When More Skills Hurt: A Survey and Empirical Study of Skill Scaling in LLM Agents

Anonymous ACL submission

Abstract

Skills have emerged as an important abstraction for improving the efficiency and reliability of LLM agents by externalizing reusable procedural knowledge. As agent systems become increasingly autonomous, skills are continuously accumulated, organized, composed, shared, and internalized across tasks and agents. This shift raises a new challenge: how skill ecosystems scale. In this paper, we present a survey and empirical study of skill scaling. We organize existing work into three complementary scaling dimensions: temporal scaling, which studies how skills evolve over time; structural scaling, which studies how large skill repositories are organized and composed; and collective scaling, which studies how skills are shared across agents and users. We further conduct empirical studies across these scaling settings to analyze how expanding skill ecosystems affect agent behavior and performance. Our results suggest that skill scaling is inherently non-monotonic: while larger and more diverse skill ecosystems can improve task success and execution efficiency, they may also introduce redundancy, retrieval ambiguity, cross-task interference, and coordination instability.

1 Introduction

Large Language Model (LLM) agents are increasingly applied to complex, long-horizon tasks. A common paradigm is to solve such tasks through step-by-step online reasoning(Yao et al., 2022b; Shinn et al., 2023). However, this reliance on transient reasoning is costly and fragile: early errors can propagate, while similar subproblems often trigger redundant re-planning.

Recent systems introduce skills as reusable units of procedural knowledge that reduce redundant planning and improve execution efficiency across related tasks(Hu et al., 2025; Xu and Yan, 2026). However, static skill reuse alone becomes insufficient as agents encounter new tasks, environments, and failure patterns over time. To remain effective, skills must be continually expanded, adapted, and refined, naturally leading to **skill self-evolution**, where agents autonomously discover, synthesize, and improve skills from interaction trajectories and execution feedback(Yang et al., 2026; Xia et al., 2026). Through this process, skill systems shift from fixed collections of procedures toward dynamic capability ecosystems that continuously grow and adapt. As skills become increasingly accumulated, updated, composed, and shared, the central challenge also shifts from acquiring individual skills to managing large evolving skill ecosystems(Liang et al., 2026).

Such evolving skill ecosystems may face issues such as redundancy, routing instability, dependency conflicts, retrieval overhead, and noisy synchronization across large skill repositories. This naturally raises a fundamental question: **What happens when skills scale?**

These challenges are not isolated, but reflect deeper systemic effects arising from different aspects of the scaling process. Looking beyond individual skill generation, we characterize this phenomenon through three dimensions of skill scaling: temporal evolution, structural composition, and collective sharing.

Temporal Scaling studies how skill systems evolve over time as agents continually accumulate and refine reusable procedural knowledge through interaction and self-improvement. **Structural Scaling** studies how agents organize, retrieve, and compose increasingly diverse skills to solve more complex tasks requiring coordinated procedural execution. **Collective Scaling** studies how skills are shared, transferred, and synchronized across multiple agents, enabling distributed experience accumulation and collaborative capability growth.

To systematically study skill scaling in LLM agents, we present a unified survey and evaluation framework that analyzes how procedural knowl-

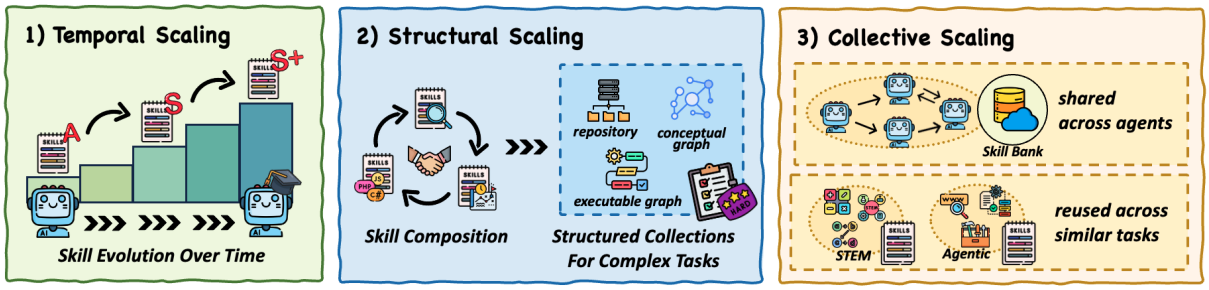

Figure 1: Three dimensions of skill scaling

edge evolves, composes, and propagates across increasingly large-scale skill ecosystems. Our contributions are summarized as follows:

