtime Coupling	Adaptability	Reference
System Prompts	Declarative	Stateless	Low	Latent	Monolithic	Tight	Low	(Yao et al., 2023; Wei et al., 2022; Chacko et al., 2026)
Episodic Memories	Semantic	Stateful	Low	Transient	Associative	Loose	High	(Packer et al., 2023a; Park et al., 2023)
Workflows / DAGs	Procedural	Stateful	High	Static	Orchestrated	Tight	Low	(Hong et al., 2023; Wu et al., 2023)
Tools / APIs	Functional	Stateless	High	Persistent	Atomic	Loose	Low	(Schick et al., 2023; Patil et al., 2024; Yuan et al., 2024)
Agentic Skills	Hybrid	Stateful	High	Persistent	Contextual	Sandboxed	High	(Wang et al., 2023; Liu et al., 2024)

constraints (Yuan et al., 2024). Skills are distinct from workflows, system prompts, episodic memory, and policies, providing a modular encapsulated capability structure (Thoppilan et al., 2022). This abstraction generalizes system prompts, workflows, tool-use procedures, reasoning traces, executable plans, and procedural memory into a unified capability model (Lu et al., 2026b; Wang et al., 2026a).

1.3 Origins of Skill-Based Agent Design

The development of agentic skills builds upon a technical evolution in model augmentation. The first phase involved static prompting paradigms like ReAct, SayCan, and Inner Monologue, demonstrating that models could perform multi-step reasoning (Sumers et al., 2024). The second phase introduced atomic tool-calling frameworks such as Toolformer, Gorilla, MRKL, APIBank, HuggingGPT, and ViperGPT, prompting language models to emit tokens representing single API calls (Schick et al., 2023; Patil et al., 2024). The third phase transitioned to programmatic loops and persistent state-tracking, as seen in Voyager, MemGPT, Generative Agents, Reflexion, and Self-Refine (Wang et al., 2023). Finally, the modern fourth phase establishes packaged agent skills as a highly capable abstraction for platforms like SWE-Agent, MetaGPT, AutoGPT, WebArena, and Mind2Web, enabling the secure deployment of pre-compiled directories (Song et al., 2403).

1.4 Contributions of This Survey

This survey provides a comprehensive systems-level review of the emerging landscape of agentic skills, organizing a rapidly growing body of literature into a unified lifecycle-oriented taxonomy spanning skill discovery, acquisition, representation, retrieval, execution, orchestration, evolution, governance, and deprecation. We establish a clear conceptual distinction between passive parametric capabilities and externalized procedural knowledge artifacts, formalizing skills as reusable procedural abstractions that separate high-level reasoning from deterministic execution. Furthermore, we synthesize representative architectures, retrieval mechanisms, memory systems, orchestration strategies, and security frameworks across diverse agentic environments. Finally, we identify key open challenges in scalability, portability, safety, evaluation, and continual learning, outlining a research roadmap that positions agentic skills as a promising but still evolving systems abstraction for long-horizon language agents.

2 Foundations of Skill-Based Agents

2.1 Cognitive Architectures and Procedural Memory

In cognitive psychology, memory is broadly divided into declarative memory, which handles semantic facts, and procedural memory, which governs the procedural mechanics of task execution (Sumers et al., 2024). Computational cognitive architectures model human problem-solving as a tight, iterative loop where declarative knowledge is compiled into procedural rules over time (Sumers et al., 2024). Within the context of language agents, agentic skills serve as the non-parametric procedural memory of the autonomous system. While parametric model weights represent a static semantic memory, and vector databases represent a transient episodic memory, skills provide the modular execution substrate required for repeatable tasks (Packer et al., 2023b). When faced with a familiar problem, the agent retrieves the corresponding procedural skill and executes it directly, bypassing slow reasoning steps (Sumers et al., 2024).

2.2 Skills as Externalized Procedural Knowledge

The core architectural innovation lies in the externalization of this procedural memory. Attempting to feed extensive procedural instructions directly into the context window inflates inference latency and degrades accuracy (Chacko et al., 2026). Externalizing procedural knowledge into packaged files resolves these limitations. Furthermore, externalization enables the system-level principle of progressive disclosure, where the agent reads only the lightweight metadata first, loading full implementation code only during active execution (Yuan et al., 2024). Because these skills are stored as

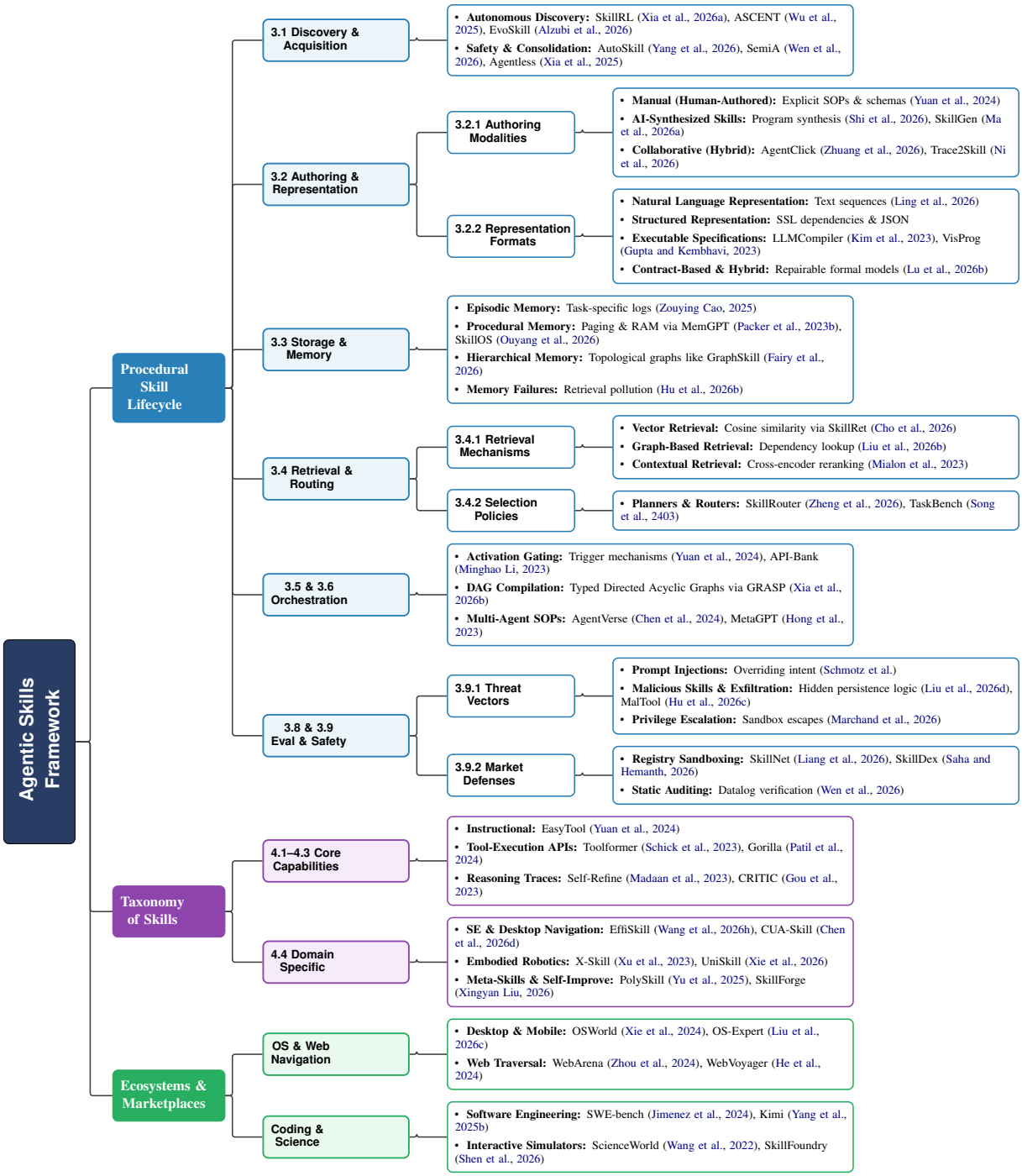

Figure 1: Taxonomy and Landscape of the Agentic Skills Framework, categorizing key system methodologies, lifecycle stages, and domain environments.

plain-text assets, they can be updated without requiring expensive model retraining loops (Zhuang et al., 2026).