- We introduce a unified perspective on skill scaling in LLM agents and organize existing research into three complementary dimensions: temporal scaling, structural scaling, and collective scaling.
- We conduct empirical studies across all three scaling dimensions to analyze how expanding skill ecosystems affect agent behavior, capability, and coordination.
- We analyze common scaling challenges observed across existing skill-based agent systems.

2 Foundations of Agent Skills

To support reusable procedural capabilities, recent studies have developed increasingly sophisticated skill mechanisms. In this section, we introduce the conceptual foundations of agent skills, including their definition, organization, acquisition, and routing mechanisms.

2.1 What is a Skill?

In LLM-based agents, a *skill* refers to a reusable unit of procedural knowledge that helps an agent solve a class of tasks more reliably than relying on ad-hoc reasoning or isolated tool calls(Zhou et al., 2026; Xu and Yan, 2026; Li et al., 2026). While early agent systems mostly extended LLMs through prompting, retrieval, or function calling(Schick et al., 2023; Qin et al., 2024), skills introduce a higher-level abstraction: they package not only what capability is available, but also when it should be invoked, how it should be executed, what resources it depends on, and how success or failure should be judged.

A typical skill includes activation metadata, executable procedural guidance, optional supporting resources, and reliability constraints or validation criteria. This view is consistent with recent filesystem-based skill standards, where a skill is often materialized as a directory centered on a `SKILL.md` file, optionally bundled with executable code and supporting assets (Li et al., 2025; Xu and Yan, 2026; Chen et al., 2026a; Li et al., 2026). It also generalizes beyond hand-authored skill files to learned or distilled skills, where reusable strategies are extracted from trajectories, failures, demonstrations, or domain corpora (Yang et al., 2026; Xia et al., 2026; Zhang et al., 2026b).

Skills differ from several related conceptsas summarized in Appendix Table 4. **Tools** expose atomic functionalities with defined inputs and execution outputs, whereas skills organize how, when, and in what sequence such capabilities should be used to accomplish broader tasks(Schick et al., 2023; Qin et al., 2024; Patil et al., 2024; Qian et al., 2026). **Workflows** encode relatively fixed control structures for recurring processes, while skills encode relatively fixed control structures for recurring processes(Yao et al., 2022b; Wang et al., 2023; Zhang et al., 2025; Hong et al., 2025). **Memory** stores contextual experience and historical interactions, whereas skills further encode relatively fixed control structures for recurring processes(Packer et al., 2023; Yan et al., 2025; Chhikara et al., 2025).

2.2 Skill Organization

Skill organization operates at two levels. At the individual level, skill representation defines how procedural knowledge is externalized into an artifact that an agent can inspect, route and execute. At the collection level, skill libraries define how multiple skills are stored, related and composed so that agents can reuse procedural knowledge at scale.

2.2.1 Skill Representation

Skill representation specifies how procedural knowledge is externalized into a form that agents can retrieve, interpret, and execute. Existing systems range from human-readable `SKILL.md`-style documents with metadata, instructions, constraints, and supporting resources, to structured forms such as parameterized functions, execution graphs, workflow templates, and composition graphs (Liang et al., 2026; Chen et al., 2026b; Jiao et al., 2026; Shao et al., 2026). Recent work also explores compact representations, such as memory tokens(Wu et al.) and compiled skill variants (Chen et al., 2026a).Overall, skill representation spans textual, structured-executable, and compressed or compiled forms, trading off interpretability, efficiency, portability, and execution control.