We formalize the core primitives of this framework under a unified mathematical abstraction: Definition 1 (Skill): A skill s is a reusable procedural abstraction consisting of activation conditions ($\mathcal{A}$), procedural instructions ($\mathcal{I}$), execution constraints ($\mathcal{C}$), tool interfaces ($\mathcal{T}$), execution policy (π), and intended effects ($\mathcal{E}$) (Yuan et al., 2024):

$$s = (\mathcal{A}, \mathcal{I}, \mathcal{C}, \mathcal{T}, \pi, \mathcal{E}) \quad (2)$$

Definition 2 (Agent State): The agent state at time t is denoted as $x_t \in \mathcal{X}$, capturing the conversation history, internal memory, environment observations, tool outputs, active goals, and retrieved documents (Packer et al., 2023b). Definition 3 (Skill Activation): A skill activates conditionally

based on the agent state and the current task goal g:

$$\mathcal{A}(x_t, g) \to \{0, 1\} \tag{3}$$

where $\mathcal{A}(x_t, g) = 1$ indicates that the skill should trigger, formalizing routing, contextual applicability, and planner-guided invocation (Zheng et al., 2026). Definition 4 (Skill Execution): A skill induces a conditional state transition:

$$x_{t+1} = s(x_t) \quad \text{if} \quad \mathcal{A}(x_t, g) = 1 \tag{4}$$

More explicitly:

$$x_{t+1} = \pi(x_t, \mathcal{I}, \mathcal{T}) \quad \text{if} \quad \mathcal{A}(x_t, g) = 1 \tag{5}$$

This formalizes procedural execution, tool interaction, and reasoning under conditional invocation.

2.3 Structural Components of Skill Execution

From a systems perspective, an executable skill is structured into four distinct functional layers. The declarative layer defines semantic metadata and strict parameter schemas (Yuan et al., 2024). The cognitive layer provides prompt-based instructions outlining the step-by-step plan and explicit error-recovery procedures. The executable layer consists of scripts or compiled binaries that execute procedural operations deterministically (Karpas et al., 2022). The sandbox layer defines the isolated containerized runtime environment required to run the assets securely.

3 Procedural Skill Lifecycle in LLM Agents

The procedural skill lifecycle represents the core operational architecture of modern agent platforms, governing how skills transition from initial creation to runtime deployment, adaptation, and eventual decommissioning. Managing these stages dynamically is critical to maintaining a reliable, safe, and efficient agent capability ecosystem (Yang et al., 2026). This section details each stage of the lifecycle, unifying them under a consistent mathematical framework.

3.1 Skill Discovery and Acquisition

When faced with novel tasks, an agent cannot rely solely on pre-existing skill libraries, necessitating the capability to autonomously discover and acquire new procedural skills (Yang et al., 2026). Skill discovery is the process of identifying novel, repeatable behaviors and abstracting them from a continuous stream of unstructured agent state trajectories (Zhang et al., 2026b; Wu et al., 2025; Alzubi et al., 2026; Yang et al., 2025a). We formalize the discovery operator D as mapping a historical trajectory log $\mathcal{H}$ to a set of candidate skills. Discovery optimizes the partition of a long-horizon trajectory into a sub-trajectory that yields a persistent environmental outcome or minimizes future state transition entropy (Xia et al., 2026a; Shu et al., 2017; Huang et al., 2026). This includes cumulative capability creation through autonomous development (Huang et al., 2026) and deriving skills through pure exploration paired with iterative feedback (Yang et al., 2025a). Multi-agent frameworks also introduce automated discovery mechanisms where systems collaboratively isolate novel behaviors (Alzubi et al., 2026). For physical systems, algorithms like ASCENT leverage foundation models to derive skills directly from complex embodied interactions (Wu et al., 2025). Once a sub-trajectory is isolated, the applicability context under which this behavior remains stable is defined, outlining the initial trigger conditions. This option discovery framework allows language agents to dynamically carve out modular procedural behaviors from complex, interactive task loops (Xia et al., 2026a).

Skill acquisition is the process of formalizing, optimizing, and consolidating a discovered skill candidate into the persistent procedural memory store (Yang et al., 2026). The acquisition pipeline optimizes the instructions and policy to maximize expected success under the applicability context, subject to the safety verification constraint, ensuring that the newly consolidated routine does not introduce malicious vulnerabilities or privilege escalations (Wen et al., 2026). This consolidation approach is critical on repository-level software engineering benchmarks, where direct, localized tool execution achieves robust success rates (Xia et al., 2025).

3.2 Skill Authoring and Representation

Once a skill is discovered or engineered, representing and serializing its artifacts for heterogeneous runtimes is a critical design challenge.

3.2.1 Skill Authoring Modalities

Agentic frameworks employ three primary authoring modalities to define procedural assets. First, manual human-authored skills allow engineers to explicitly author standard operating procedures

(SOPs), declarative metadata schemas, and native scripts (Yuan et al., 2024). This manual approach ensures high safety, deterministic predictability, and exact correctness, but presents a severe human engineering bottleneck and struggles with horizontal scale. Second, AI-synthesized skills empower the agent to autonomously generate its own skills without human intervention (Yang et al., 2026; Jiao et al., 2026; Wang et al., 2025b). This self-generation paradigm leverages techniques like program synthesis networks (PSN) or verifier-guided code generation where the model writes, evaluates, and registers its own code assets and triggering rules (Shi et al., 2026). Advanced frameworks employ contrastive induction over successful and failed trajectories to programmatically synthesize standardized and verified skill directories at runtime (Ma et al., 2026a). Third, collaborative hybrid-authored skills establish a paradigm where human operators collaborate with the model in a human-in-the-loop setup (Ni et al., 2026). Layers like AgentClick allow continuous human verification of terminal actions prior to strict integration (Zhuang et al., 2026). Humans typically define the declarative schemas and safety constraints, while the model generates the underlying procedural implementation and verifies it through sandboxed unit testing (Ni et al., 2026).

3.2.2 Skill Representation Formats

To enable portability, several representation paradigms have been introduced (Ling et al., 2026). Natural language representation remains the primary modality for representing agentic skills, where procedural instructions are represented textually as a sequence of procedural steps (Ling et al., 2026). The activation specification may also be encoded textually, and the model reads these instructions to dynamically align its cognitive planning with task-specific operational constraints (Yuan et al., 2024). Structured representation offers a solution to natural language verbosity, using XML, JSON, or domain-specific languages (DSLs). Under this paradigm, a structured skill is represented as a dependency graph defining execution dependencies, disentangling scheduling signals, execution structure, and logical evidence into distinct schemas. Executable specifications go beyond declarative metadata, requiring formal programmatic specifications to provide a deterministic execution wrapper that schedules parallel tool invocations and handles errors programmatically (Kim et al., 2023; Suris et al., 2023; Gupta and Kembhavi, 2023). Contract-based and unified models improve reliability by establishing formal guarantees for multimodal web agents (Lu et al., 2026b), advocating for an experience compression spectrum that seamlessly unifies memory, rules, and actionable capabilities (Zhang et al., 2026d). Hybrid representations reconcile cognitive flexibility with systems-level safety by combining natural language guidelines, symbolic constraints, and executable native code into a single unified package (Bi et al., 2026; Shi et al., 2026). Finally, skill metadata guarantees interoperability through rich semantic versioning (Liang et al., 2026), while skill embedding supports dynamic retrieval by mapping a skill to a low-dimensional vector representation using an encoder model, providing a continuous representation space for semantic similarity operations (Cho et al., 2026).

3.3 Skill Storage and Memory

An agent's operational expertise is stored within distinct temporal memory tiers to maintain context efficiency and prevent cognitive overload (Sumers et al., 2024; Sun et al., 2026; Tu et al., 2026; Zhang et al., 2026d,c; Lu et al., 2026a; Zhu et al., 2026; Ouyang et al., 2026). Episodic skill memory represents the agent's historical log of task-specific experiences, recording past execution traces, successes, and failed trajectories (Zouying Cao, 2025). Modern platforms extract fine-grained lessons through multi-faceted distillation of these traces to isolate transferable lessons (Ni et al., 2026). Long-term procedural memory serves as the persistent, non-parametric storage substrate for an agent's operational expertise, distinguishing practical knowledge from semantic facts (Sumers et al., 2024). By managing the pre-compiled repository of skills through an active virtual memory paging system, agents can scale their operational expertise significantly (Packer et al., 2023b). Rather than purely retrieving isolated facts, modern RAG systems directly distill enterprise knowledge into these navigable procedural formats (Sun et al., 2026). While vector search is highly effective for coarse-grained matches, capturing the hierarchical dependencies between complex, multi-file skills requires a topological execution graph or hierarchical memory tree to identify candidate routines securely (Fairy et al., 2026). As an agent's memory store grows across sequential tasks, the storage process encounters severe memory failures, including retrieval pol-

lution, context competition, and memory dilution (Hu et al., 2026b; Fang et al., 2026).

3.4 Skill Retrieval and Selection

Retrieving and selecting the optimal skill (or subset of skills) to route a task is driven by a tight, multi-stage runtime selection pipeline (Zheng et al., 2026). Vector retrieval represents the most widely adopted paradigm for dynamically exposing relevant capability packages to agents on-demand. At inference time, the agent platform computes the cosine similarity between the embedded user query and the skill vector store, retrieving the best skill restricted to the active triggering subset to provide fast, low-latency, and context-aware access (Zheng et al., 2026; Cho et al., 2026). Advanced graph-based retrieval incorporates dependency-aware structural mechanisms to filter massive registries efficiently (Liu et al., 2026b). For complex coding scenarios, documentation-guided hierarchical retrieval augments standard lookup paths by strictly following graph reasoning steps to ensure prerequisite and dependent sub-skills are sequenced correctly (Wang et al., 2026b; Fairy et al., 2026). Contextual retrieval utilizes a lightweight sparse or dense vector query to obtain candidates, subsequently deploying a heavy cross-encoder reranker to reason over the full skill instructions and parameters (Cho et al., 2026; Mialon et al., 2023). Once candidate skills are retrieved, the platform dictates selection policies strictly constrained by active triggering rules, utilizing sparse-dense routers and task-specific classification policies to resolve semantic confusability (Zheng et al., 2026; Su et al., 2026; Wang et al., 2026d). Finally, planner-guided routing parses natural language intents into optimized execution plans, scheduling parallel skill invocations and batching requests to reduce latency (Kim et al., 2023; Song et al., 2403; Chen et al., 2026a).

3.5 Skill Activation and Execution

Skill activation and execution manage the active invocation, runtime tracking, and safety boundary enforcement of selected procedural routines (Chen et al., 2026c). Trigger mechanisms define how a skill activates conditionally based on the agent state and the current task goal (Zheng et al., 2026; Minghao Li, 2023). Contextual gating ensures the cognitive planner can gate execution so the agent only acts under verified contexts, utilizing explicit pre-conditions, validation gates, and expected outputs (Yuan et al., 2024). To prevent malicious triggers, permission-aware activation enforces a strict subset of allowed tool invocations, injecting safe activation policies that verify trigger-level safety before execution based on user credentials (Li et al., 2026e). Runtime execution then induces a conditional state transition. To prevent damage to the host system, execution must strictly occur in isolated sandbox containers. Finally, error recovery processes actively monitor execution traces and standard error logs to trigger automated recovery policies and recursive backtracking if a step fails (Li et al.).

3.6 Skill Composition and Orchestration

As tasks scale in complexity, single-skill execution shifts toward advanced composition and orchestration protocols (Xia et al., 2026b; Zabounidis et al., 2026; Xia et al., 2026c; Fan et al., 2026). Sequential composition allows agents to chain multiple distinct skills into a linear execution pipeline, where the output state of a preceding skill directly populates the input parameters of the subsequent routine (Li et al., 2026a; Microsoft, 2024). Hierarchical composition coordinates the workflow by utilizing a master skill to delegate sub-tasks to atomic sub-skills, isolating context windows effectively (Xia et al., 2026b; Du et al., 2024). Dynamic DAG orchestration addresses advanced platform needs by compiling flat retrieved skill candidates into typed Directed Acyclic Graphs, resolving exceptions locally without triggering full-system replanning (Xia et al., 2026b). Recent approaches deploy large language models to guide symbolic planning, subsequently grounding the composed structures via deep reinforcement learning (Zabounidis et al., 2026), while agentic proposing mechanisms enhance reasoning by synthesizing compositional skills dynamically (Jiao et al., 2026; Xia et al., 2026c). Multi-agent coordination shifts orchestration from central scheduling to decentralized coordination among specialized, role-playing agents (Chen et al., 2024; Nie et al., 2026; Li, 2026). These systems define standard operating procedures (SOPs) enabling communicative cooperation while preventing context clashes (Hong et al., 2023; Qian et al., 2024). Finally, recursive invocation enables skills to invoke themselves or their ancestral nodes to resolve nested problems, which is heavily modeled during self-improving reinforcement learning rollouts (Xia et al., 2026a).

3.7 Skill Adaptation and Evolution

Deployed skills require constant refinement, shifting from short-term context adjustments to long-term reinforcement self-improvement (Zhang et al., 2026b; Li et al., 2026d; Wu et al., 2026). Short-term adaptation allows agents to adapt retrieved generic skills dynamically, refining execution parameters based on verifier feedback (Du and Pinckney, 2026). Feedback-driven refinement utilizes reasoning skills to actively analyze previous execution logs, identifying errors and recursively refining generated plans (Madaan et al., 2023; Gou et al., 2023). Long-term evolution ensures that an agent's capability library continually self-improves through operational feedback (Vishe et al., 2026; Ye et al., 2026; Du et al., 2025; Wang et al., 2025a; Ma et al., 2026b; Wang et al., 2026c; Xu et al., 2026b; Mi et al., 2026). To handle complex goals, modern frameworks co-evolve high-level decision policies alongside expanding skill banks (Wu et al., 2026). This intrinsic evolution integrates deeply with hierarchical reinforcement learning protocols, allowing agents to iteratively refine internal logic structures (Li et al., 2026d). To support this, self-improving skills optimize operational primitives and instructions based on historical feedback traces in a lifelong learning cycle, allowing agents to scale expertise dynamically (Yang et al., 2026; Liu et al., 2023; Zhang et al., 2026e; Tziafas and Kasaei, 2024; Jiang et al., 2026a).

3.8 Skill Evaluation

Evaluating performance is essential to establish benchmarks for systems reliability, token economy, and human-agent alignment (Li et al., 2026c; Han et al., 2026; Zhong et al., 2026; Jiang et al., 2026b; Liu et al., 2026f). Correctness evaluates binary reward functions and conditional expected success under specific triggering constraints, providing a reliable correctness metric (Li et al., 2026c). Generalization evaluates the capability to deploy routines in zero-shot situations where no specific match exists, measuring performance differences across domains. Robustness measures resilience to execution exceptions and adversarial inputs (Liu et al., 2026d). Composability verifies the joint success of chaining coordinated graph routines (Xia et al., 2026b). Efficiency focuses on execution costs, incorporating token usage, inference latency, and tool overhead (Gao et al., 2026; Chen et al., 2026a). Interpretability and usability provide structural transparency, allowing users to form bounded expectations and construct local verification checks directly from the procedural specifications (Wen, 2026). In the wild benchmarking in realistic, unconstrained settings is critical to understanding true operational reliability and failure modes outside sterile laboratories (Liu et al., 2026f), while standardized security suites like HarmfulSkillBench measure how compromised files weaponize autonomous systems (Jiang et al., 2026b). High transferability allows decentralized registries to dynamically share and run verified procedures across diverse endpoints (Chen et al., 2026a; Liang et al., 2026).

3.9 Skill Governance and Safety

As agentic modules execute systems-level commands, securing these packages from adversarial exploits is of paramount importance (Li et al., 2026e,f). We map security risks to specific stages of the procedural lifecycle, tracking threat vectors across acquisition, distribution, retrieval, and execution (Li et al., 2026e; Tie et al., 2026). Prompt injections and model poisoning present severe risks where attacking payloads embedded in files or web pages can override instructions, forcing the model to invoke high-impact tools without authorization (Schmotz et al., 2025; Maloyan and Namiot, 2026; Schmotz et al., 2026; Jia et al., 2026). Attackers increasingly deploy model-in-skill poisoning techniques to create stealthy backdoor exploits, fundamentally compromising the execution integrity of shared registries (Tie et al., 2026). Malicious skills and exfiltration frameworks distribute hidden logic on community registries targeting filesystems, registry keys, and network interfaces to establish persistence (Liu et al., 2026d,e; Holzbauer et al., 2026). Large-scale empirical studies reveal extensive security vulnerabilities concerning severe credential leakage and unauthorized file exfiltration to attacker-controlled endpoints (Liu et al., 2026e; Chen et al., 2026e; Hu et al., 2026c; Wang et al., 2026f). Privilege escalation occurs when an executable escapes its designated boundary, demanding that safe activation policies be verified before execution (Li et al., 2026e). Highly capable planners can systematically discover kernel flaws to execute host escapes from misconfigured sandboxes (Marchand et al., 2026; Xiao et al., 2026; Duan et al., 2026). Marketplace security relies on registries establishing automated dynamic sandboxing and cryptographic signatures (Li et al., 2026e; **?**; Hu et al., 2026a; Saha and Hemanth, 2026; Bhard-

waj; Guo et al., 2026; Hou and Yang, 2026; Feng et al., 2026; Zhang et al., 2026a; Lv et al., 2026; Qu et al., 2026). Finally, auditing and synthesis frameworks deploy static analyzers using verification functions to mathematically prove safety compliance before execution (Wen et al., 2026).