2.2.2 Skill Library

A skill library organizes multiple skills into a persistent repository for reuse across tasks and environments. Rather than loading all skills into the context window, libraries expose compact descriptors, embeddings, tags, taxonomies, or relation structures that allow agents to retrieve and activate only relevant skills. As summarized in Appendix Table 5, existing libraries differ mainly in how they structure the search space and expose relations among skills.

Flat repositories treat skills as independent retrievable entries, making them suitable for direct semantic matching or reranking (Li et al., 2025; Gao et al., 2026). **Hierarchical** libraries group skills by generality, domain, task type, or granularity, supporting coarse-to-fine retrieval and separating broadly useful strategies from task-specific procedures (Xia et al., 2026; Lu et al., 2026). **Graph-structured** libraries explicitly model inter-skill relations such as similarity, dependency, prerequisite, and composition, enabling structure-aware retrieval and multi-skill coordination (Liang et al., 2026; Liu et al., 2026a). **Shared** repositories further treat skills as ecosystem-level assets that can be reused, evaluated, synchronized, or governed across users, agents, and communities (Ma et al., 2026; Liu et al., 2026b). In this sense, a skill library functions as an external procedural memory for the agent: it stores reusable know-how, supports routing and composition, and, when coupled with feedback or governance mechanisms, provides the substrate for continual skill refinement.

2.3 Skill Acquisition

Skill acquisition addresses how reusable skill artifacts are produced and made available for later agent use. As summarized in Table 6, existing methods can be organized by skill provenance into two broad families: human-authored skills and experience-derived skills. The former rely on explicit expert design, whereas the latter derive procedural knowledge from agent interaction, optimization, memory, or execution traces.

2.3.1 Human-authored Skills

Human-authored skills are manually created by users, developers, or domain experts. They typically encode metadata, instructions, applicability conditions, and optional resources such as scripts, templates, or reference documents. Because they are intentionally designed, such skills are interpretable, auditable, and well suited to professional workflows where correctness, style, or compliance matters (Li et al., 2025; Xu and Yan, 2026; Li et al., 2026). Structured skill bases further organize human-designed skills into parameterized units or repositories for scalable reuse, as in desktop and computer-use settings(Chen et al., 2026b; Liang et al., 2026). However, manual authoring requires domain expertise and substantial maintenance effort, motivating automatic acquisition from agent experience.

2.3.2 Experience-derived Skills

Experience-derived skills are induced from agent behavior rather than specified entirely in advance. They convert interaction histories, optimization signals, demonstrations, failures, or execution traces into reusable procedural knowledge (Ni et al., 2026; Xia et al., 2026; Ouyang et al., 2026). We distinguish three representative acquisition patterns according to the primary signal used for skill construction.

Policy-driven. Policy-driven methods use rollouts, rewards, or policy optimization signals to distill effective behaviors into reusable skill libraries (Lu et al., 2026; Wang et al., 2026b; Xia et al., 2026). The resulting skills are often organized by generality, task type, or utility to guide future decision making and policy learning (Tu et al., 2026; Shi et al., 2026).

Memory-centric. Memory-centric methods view skill acquisition as procedural memory construction. They transform long-term interactions, user-specific contexts, or memory operations into

skill artifacts that can be selected, updated, and reused over time (Yang et al., 2026; Zhang et al., 2026b; Ouyang et al., 2026; Ma et al., 2026).

Trace-grounded. Trace-grounded methods acquire skills from concrete execution traces. They extract lessons or failure and consolidate them into transferable skill documents or validated skill folders (Ni et al., 2026; Alzubi et al., 2026; Zhang et al., 2026a). Unlike memory-centric methods that emphasize evolving experience stores, trace-grounded methods emphasize explicit execution evidence and its conversion into generalizable procedural guidance.

2.4 Skill Routing

Skill routing determines how an agent retrieves, selects, and activates relevant skills for a given task or execution context. As skill libraries grow, loading all skills becomes inefficient and may introduce irrelevant or conflicting guidance. Routing therefore serves as the interface between a large procedural repository and the agent's limited context window, selecting a compact set of skills for execution, ordering, or composition. Appendix Table 7 summarizes representative routing mechanisms.