3.10 Skill Deprecation and Forgetting

As libraries expand, they accumulate redundant files necessitating automated pruning, memory compression, and retirement policies to maintain operational registry solvency (Pu et al., 2026; Hu et al., 2026b).

4 Open Challenges and Future Directions

As the landscape matures, several structural limitations define the technical frontier. Skills must execute reliably across heterogeneous model APIs. Furthermore, deploying these complex skill frameworks on small language models presents a critical frontier for industrial and edge environments, where computational constraints mandate highly efficient execution and unified serialization standards (Xu et al., 2026a; Ling et al., 2026; Chen et al., 2026b; Wang et al., 2026e). Performing precise selection at scale requires sparse-dense hierarchical routers to avoid semantic confusability (Zheng et al., 2026). Similarly, orchestrating complex dependencies requires compiling files into typed DAGs to support dynamic error repair (Xia et al., 2026b). Enforcing robust capability-based permission models is essential to prevent container breakout exploits (Marchand et al., 2026; Li et al., 2026e). Overcoming verification trust gaps requires static auditors that lift prose into Datalog databases to mathematically prove safety compliance (Wen et al., 2026). Enabling acquisition without triggering catastrophic forgetting requires decoupling learned abstractions via virtual memory paging (Yang et al., 2026; Packer et al., 2023b). Finally, constructing robust, scalable, interactive environments is essential to safely evaluate execution efficiency and safety guardrails under massive distribution scale (Li et al., 2026c).

5 Conclusion

The transition from monolithic prompting and atomic tool use to externalized executable skills represents a critical maturation in the design of autonomous language agents. By decoupling high-level cognitive planning from deterministic, procedural execution, skill-based architectures directly address the context scaling and reliability bottlenecks that plague contemporary LLM systems. This survey has formalized the agentic skill abstraction and provided a unified lifecycle-oriented taxonomy, synthesizing the latest advancements in skill discovery, representation, memory retrieval, orchestration, and governance. As agents are increasingly deployed in complex ecosystems, advancing robust permission models, dynamic adaptation, and standardized evaluation will be paramount. Ultimately, formalizing agent capabilities as modular, secure, and self-improving skills paves the way for the next generation of scalable and reliable autonomous systems.

6 Limitations

While this survey provides a comprehensive synthesis of the agentic skills landscape, several limitations exist. First, the field of autonomous language agents is evolving at a rapid pace; thus, specific frameworks, models, and retrieval implementations discussed may face deprecation as industry standards consolidate. Second, while we focus extensively on digital, software-oriented, and desktop capabilities, the integration of multimodal and embodied robotic skills introduces distinct physical grounding challenges that are only briefly surveyed here. Finally, the evaluation of agentic skills remains an open challenge. Current benchmarks often struggle to capture the open-ended complexity and cascading failure modes of real-world operational environments, meaning the true reliability of these externalized systems in production remains difficult to quantify completely.

References

Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Weiyuan Chen, and Tu Vu. 2026. Evoskill: Automated skill discovery for multi-agent systems. *Preprint*, arXiv:2603.02766.

Varun Pratap Bhardwaj. Skillfortify: Formal analysis and supply chain security for agentic ai skills, 2026.

Shuzhen Bi, Mengsong Wu, Hao Hao, Keqian Li, Wentao Liu, Siyu Song, Hongbo Zhao, and Aimin Zhou. 2026. Automating skill acquisition through large-scale mining of open-source agentic repositories: A framework for multi-agent procedural knowledge extraction. *Preprint*, arXiv:2603.11808.

Samuel Jacob Chacko, James Hugglestone, Chashi Mahiul Islam, and Xiuwen Liu. 2026.

When skills don't help: A negative result on procedural knowledge for tool-grounded agents in offensive cybersecurity. *Preprint*, arXiv:2605.20023.

Le Chen, Erhu Feng, Yubin Xia, and Haibo Chen. 2026a. Skvm: Revisiting language vm for skills across heterogenous llms and harnesses. *Preprint*, arXiv:2604.03088.

Qijia Chen, Andrea Bellucci, Zhida Sun, and Giulio Jacucci. 2026b. Skilldroid: Compile once, reuse forever. *arXiv preprint arXiv:2604.14872*.

Shiqi Chen, Jingze Gai, Ruochen Zhou, Jinghan Zhang, Tongyao Zhu, Junlong Li, Kangrui Wang, Zihan Wang, Zhengyu Chen, Klara Kaleb, and 1 others. 2026c. Skillcraft: Can llm agents learn to use tools skillfully? *arXiv preprint arXiv:2603.00718*.

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. 2026d. Cua-skill: Develop skills for computer using agent. *Preprint*, arXiv:2601.21123.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2024. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*.

Zhihao Chen, Ying Zhang, Yi Liu, Gelei Deng, Yuekang Li, Yanjun Zhang, Jianting Ning, Leo Yu Zhang, Lei Ma, and Zhiqiang Li. 2026e. Credential leakage in llm agent skills: A large-scale empirical study. *Preprint*, arXiv:2604.03070.

Hongcheol Cho, Ryangkyung Kang, and Youngeun Kim. 2026. Skillret: A large-scale benchmark for skill retrieval in llm agents. *Preprint*, arXiv:2605.05726.

Xiang Deng, Yu Gu, and 1 others. 2023. Mind2web: Towards a generalist agent for the web. *arXiv preprint arXiv:2306.06070*.

Jiawei Du, Jinlong Wu, Yuzheng Chen, Yucheng Hu, Bing Li, and Joey Tianyi Zhou. 2025. Rethinking agent design: From top-down workflows to bottom-up skill evolution. *arXiv preprint arXiv:2505.17673*.

Yu Du, Fangyun Wei, and Hongyang Zhang. 2024. Anytool: self-reflective, hierarchical agents for large-scale api calls. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Zijian Du and Nathaniel Pinckney. 2026. Trace2skill: Verifier-guided skill evolution for long-context eda agents. *Preprint*, arXiv:2605.21810.

Zenghao Duan, Yuxin Tian, Zhiyi Yin, Liang Pang, Jingcheng Deng, Zihao Wei, Shicheng Xu, Yuyao Ge, and Xueqi Cheng. 2026. Skillattack: Automated red teaming of agent skills through attack path refinement. *arXiv preprint arXiv:2604.04989*.

Fali Fairy and 1 others. 2026. Graphskill: Documentation-guided hierarchical retrieval-augmented coding for complex graph reasoning. *arXiv preprint arXiv:2603.06620*.

Zhiyuan Fan, Tinghao Yu, Yuanjun Cai, Jiangtao Guan, Yun Yang, Dingxin Hu, Jiang Zhou, Xing Wu, Zhuo Han, Feng Zhang, and 1 others. 2026. Toward scalable terminal task synthesis via skill graphs. *arXiv preprint arXiv:2604.25727*.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2026. Memp: Exploring agent procedural memory. *Preprint*, arXiv:2508.06433.

Yunhao Feng, Yifan Ding, Yingshui Tan, Boren Zheng, Yanming Guo, Xiaolong Li, Kun Zhai, Yishan Li, and Wenke Huang. 2026. Skilltrojan: Backdoor attacks on skill-based agent systems. *arXiv preprint arXiv:2604.06811*.

Yudong Gao, Zongjie Li, Yuanyuanyuan, Zimo Ji, Pingchuan Ma, and Shuai Wang. 2026. Skillreducer: Optimizing llm agent skills for token efficiency. *Preprint*, arXiv:2603.29919.

Zhibin Gou and 1 others. 2023. Critic: Large language models can self-correct with tools. *arXiv preprint arXiv:2305.11738*.