Similarity-based. Similarity-based routing matches task descriptions with skill metadata, descriptions, or embeddings, typically through Top-K semantic retrieval(Xia et al., 2026; Li et al., 2025). This strategy provides scalable access to large repositories, but may miss dependencies, prerequisites, or context-specific applicability.

Context-aware. Context-aware routing selects skills based on the agent's current execution context rather than only the initial task description. In interactive environments, skill selection may depend on observations, retrieved memories, application state, or intermediate execution results (Chen et al., 2026b; Zhang et al., 2026b). This dynamic routing is particularly important when relevant skills change during execution.

Adaptive routing. Adaptive routing treats skill selection as a learned or utility-aware decision process. Rather than relying only on semantic similarity, these methods use utility estimates, retrieval statistics, or learned policies to select skills that are effective in similar contexts (Tu et al., 2026; Wang et al., 2026b; Shi et al., 2026). However, noisy or redundant skills may still degrade performance, motivating efforts to reduce dependence on external retrieval through skill internalization (Lu et al., 2026).

3 Skill Scaling

As skills evolve from isolated prompts into reusable procedural artifacts, scaling becomes a central challenge for agent skill systems. Skill repositories must continuously evolve, organize increasingly diverse procedures, support sharing across agents, and manage the growing cost of external skill usage. We therefore discuss skill scaling through four interconnected dimensions: temporal scaling, structural scaling, collective scaling, and internalization.

3.1 Temporal Scaling

Temporal scaling studies how skill systems improve over extended horizons through continuous accumulation, abstraction, refinement, and maintenance. Unlike static skill repositories that remain fixed, temporally scalable systems treat skills as evolving procedural knowledge that can be created from new experiences, updated after failures, and reorganized as the agent encounters more diverse tasks. The focus shifts from whether an agent can use skills to how its skill evolves over time.

Early work mainly studies continual skill accumulation from interaction experience. SkillRL (Xia et al., 2026) and AutoSkill (Yang et al., 2026) continually distill interaction trajectories into reusable procedural knowledge, allowing agents to expand their capabilities through long-term experience.

As repositories grow, temporal scaling increasingly requires skills to become more transferable and adaptable rather than simply more numerous. Trace2Skill (Ni et al., 2026) and ASDA (Yim et al., 2026) therefore extract generalized procedural patterns from successful and failed executions instead of storing isolated trajectories. EvoSkill (Alzubi et al., 2026), MemSkill (Zhang et al., 2026b), and Skills-Coach (Tian et al., 2026) further treat skills as mutable artifacts that can be repeatedly revised, validated, and improved through execution feedback and failure-driven updates.

As evolving repositories become larger and more dynamic, recent work further explores lifecycle management for long-term skill ecosystems. SkillOS (Ouyang et al., 2026), SLIM (Shen et al., 2026), and SkillClaw (Ma et al., 2026) study mechanisms for skill curation, retirement, synchronization, and repository maintenance, emphasizing that temporal scaling depends not only on continual skill growth, but also on maintaining the stability and utility of evolving procedural knowledge.

3.2 Structural Scaling

Structural scaling concerns how increasingly large and heterogeneous skill collections are organized into usable capability systems. As skill repositories grow, agents must distinguish related skills, identify dependencies, retrieve compatible procedures, and compose multiple skills without introducing execution conflicts. Structural scaling shifts the focus from merely storing skills to modeling the relations and compositional structures among skills.

One line of work studies how to organize large-scale skill repositories. SkillNet (Liang et al., 2026) constructs structured skill ecosystems using taxonomies, relation graphs, and modular skill packages.Related work further explores hierarchical or graph-based organizations that explicitly model relations such as dependency, similarity, and composition (Wang et al., 2026a). These approaches suggest that structural organization itself becomes increasingly important as skill repositories scale.