Zihan Guo, Zhiyu Chen, Xiaohang Nie, Jianghao Lin, Yuanjian Zhou, and Weinan Zhang. 2026. Skillprobe: Security auditing for emerging agent skill marketplaces via multi-agent collaboration. *arXiv preprint arXiv:2603.21019*.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tingxu Han, Yi Zhang, Wei Song, Chunrong Fang, Zhenyu Chen, Youcheng Sun, and Lijie Hu. 2026. Swe-skills-bench: Do agent skills actually help in real-world software engineering? *arXiv preprint arXiv:2603.15401*.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. Webvoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890.

Florian Holzbauer, David Schmidt, Gabriel Gegenhuber, Sebastian Schrittwieser, and Johanna Ullrich. 2026. Malicious or not: Adding repository context to agent skill classification. *arXiv preprint arXiv:2603.16572*.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, and 1 others. 2023. Metagpt: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

Yinghan Hou and Zongyou Yang. 2026. Skillsieve: A hierarchical triage framework for detecting malicious ai agent skills. *arXiv preprint arXiv:2604.06550*.

Haichuan Hu, Ye Shang, and Quanjun Zhang. 2026a. Red skills or blue skills? a dive into skills published on clawhub. *arXiv preprint arXiv:2604.13064*.

Qisheng Hu, Quanyu Long, and Wenya Wang. 2026b. When continual learning moves to memory: A study of experience reuse in llm agents. *Preprint*, arXiv:2604.27003.

Yuepeng Hu, Yuqi Jia, Mengyuan Li, Dawn Song, and Neil Gong. 2026c. Maltool: Malicious tool attacks on llm agents. *Preprint*, arXiv:2602.12194.

Xu Huang, Junwu Chen, Yuxing Fei, Zhuohan Li, Philippe Schwaller, and Gerbrand Ceder. 2026. Cascade: Cumulative agentic skill creation through autonomous development and evolution. *Preprint*, arXiv:2512.23880.

Xiaojun Jia, Jie Liao, Simeng Qin, Jindong Gu, Wenqi Ren, Xiaochun Cao, Yang Liu, and Philip Torr. 2026. Skillject: Automating stealthy skill-based prompt injection for coding agents with trace-driven closed-loop refinement. In *The 6th Workshop of Adversarial Machine Learning on Computer Vision: Safety of Vision-Language Agents*.

Guanyu Jiang, Zhaochen Su, Xiaoye Qu, and Yi R Fung. 2026a. Xskill: Continual learning from experience and skills in multimodal agents. *arXiv preprint arXiv:2603.12056*.

Yukun Jiang, Yage Zhang, Michael Backes, Xinyue Shen, and Yang Zhang. 2026b. Harmfulskillbench: How do harmful skills weaponize your agents? *Preprint*, arXiv:2604.15415.

Zhengbo Jiao, Shaobo Wang, Zifan Zhang, Xuan Ren, Wei Wang, Bing Zhao, Hu Wei, and Linfeng Zhang. 2026. Agentic proposing: Enhancing large language model reasoning via compositional skill synthesis. *Preprint*, arXiv:2602.03279.

Carlos E Jimenez, John Yang, Alexander Wettinger, and 1 others. 2024. Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations (ICLR)*.

Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, and 1 others. 2022. Mrkl systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *arXiv preprint arXiv:2205.00445*.

Sehoon Kim and 1 others. 2023. An llm compiler for parallel function calling. *arXiv preprint arXiv:2312.04511*.

Hao Li and 1 others. 2026a. Organizing, orchestrating, and benchmarking agent skills at ecosystem scale. *arXiv preprint arXiv:2603.02176*.

Hongxin Li, Xiping Wang, Jingran Su, Zheng Ju, Yuntao Chen, Qing Li, and Zhaoxiang Zhang. 2026b. Autogui-v2: A comprehensive multi-modal gui functionality understanding benchmark. *Preprint*, arXiv:2604.24441.

Xiangyi Li, Wenbo Chen, Yimin Liu, Shenghan Zheng, Xiaokun Chen, Yifeng He, Yubo Li, Bingran You, Haotian Shen, Jiankai Sun, Shuyi Wang, Binxu Li, Qunhong Zeng, Di Wang, Xuandong Zhao, Yuanli Wang, Roey Ben Chaim, Zonglin Di, Yipeng Gao, and 22 others. 2026c. Skillsbench: Benchmarking how well agent skills work across diverse tasks. *Preprint*, arXiv:2602.12670.

Xiaoxiao Li. 2026. When single-agent with skills replace multi-agent systems and when they fail. *arXiv preprint arXiv:2601.04748*.

Yu Li, Rui Miao, Zhengling Qi, and Tian Lan. 2026d. Arise: Agent reasoning with intrinsic skill evolution in hierarchical reinforcement learning. *Preprint*, arXiv:2603.16060.

Yuyang Li, Yiran Dou, Jie-Jing Shao, Yueming Lyu, Ivor Tsang, and Haiyan Yin. Skilltracer: Structural failure attribution and refinement of agentic skills in long-horizon web tasks. In *Workshop on Multi-Agent Learning and Its Opportunities in the Era of Generative AI*.

Zhiyuan Li, Jingzheng Wu, Xiang Ling, Xing Cui, and Tianyue Luo. 2026e. Towards secure agent skills: Architecture, threat taxonomy, and security analysis. *Preprint*, arXiv:2604.02837.

Zhiyuan Li, Jingzheng Wu, Xiang Ling, Xing Cui, and Tianyue Luo. 2026f. Towards secure agent skills: Architecture, threat taxonomy, and security analysis. *arXiv preprint arXiv:2604.02837*.

Yuan Liang, Ruobin Zhong, Haoming Xu, Chen Jiang, Yi Zhong, Runnan Fang, Jia-Chen Gu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Xin Xu, Tongtong Wu, Kun Wang, Yang Liu, Zhen Bi, Jungang Lou, Yuchen Eleanor Jiang, Hangcheng Zhu, and 30 others. 2026. Skillnet: Create, evaluate, and connect ai skills. *Preprint*, arXiv:2603.04448.

George Ling, Shanshan Zhong, and Richard Huang. 2026. Agent skills: A data-driven analysis of claude skills for extending large language model functionality. *arXiv preprint arXiv:2602.08004*.

Anthony Zhe Liu, Jongwook Choi, Sungryull Sohn, Yao Fu, Jaekyeom Kim, Dong-Ki Kim, Xinhe Wang, Jaewon Yoo, and Honglak Lee. 2024. Skillact: Using skill abstractions improves llm agents. In *ICML 2024 Workshop on LLMs and Cognition*.

Chang Liu, Sibo Tian, Xiao Liang, and Minghui Zheng. 2026a. Self-vla: A skill enhanced agentic vision-language-action framework for contact-rich disassembly. *Preprint*, arXiv:2603.11080.

Dawei Liu, Zongxia Li, Hongyang Du, Xiyang Wu, Shihang Gui, Yongbei Kuang, and Lichao Sun. 2026b. Graph of skills: Dependency-aware structural retrieval for massive agent skills. *Preprint*, arXiv:2604.05333.

Jiateng Liu, Zhenhailong Wang, Rushi Wang, Bingxuan Li, Jeonghwan Kim, Aditi Tiwari, Pengfei Yu, Denghui Zhang, and Heng Ji. 2026c. Osexpert: Computer-use agents learning professional skills via exploration. *arXiv preprint arXiv:2603.07978*.

Ruibo Liu, Jason Wei, Shixiang Shane Gu, Te-Yen Wu, Soroush Vosoughi, Claire Cui, Denny Zhou, and Andrew M. Dai. 2023. Mind's eye: Grounded language model reasoning through simulation. In *The Eleventh International Conference on Learning Representations*.

Shunyu Liu, Yaoru Li, Kongcheng Zhang, Zhenyu Cui, Wenkai Fang, Yuxuan Zheng, Tongya Zheng, and Mingli Song. 2025. Odyssey: Empowering minecraft agents with open-world skills. *Preprint*, arXiv:2407.15325.

Yi Liu, Zhihao Chen, Yanjun Zhang, Gelei Deng, Yuekang Li, Jianting Ning, Ying Zhang, and Leo Yu Zhang. 2026d. Malicious agent skills in the wild: A large-scale security empirical study. *arXiv preprint arXiv:2605.08115*.

Yi Liu, Weizhe Wang, Ruitao Feng, Yao Zhang, Guangquan Xu, Gelei Deng, Yuekang Li, and Leo Zhang. 2026e. Agent skills in the wild: An empirical study of security vulnerabilities at scale. *Preprint*, arXiv:2601.10338.