Another direction focuses on dependency-aware retrieval and compositional execution. Graph of Skills (Liu et al., 2026a) models skills as interconnected nodes and retrieves prerequisite or structurally compatible skills instead of relying only on semantic similarity. Agentic Proposing (Jiao et al., 2026) further treats skills as composable building blocks that can be combined into more complex reasoning procedures. In interactive environments, systems such as CUA-Skill (Chen et al., 2026b) organize reusable procedures into parameterized execution structures that support compositional action execution across different abstraction levels.

Overall, structural scaling shows that the effectiveness of large-scale skill systems depends not only on the quality of individual skills, but also on how skills are organized, retrieved, and connected.

3.3 Collective Scaling

Collective scaling extends skill learning from individual agents to shared ecosystems. Rather than relying on isolated trajectories, agents can accumulate and refine skills across overlapping tasks and recurring patterns, transforming distributed experience into shared procedural knowledge.

A central form of collective scaling is cross-user skill evolution. SkillClaw (Ma et al., 2026) treats skills as shared assets across multiple agents, enabling knowledge to be accumulated from distributed interaction experience and shifting learning from individual trajectories to a shared evidence base.

Collective scaling can also emerge through co-evolutionary skill construction. Co-EvoSkills (Zhang et al., 2026a) couples a generator and a verifier, allowing skills to be iteratively proposed and evaluated through interaction between complementary agent roles, rather than relying on a single learning process.

Another line of work focuses on shared skill repositories. Systems such as SkillX (Wang et al., 2026a) and SkillNet (Liang et al., 2026) organize reusable skills into structured knowledge bases, enabling skills learned in one context to be reused across tasks and agents.

Overall, collective scaling reframes skills as shared infrastructure rather than private agent memory. Its central challenge lies in how distributed experiences are aggregated, how updates are validated, and how improved skills are propagated without introducing noise or context-specific bias. This perspective complements temporal and structural scaling by explaining how skill improvements accumulate across agents, users, and environments.

3.4 Internalization

As skill repositories scale, keeping all procedural knowledge as external context becomes increasingly costly. Although external skills are modular, editable, and inspectable, they also introduce retrieval overhead, context consumption, routing uncertainty, and interference from irrelevant or redundant skills. Skill internalization addresses this issue by compressing, compiling, or distilling reusable procedures into more compact behavioral mechanisms, asking whether skills should remain as external files, retrieved prompts, compact memory tokens, executable artifacts, or policy-level behaviors.

Existing work explores internalization at different degrees. Some methods retain an external skill interface while reducing its cost. For example, SkillReducer (Gao et al., 2026) removes redundant content from skill descriptions, TokMem (Wu et al.) compresses reusable procedures into token-level memory, and SkVM (Chen et al., 2026a) treats skills as program-like artifacts that can be compiled or adapted for execution. Other methods use skills as temporary supervision during training. Skill-SD (Wang et al., 2026b) distills trajectory-derived skills into student behavior through teacher-guided self-distillation, while Skill0 (Lu et al., 2026) gradually removes skill context to enable zero-skill in-

ference. From a lifecycle perspective, SLIM (Shen et al., 2026) further suggests that skills should be retained, retired, expanded, or internalized according to their marginal utility. Overall, internalization highlights a key trade-off: external skills improve transparency and controllability, whereas internalized skills reduce inference-time overhead and routing burden.

4 Evaluation

We evaluate skill scaling along three dimensions: temporal, structural, and collective scaling, each corresponding to a different source of scaling pressure. Our experiments focus on how scaling the quantity, diversity, and sharing scope of skills affects overall agent behavior.

4.1 Experimental Setup

Benchmarks. We conduct experiments on three interactive agent benchmarks: WebShop (Yao et al., 2022a), ScienceWorld (Wang et al., 2022), and ALFWorld (Shridhar et al., 2020).

WebShop is a web shopping environment for instruction-following product search and purchase tasks. ScienceWorld contains diverse scientific reasoning tasks across multiple domains. ALFWorld is a household embodied environment for long-horizon sequential decision-making.

Metrics. We report three evaluation metrics: (1) success rate (SR), which measures the proportion of successfully completed tasks; (2) reward, which reflects task completion quality; and (3) average number of steps, which measures efficiency.

4.2 Temporal Scaling Evaluation

Temporal scaling studies how skill systems evolve as agents continuously accumulate new procedural knowledge from interaction experience. We investigate whether expanding skill repositories consistently improve agent performance over time.

Protocol. We evaluate temporal scaling on WebShop by studying how agent performance changes as skills are iteratively evolved over multiple rounds. In each round, the agent solves a batch of WebShop tasks, records its interaction trajectories, and distills reusable skills from them. The extracted skills are added to the skillbank and used in subsequent rounds, forming a continual skill evolution process.

Setting	Cap.	Round	Reward	SR	Steps
		No Skill			
–	–	–	0.64	0.12	18.3
		Static Skillbank			
Static	5	–	0.68	0.30	12.30
Static	10	–	0.78	0.44	7.44
Static	20	–	0.72	0.40	8.80
		Evolving Skillbank			
Evolving	5	1	0.65	0.40	11.6
		2	0.84	0.53	5.9
		3	0.79	0.53	10.4
		4	0.59	0.31	12.0
		5	0.72	0.33	10.1
Evolving	10	1	0.87	0.60	5.6
		2	0.80	0.50	8.5
		3	0.62	0.22	7.1
		4	0.51	0.32	14.2
		5	0.72	0.39	8.5
Evolving	20	1	0.81	0.63	10.1
		2	0.63	0.30	12.9
		3	0.71	0.55	11.0
		4	0.54	0.21	17.4
		5	0.50	0.20	15.1

Table 1: Temporal scaling results.

Scaling Variables. We vary two key factors: the number of evolution rounds and the maximum capacity of the skillbank.

The number of evolution rounds evaluates how performance changes as the skill repository continuously expands over time, while the capacity limit tests whether larger skillbanks consistently improve performance or introduce redundancy and retrieval noise.

Compared Settings. We compare three skill settings: (1) No Skill, where the agent operates without any external skill repository; (2) Static Skillbank, where the agent uses a fixed repository with a capacity of 5, 10, or 20 skills; and (3) Evolving Skillbank, where the repository is continuously updated through iterative skill evolution under the same capacity limits.

Results. Table 1 presents the temporal scaling results on WebShop under different skill capacities and evolution rounds.

Compared with the no-skill baseline, both static and evolving skillbanks substantially improve task success and interaction efficiency. Static skillbanks consistently outperform the vanilla setting, showing that reusable procedural knowledge can effectively reduce redundant planning in long-horizon tasks.

For evolving skillbanks, performance initially improves during early evolution rounds. For example, under a capacity of 10 skills, the first evolution round achieves the best overall performance, reaching a reward of 0.87 and a success rate of 0.60.

However, performance does not improve monotonically as evolution continues. Across most settings, reward and SR begin to fluctuate or decline after additional evolution rounds, while interaction steps increase again. This behavior becomes more pronounced under larger skill capacities.

Analysis. These results suggest that temporal scaling does not lead to monotonic performance improvement. While evolving additional skills can initially enhance task performance, continuously expanding skill repositories may gradually introduce redundancy and unstable retrieval behavior, reducing the effectiveness of further skill accumulation.

4.3 Structural Scaling Evaluation

Structural scaling studies how agents use skills as skill repositories include skills from more task categories. We investigate whether increasing the diversity of the skill repository improves task coverage and skill reuse.

Protocol. We evaluate structural scaling on ScienceWorld using five task categories with distinct procedural patterns. For each category, we first evolve a task-specific skillbank from interaction trajectories. We then progressively combine skills from multiple categories to construct mixed skill repositories of increasing diversity.

The agent is evaluated using repositories containing skills from one to five task categories. As more categories are included, the repository contains a broader range of skills, including procedures that may be irrelevant to the current task.

Scaling Variable. The main scaling variable is the number of task categories included in the skill repository. Unlike temporal scaling, the skill repositories remain fixed after evolution. We gradually combine skills from more task categories to study how increasing repository diversity affects skill retrieval and task performance.