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

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

Zijian Lu, Yiping Zuo, Yupeng Nie, Xin He, Weibei Fan, Lianyong Qi, and Shi Jin. 2026b. Contractskill: Repairable contract-based skills for multimodal web agents. *Preprint*, arXiv:2603.20340.

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, Rongcheng Tu, Xiao Luo, Wei Ju, Zhiping Xiao, Yifan Wang, Meng Xiao, Chenwu Liu, Jingyang Yuan, Shichang Zhang, and 7 others. 2025. Large language model agent: A survey on methodology, applications and challenges. *Preprint*, arXiv:2503.21460.

Lijia Lv, Xuehai Tang, Jie Wen, Jizhong Han, and Songlin Hu. 2026. Structured security auditing and robustness enhancement for untrusted agent skills. *arXiv preprint arXiv:2604.25109*.

Yuchen Ma, Yue Huang, Han Bao, Haomin Zhuang, Swadheen Shukla, Michel Galley, Xiangliang Zhang, and Stefan Feuerriegel. 2026a. Skillgen: Verified inference-time agent skill synthesis. *Preprint*, arXiv:2605.10999.

Ziyu Ma, Shidong Yang, Yuxiang Ji, Xucong Wang, Yong Wang, Yiming Hu, Tongwen Huang, and Xiangxiang Chu. 2026b. Skillclaw: Let skills evolve collectively with agentic evolver. *arXiv preprint arXiv:2604.08377*.

Aman Madaan, Niket Tandon, Prakhar Gupta, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2305.00811*.

Lars Malmqvist. 2024. Sycophancy in large language models: Causes and mitigations. *Preprint*, arXiv:2411.15287.

Narek Maloyan and Dmitry Namiot. 2026. Prompt injection attacks on agentic coding assistants: A systematic analysis of vulnerabilities in skills, tools, and protocol ecosystems. *International Journal of Open Information Technologies*, 14(2):1–10.

Rahul Marchand, Art O Cathain, Jerome Wynne, Philippos Maximos Giavridis, Sam Deverett, John Wilkinson, Jason Gwartz, and Harry Coppock. 2026. Quantifying frontier llm capabilities for container sandbox escape. *Preprint*, arXiv:2603.02277.

Qirui Mi, Zhijian Ma, Mengyue Yang, Haoxuan Li, Yisen Wang, Haifeng Zhang, and Jun Wang. 2026. Skill-pro: Learning reusable skills from experience via non-parametric ppo for llm agents. *arXiv preprint arXiv:2602.01869*.

Gregoire Mialon and 1 others. 2023. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*.

Microsoft. 2024. Prompt flow: Build high-quality llm apps from prototyping to production.

Bowen Yu Feifan Song Hangyu Li Haiyang Yu Zhoujun Li Fei Huang Yongbin Li Minghao Li, Yingxiu Zhao. 2023. Api-bank: A comprehensive benchmark for tool-augmented llms. *arXiv preprint arXiv:2304.08244*.

Jingwei Ni, Yihao Liu, Xinpeng Liu, Yutao Sun, Mengyu Zhou, Pengyu Cheng, Dexin Wang, Erchao Zhao, Xiaoxi Jiang, and Guanjun Jiang. 2026. Trace2skill: Distill trajectory-local lessons into transferable agent skills. *arXiv preprint arXiv:2603.25158*.

Zheng Nie, Ruolin Shen, Xinlei Yu, Bo Yin, Jiangning Zhang, and Xiaobin Hu. 2026. Skillgraph: Self-evolving multi-agent collaboration with multimodal graph topology. *arXiv preprint arXiv:2604.17503*.

Siru Ouyang, Jun Yan, Yanfei Chen, Rujun Han, Zifeng Wang, Bhavana Dalvi Mishra, Rui Meng, Chun-Liang Li, Yizhu Jiao, Kaiwen Zha, and 1 others. 2026. Skillos: Learning skill curation for self-evolving agents. *arXiv preprint arXiv:2605.06614*.

Charles Packer, Vivian Fang, Shishir G Patil, Kevin Lin, Sarah Wooders, and Joseph E Gonzalez. 2023a. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*.

Charles Packer and 1 others. 2023b. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*.

Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 1–22.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2024. Gorilla: Large language model connected with massive apis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37.

Hongji Pu, Xinyuan Song, and Liang Zhao. 2026. Skillops: Managing llm agent skill libraries as self-maintaining software ecosystems. *Preprint*, arXiv:2605.13716.

Denys Pushkin and Emmanuel Abbe. 2026. Lead: Breaking the no-recovery bottleneck in long-horizon reasoning. *Preprint*, arXiv:2603.06870.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, and 1 others. 2024. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 15174–15186.

Yubin Qu, Yi Liu, Tongcheng Geng, Gelei Deng, Yuekang Li, Leo Yu Zhang, Ying Zhang, and Lei Ma. 2026. Supply-chain poisoning attacks against llm coding agent skill ecosystems. *arXiv preprint arXiv:2604.03081*.

Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy P Lillicrap. 2023. Androidinthewild: A large-scale dataset for android device control. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Sampriti Saha and Pranav Hemanth. 2026. Skilldex: A package manager and registry for agent skill packages with hierarchical scope-based distribution. *arXiv preprint arXiv:2604.16911*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

David Schmotz, Sahar Abdelnabi, and Maksym Andriushchenko. Agent skills enable a new class of realistic and trivially simple prompt injections.

David Schmotz, Sahar Abdelnabi, and Maksym Andriushchenko. 2025. Agent skills enable a new class of realistic and trivially simple prompt injections. *arXiv preprint arXiv:2510.26328*.

David Schmotz, Luca Beurer-Kellner, Sahar Abdelnabi, and Maksym Andriushchenko. 2026. Skill-inject: Measuring agent vulnerability to skill file attacks. *arXiv preprint arXiv:2602.20156*.

Shuaike Shen, Wenduo Cheng, Mingqian Ma, Alistair Turcan, Martin Jinye Zhang, and Jian Ma. 2026. Skillfoundry: Building self-evolving agent skill libraries from heterogeneous scientific resources. *arXiv preprint arXiv:2604.03964*.

Haochen Shi, Xingdi Yuan, and Bang Liu. 2026. Evolving programmatic skill networks. *Preprint*, arXiv:2601.03509.

Tianmin Shu, Caiming Xiong, and Richard Socher. 2017. Hierarchical and interpretable skill acquisition in multi-task reinforcement learning. *Preprint*, arXiv:1712.07294.

Yifan Song and 1 others. 2403. Taskbench: Evaluating tool use and task planning capabilities in llm agents. *arXiv preprint arXiv:2403.09115*.

Weihang Su, Jianming Long, Qingyao Ai, Yichen Tang, Changyue Wang, Yiteng Tu, and Yiqun Liu. 2026. Skill retrieval augmentation for agentic ai. *arXiv preprint arXiv:2604.24594*.

Theodore Sumers, Shunyu Yao, Karthik R Narasimhan, and Thomas L. Griffiths. 2024. Cognitive architectures for language agents. *Transactions on Machine Learning Research*. Survey Certification, Featured Certification.

Yiqun Sun, Pengfei Wei, and Lawrence B. Hsieh. 2026. Don't retrieve, navigate: Distilling enterprise knowledge into navigable agent skills for qa and rag. *Preprint*, arXiv:2604.14572.

Didac Suris, Ruoshi Gupta, and Carl Vondrick. 2023. Vipergpt: Visual grounding via python execution. In *IEEE International Conference on Computer Vision (ICCV)*.

Romal Thoppilan, Daniel De Freitas, and 1 others. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Guiyao Tie, Jiawen Shi, Pan Zhou, and Lichao Sun. 2026. Badskill: Backdoor attacks on agent skills via model-in-skill poisoning. *Preprint*, arXiv:2604.09378.

Songjun Tu, Chengdong Xu, Qichao Zhang, Yaocheng Zhang, Xiangyuan Lan, Linjing Li, Dong Li, and Dongbin Zhao. 2026. Dynamic dual-granularity skill bank for agentic rl. *Preprint*, arXiv:2603.28716.

Georgios Tziafas and Hamidreza Kasaei. 2024. Lifelong robot library learning: Bootstrapping composable and generalizable skills for embodied control with language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 515–522. IEEE.

Yash Vishe, Rohan Surana, Xunyi Jiang, Zihan Huang, Xintong Li, Nikki Lijing Kuang, Tong Yu, Ryan A. Rossi, Jingbo Shang, Julian McAuley, and Junda Wu. 2026. Skill-r1: Agent skill evolution via reinforcement learning. *Preprint*, arXiv:2605.09359.

Chenxi Wang, Zhuoyun Yu, Xin Xie, Wuguannan Yao, Runnan Fang, Shuofei Qiao, Kexin Cao, Guozhou Zheng, Xiang Qi, Peng Zhang, and 1 others. 2026a. Skillx: Automatically constructing skill knowledge bases for agents. *arXiv preprint arXiv:2604.04804*.

Fali Wang, Chenglin Weng, Xianren Zhang, Siyuan Hong, Hui Liu, and Suhang Wang. 2026b. Graphskill: Documentation-guided hierarchical retrieval-augmented coding for complex graph reasoning. *Preprint*, arXiv:2603.06620.

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

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, and Xiaoxin Chen. 2026c. Skill-conditioned self-distillation for multi-turn llm agents. *arXiv preprint arXiv:2604.10674*.

Jiayu Wang, Yifei Ming, Zixuan Ke, Shafiq Joty, Aws Albarghouthi, and Frederic Sala. 2026d. Skillorchestra: Learning to route agents via skill transfer. *Preprint*, arXiv:2602.19672.

Jiayu Wang, Yifei Ming, Zixuan Ke, Shafiq Joty, Aws Albarghouthi, and Frederic Sala. 2026e. Skillorchestra: Learning to route agents via skill transfer. *arXiv preprint arXiv:2602.19672*.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. 2025a. Reinforcement learning for self-improving agent with skill library. *arXiv preprint arXiv:2512.17102*.

Qianli Wang, Boyang Ma, Minghui Xu, and Yue Zhang. 2026f. When skills lie: Hidden-comment injection in llm agents. *arXiv preprint arXiv:2602.10498*.

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

Xinyu Jessica Wang, Haoyue Bai, Yiyou Sun, Haorui Wang, Shuibai Zhang, Wenjie Hu, Mya Schroder, Bilge Mutlu, Dawn Song, and Robert D Nowak. 2026g. The long-horizon task mirage? diagnosing where and why agentic systems break. *Preprint*, arXiv:2604.11978.

Zimu Wang, Yuling Shi, Mengfan Li, Zijun Liu, Jie M. Zhang, Chengcheng Wan, and Xiaodong Gu. 2026h. Effiskill: Agent skill based automated code efficiency optimization. *Preprint*, arXiv:2603.27850.

Zora Zhiruo Wang, Apurva Gandhi, Graham Neubig, and Daniel Fried. 2025b. Inducing programmatic skills for agentic tasks. *arXiv preprint arXiv:2504.06821*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hongbo Wen, Ying Li, Hanzhi Liu, Chaofan Shou, Yanju Chen, Yuan Tian, and Yu Feng. 2026. Semia: Auditing agent skills via constraint-guided representation synthesis. *Preprint*, arXiv:2605.00314.

Zikai Alex Wen. 2026. Toward user comprehension supports for LLM agent skill specifications. In *First Workshop on Agent Skills*.

Niklas Wretblad, Fredrik Gordh Riseby, Rahul Biswas, Amin Ahmadi, and Oskar Holmström. 2024. Understanding the effects of noise in text-to-sql: An examination of the bird-bench benchmark. *Preprint*, arXiv:2402.12243.

Haolin Wu, Yuecheng Liu, Junyi Dong, Heng Zhang, Sitong Mao, Hesheng Wang, Weigang Wu, and Shunbo Zhou. 2025. Ascent: Autonomous skill learning toward complex embodied tasks with foundation models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16752–16758.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Riley Zhang, Jian Liu, and 1 others. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*.

Xiyang Wu, Zongxia Li, Guangyao Shi, Alexander Duffy, Tyler Marques, Matthew Lyle Olson, Tianyi Zhou, and Dinesh Manocha. 2026. Co-evolving llm decision and skill bank agents for long-horizon tasks. *Preprint*, arXiv:2604.20987.

Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. 2025. Demystifying llm-based software engineering agents. *Proc. ACM Softw. Eng.*, 2(FSE).

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, and 1 others.

2026a. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*.

Tianle Xia, Lingxiang Hu, Yiding Sun, Ming Xu, Lan Xu, Siying Wang, Wei Xu, and Jie Jiang. 2026b. Grasp: Graph-structured skill compositions for llm agents. *arXiv preprint arXiv:2604.17870*.

Tianle Xia, Lingxiang Hu, Yiding Sun, Ming Xu, Lan Xu, Siying Wang, Wei Xu, and Jie Jiang. 2026c. Grasp: Graph-structured skill compositions for llm agents. *Preprint*, arXiv:2604.17870.

Wenjie Xiao, Xuehai Tang, Biyu Zhou, Songlin Hu, and Jizhong Han. 2026. Routeguard: Internal-signal detection of skill poisoning in llm agents. *arXiv preprint arXiv:2604.22888*.

Senwei Xie, Yuntian Zhang, Ruiping Wang, and Xilin Chen. 2026. Uni-skill: Building self-evolving skill repository for generalizable robotic manipulation. *Preprint*, arXiv:2603.02623.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Preprint*, arXiv:2404.07972.

Hanwen Xing, Haomin Zhuang, Xuandong Zhao, Yue Huang, Zhenheng Tang, and Xiangliang Zhang. 2026. Recipes for agents: Understanding skills and their open questions. *Preprint, ResearchGate. doi*, 10.

Linyu Li Ganghong Huang Jianfeng Liu Honglin Qiao Xingyan Liu, Xiyue Luo. 2026. Forging domain-specific, self-evolving agent skills in cloud technical support. *arXiv preprint arXiv:2604.08618*.

Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. 2023. Xskill: Cross embodiment skill discovery. *Preprint*, arXiv:2307.09955.

Yangjie Xu, Lujun Li, Lama Sleem, Niccolo Gentile, Yewei Song, Yiqun Wang, Siming Ji, Wenbo Wu, and Radu State. 2026a. Agent skill framework: Perspectives on the potential of small language models in industrial environments. *Preprint*, arXiv:2602.16653.

Zishan Xu, Yifu Guo, Yuquan Lu, Fengyu Yang, Zhiyuan Yao, Jiaye Lin, Ruyi Gong, and Lihua Cai. 2026b. Skillevo: An experience learning framework with reinforcement learning for skill evolution.

John Yang and 1 others. 2024. Swe-agent: Agent-computer interfaces for resolving real-world software engineering problems. *arXiv preprint arXiv:2405.01115*.

Yongjin Yang, Sinjae Kang, Juyong Lee, Dongjun Lee, Se-Young Yun, and Kimin Lee. 2025a. Automated skill discovery for language agents through exploration and iterative feedback. *Preprint*, arXiv:2506.04287.

Yutao Yang, Junsong Li, Qianjun Pan, Bihao Zhan, Yuxuan Cai, Lin Du, Jie Zhou, Kai Chen, Qin Chen, Xin Li, Bo Zhang, and Liang He. 2026. Autoskill: Experience-driven lifelong learning via skill self-evolution. *Preprint*, arXiv:2603.01145.

Zonghan Yang, Shengjie Wang, Kelin Fu, Wenyang He, Weimin Xiong, Yibo Liu, Yibo Miao, Bofei Gao, Yejie Wang, Yingwei Ma, and 1 others. 2025b. Kimi-dev: Agentless training as skill prior for swe-agents. *arXiv preprint arXiv:2509.23045*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Haoran Ye, Xuning He, Vincent Arak, Haonan Dong, and Guojie Song. 2026. Meta context engineering via agentic skill evolution. *arXiv preprint arXiv:2601.21557*.

Simon Yu, Gang Li, Weiyan Shi, and Peng Qi. 2025. Polyskill: Learning generalizable skills through polymorphic abstraction. *arXiv preprint arXiv:2510.15863*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Haoqi Yuan, Chi Zhang, Hongcheng Wang, Feiyang Xie, Penglin Cai, Hao Dong, and Zongqing Lu. 2023. Skill reinforcement learning and planning for open-world long-horizon tasks. *arXiv preprint arXiv:2303.16563*.

Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Yongliang Shen, Ren Kan, Dongsheng Li, and Deqing Yang. 2024. Easytool: Enhancing llm-based agents with concise tool instruction. *Preprint*, arXiv:2401.06201.