Compared Settings. We compare three settings: (1) No Skill, where the agent solves ScienceWorld tasks without any external skill repository; (2) Single-task Skillbank, where the agent uses skills evolved only from the same task category as the evaluation task; and (3) Mixed Skillbank, where the repository combines skills from multiple task categories.

By increasing the number of categories in the mixed repository, we evaluate whether broader procedural diversity improves generalization or instead introduces retrieval interference across heterogeneous skills.

Results. Table 2 evaluates structural scaling on ScienceWorld by progressively increasing the diversity of task categories represented within a shared skill repository.

Compared with the no-skill baseline, task-specific skillbanks generally improve performance across most tasks. Moderate cross-task skill sharing can further improve performance in some settings. For example, the Matter task achieves its best result under the 3-task mixed skillbank setting, outperforming both the no-skill baseline and the task-specific skillbank.

However, increasing structural diversity does not consistently improve performance. As more heterogeneous task categories are merged into the same repository, performance becomes increasingly unstable across tasks. For example, Electricity performance drops substantially under the 4-task and 5-task settings, while Biology-plant exhibits large fluctuations across different repository scales.

Analysis. These results suggest that structural scaling introduces a trade-off between capability diversity and execution stability. While structurally diverse repositories can provide transferable procedural knowledge across related tasks, increasingly heterogeneous skill compositions do not consistently produce better agent behavior.

4.4 Collective Scaling Evaluation

Collective scaling studies how skills evolve when multiple agents jointly contribute experience to a shared skill repository.

Protocol. We evaluate collective scaling on ALFWorld using multiple task-specific agents. ALFWorld contains many related household tasks with overlapping procedures, such as object finding, placement, heating, cooling, and cleaning, making cross-task skill sharing meaningful.

Each agent first evolves its own skillbank from trajectories. We then construct shared skill repositories by allowing multiple agents to jointly contribute skills to the same repository.

Test Task	Individual Skillbanks				Mixed Skillbank by Number of Task Categories							
	No Skill		Own Skill		2 Tasks		3 Tasks		4 Tasks		5 Tasks	
	R	Steps	R	Steps	R	Steps	R	Steps	R	Steps	R	Steps
Matter	66.8	19.1	70.7	19.3	68.6	22.5	80.0	18.1	68.8	22.2	61.5	21.6
Electricity	69.9	25.4	77.2	25.6	70.1	19.5	74.6	20.3	57.0	22.1	61.8	22.2
Biology-plant	45.3	23.5	52.3	25.8	–	–	45.6	28.7	38.7	22.2	61.2	18.0
Chemistry	69.7	18.29	80.6	16.95	–	–	–	–	60.75	16.77	62.75	17.00
Biology-genetics	58.4	10.20	58.6	10.65	–	–	–	–	–	–	63.0	13.38

Table 2: Structural scaling results.

Test Task	Individual Skillbanks				Shared Skillbank by Number of Agents							
	No Skill		1 Agent		2 Agents		3 Agents		4 Agents		5 Agents	
	R	Steps	R	Steps	R	Steps	R	Steps	R	Steps	R	Steps
Clean & Place	70.4	19.30	88.9	10.22	85.2	11.67	96.3	9.11	88.9	10.67	92.6	11.11
Pick Two & Place	58.3	22.21	95.8	14.25	75.0	19.42	87.5	16.29	66.7	20.79	91.7	14.46
Heat & Place	25.0	28.00	87.5	12.00	–	–	81.3	12.19	81.3	12.81	87.5	11.31
Cool & Place	56.0	20.12	68.0	16.60	–	–	–	–	84.0	13.28	76.0	15.88
Examine in Light	76.9	15.85	92.3	12.69	–	–	–	–	–	–	96.4	11.15

Table 3: Collective scaling results.

We progressively increase the number of participating agents from two to five. After shared evolution, each agent is evaluated on its own task using the shared skill repository.

Scaling Variable. The main scaling variable is the number of agents.As more agents participate, the repository accumulates experience from a broader range of related tasks. This may improve skill transfer across tasks, but it may also introduce noisy or task-specific skills that do not generalize well to other agents.