Renos Zabounidis, Yue Wu, Simon Stepputtis, Woojun Kim, Yuanzhi Li, Tom Mitchell, and Katia Sycara. 2026. Scalar: Learning and composing skills through llm guided symbolic planning and deep rl grounding. *Preprint*, arXiv:2603.09036.

Guijia Zhang, Shu Yang, Xilin Gong, and Di Wang. 2026a. Stars: Skill-triggered audit for request-conditioned invocation safety in agent systems. *arXiv preprint arXiv:2604.10286*.

Hanrong Zhang, Shicheng Fan, Henry Peng Zou, Yankai Chen, Zhenting Wang, Jiayu Zhou, Chengze Li, Wei-Chieh Huang, Yifei Yao, Kening Zheng, Xue Liu, Xiaoxiao Li, and Philip S. Yu. 2026b. Coevoskills: Self-evolving agent skills via co-evolutionary verification. *Preprint*, arXiv:2604.01687.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. 2026c. Memskill: Learning and evolving memory skills for self-evolving agents. *arXiv preprint arXiv:2602.02474*.

Xing Zhang, Guanghui Wang, Yanwei Cui, Wei Qiu, Ziyuan Li, Bing Zhu, and Peiyang He. 2026d. Experience compression spectrum: Unifying memory, skills, and rules in llm agents. *Preprint*, arXiv:2604.15877.

Ziao Zhang, Kou Shi, Shiting Huang, Avery Nie, Yu Zeng, Yiming Zhao, Zhen Fang, Qishen Su, Haibo Qiu, Wei Yang, and 1 others. 2026e. Skillflow: Benchmarking lifelong skill discovery and evolution for autonomous agents. *arXiv preprint arXiv:2604.17308*.

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

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

Chenyu Zhou, Huacan Chai, Wenteng Chen, Zihan Guo, Rong Shan, Yuanyi Song, Tianyi Xu, Yingxuan Yang, Aofan Yu, Weiming Zhang, Congming Zheng, Jiachen Zhu, Zeyu Zheng, Zhuosheng Zhang, Xingyu Lou, Changwang Zhang, Zhihui Fu, Jun Wang, Weiwen Liu, and 2 others. 2026a. Externalization in llm agents: A unified review of memory, skills, protocols and harness engineering. *Preprint*, arXiv:2604.08224.

Huichi Zhou, Siyuan Guo, Anjie Liu, Zhongwei Yu, Ziqin Gong, Bowen Zhao, Zhixun Chen, Menglong Zhang, Yihang Chen, Jinsong Li, and 1 others. 2026b. Memento-skills: Let agents design agents. *arXiv preprint arXiv:2603.18743*.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. *Preprint*, arXiv:2307.13854.

Jiaying Zhu, Lyuye Zhang, Wenbo Guo, and Yang Liu. 2026. Skillclone: Multi-modal clone detection and clone propagation analysis in the agent skill ecosystem. *arXiv preprint arXiv:2603.22447*.

Haomin Zhuang, Hanwen Xing, and Xiangliang Zhang. 2026. Agentclick: A skill-based human-in-the-loop review layer for terminal ai agents. *Preprint*, arXiv:2604.16520.

Li Yu Weikang Zhou Zhaoyang Liu Bolin Ding Hai Zhao Zouying Cao, Jiaji Deng. 2025. Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution. *arXiv preprint arXiv:2512.10696*.

A Taxonomy of Skills

A.1 Instructional Skills

Instructional skills translate descriptive instruction manuals into executive checklists restricting the model's action space to predictable trajectories (Bi et al., 2026; Yuan et al., 2024).

A.2 Tool-Execution Skills

Tool-execution skills induce immediate state updates by invoking external tools required for a goal (Schick et al., 2023). Categories include API routines (Patil et al., 2024; Minghao Li, 2023), IDE modifications (Yang et al., 2024), Browser DOM-parsing (Deng et al., 2023), and Database querying (Yu et al., 2018; Wretblad et al., 2024).

A.3 Reasoning Skills

Reasoning skills map the state to a latent reasoning trace when multi-step inference is required. These coordinate the internal reasoning steps, self-correction pathways, and verification gates autonomously (Sumers et al., 2024; Gou et al., 2023; Madaan et al., 2023).

A.4 Domain-Specific Skills

A.4.1 Software Engineering and Desktop Skills

Skills are heavily utilized for resolving bugs (Yang et al., 2024; Xia et al., 2025) and automated code efficiency optimization (Wang et al., 2026h). For computer-using agents, dedicated frameworks like CUA-Skill develop specialized routines for traversing complex desktop operating systems and file hierarchies securely (Chen et al., 2026d).

A.4.2 Scientific and Research Skills

Capabilities automate literature reviews across public databases (Mialon et al., 2023), perform visual-grounded data exploration (Suris et al., 2023; Gupta and Kembhavi, 2023), and ground reasoning in interactive scientific simulators (Liu et al., 2023; Wang et al., 2022).

A.4.3 Robotics and Embodied Agents

In embodied domains, cross-embodiment discovery enables robotic systems to transfer learned skills across heterogeneous hardware boundaries natively (Xu et al., 2023). Specialized frameworks leverage agentic vision-language-action models to execute contact-rich physical tasks (Liu et al., 2026a), while self-evolving capability repositories drive generalizable robotic manipulation across open-world environments (Xie et al., 2026).

A.4.4 Open-World and Gaming Environments

In unconstrained gaming environments like Minecraft, frameworks empower agents with survival capabilities through expansive, open-world skill libraries (Liu et al., 2025; Yuan et al., 2023).

A.4.5 Collaborative and Meta-Skills

Meta-skills are advanced routines operating directly over the capability space to diagnose deficiencies and rewrite existing files to continuously self-improve (Yang et al., 2026; Zhang et al., 2026b; Zhou et al., 2026b; Yu et al., 2025; Xingyan Liu, 2026).

B Skill Ecosystems and Marketplaces

The rapid maturation of the agent skills abstraction has catalyzed the development of specialized, domain-specific ecosystems and public marketplaces. These platforms package procedural expertise to solve complex, long-horizon tasks across diverse computer environments and interactive simulators.

B.1 Desktop Operating Systems

Desktop operating systems represent a mature application environment for agentic skills, enabling models to execute open-ended administrative tasks across complex file systems and local software applications (Xie et al., 2024; Chen et al., 2026d; Liu et al., 2026c). In these setups, agents must coordinate visual observations and accessibility tree states to execute deterministic mouse clicks and keyboard typing routines. State-of-the-art computer-use agents are evaluated on real Ubuntu environments via the OSWorld benchmark, which measures task completion rates across diverse software environments.

B.2 Mobile Operating Systems

Mobile operating systems present a highly dynamic, touch-centric environment where agents must automate complex task sequences on real smartphones (Rawles et al., 2023). To navigate these devices, mobile GUI agents translate natural language user intents into sequences of swipe, scroll, tap, and back buttons (Li et al., 2026b). These mobile automation capabilities are validated on large-scale crowdsourced datasets like AndroidInTheWild, testing the agent's ability to operate diverse apps under dynamic layout variations (Rawles et al., 2023).

B.3 Web Navigation

Web navigation represents a massive, open-world ecosystem where agents leverage specialized DOM-parsing and HTML-filtering skills to browse websites (Deng et al., 2023). In these scenarios, generalist web agents are prompted to scroll, click, and input text across real-world, complex e-commerce platforms and search engines (He et al., 2024). Web-use capabilities are validated on robust benchmarks like WebArena, testing the model's robustness under stateful, dynamic web page modifications (Zhou et al., 2024).

B.4 Software Engineering

Software engineering is the most demanding domain for agentic computing, requiring the coordination of file navigation, code parsing, compiler interaction, and patch generation (Jimenez et al., 2024). Rather than relying on simple, stateless code completion, software agents use packaged IDE skills to resolve real-world software issues autonomously (Yang et al., 2024). These capabilities are rigorously evaluated on repository-level codebases like SWE-bench, measuring functional correctness via execution-based test verification (Jimenez et al., 2024).

B.5 Scientific Exploration

Scientific exploration represents an emerging frontier where agents deploy specialized skills to plan experiments and reason about physical and chemical laws (Wang et al., 2022). In these settings, agents are integrated with interactive text environments and symbolic physics simulators to perform hypothesis testing and analyze experimental outcomes (Wang et al., 2022). Grounding language reasoning in interactive simulations has been shown to improve the success rate of scientific exploration agents, paving the way for autonomous laboratory assistants (Wang et al., 2022).