Compared Settings. We compare three settings: (1) No Skill, where the agent solves tasks without any skill; (2) Individual Skillbank, where each agent evolves and uses its own bank; and (3) Shared Skillbank, where multiple agents jointly evolve and use a shared repository.

Results. Table 3 evaluates collective scaling in ALFWorld by allowing multiple task-specific agents to jointly maintain a shared skill repository.

Compared with isolated task-specific skillbanks, shared repositories often improve both task success and execution efficiency. For example, Clean & Place achieves its best performance under the 3-agent shared setting, while Examine in Light reaches its highest reward under the 5-agent setting. These results show that procedural knowledge discovered by one agent can transfer effectively to related tasks performed by other agents.

However, increasing the number of participating agents does not consistently improve performance across all tasks. Several tasks exhibit noticeable fluctuations under larger shared repositories. For example, Pick Two & Place shows unstable performance across different sharing scales, while Cool & Place only benefits under larger shared settings.

Analysis. These results suggest that collective scaling introduces a trade-off between experience sharing and behavioral consistency. While shared repositories can improve capability transfer across related tasks, larger shared skill ecosystems do not consistently produce more stable behavior.

5 Conclusion

In this paper, we presented a survey and empirical study of skill scaling in LLM agents. We organized existing research into three complementary dimensions: temporal scaling, structural scaling, and collective scaling, providing a unified perspective on how skill ecosystems evolve, compose, and propagate across increasingly autonomous agent systems. Through empirical studies, we showed that skill scaling is not purely beneficial: while larger and more diverse skill ecosystems can improve agent capability and efficiency, they may also introduce redundancy, interference, retrieval ambiguity, and coordination instability.

Limitations

Our study has several limitations. First, although we analyze skill scaling across temporal, structural, and collective dimensions, the proposed taxonomy may not fully cover all possible forms of skill evolution in future agent systems. Second, our empirical evaluations are conducted on a limited set of representative environments and skill frameworks, which may not capture the full diversity of real-world agent deployments. Finally, our analysis primarily focuses on externalized skill systems, while the interaction between skill scaling and emerging forms of internalized or latent procedural knowledge remains underexplored.

References

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

Le Chen, Erhu Feng, Yubin Xia, and Haibo Chen. 2026a. Skvm: Compiling skills for efficient execution everywhere. *arXiv e-prints*, pages arXiv–2604.

Tianyi Chen, Yinheng Li, Michael Solodko, Sen Wang, Nan Jiang, Tingyuan Cui, Junheng Hao, Jongwoo Ko, Sara Abdali, Leon Xu, and 1 others. 2026b. Cuaskill: Develop skills for computer using agent. *arXiv preprint arXiv:2601.21123*.

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

Yudong Gao, Zongjie Li, Zimo Ji, Pingchuan Ma, Shuai Wang, and 1 others. 2026. Skillreducer: Optimizing llm agent skills for token efficiency. *arXiv preprint arXiv:2603.29919*.

Sirui Hong, Yizhang Lin, Bang Liu, Bangbang Liu, Binhao Wu, Ceyao Zhang, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, and 1 others. 2025. Data interpreter: An llm agent for data science. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19796–19821.

Shengran Hu, Cong Lu, and Jeff Clune. 2025. Automated design of agentic systems. In *International Conference on Learning Representations*, volume 2025, pages 21344–21377.

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. *arXiv preprint arXiv:2602.03279*.

Fangzhou Li, Pagkratios Tagkopoulos, and Ilias Tagkopoulos. 2025. Skillflow: Scalable and efficient agent skill retrieval system. In *First Workshop on Agent Skills*.

Xiangyi Li, Wenbo Chen, Yimin Liu, Shenghan Zheng, Xiaokun Chen, Yifeng He, Yubo Li, Bingran You, Haotian Shen, Jiankai Sun, and 1 others. 2026. Skillsbench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*.

Yuan Liang, Ruobin Zhong, Haoming Xu, Chen Jiang, Yi Zhong, Runnan Fang, Jia-Chen Gu, Shumin Deng, Yunzhi Yao, Mengru Wang, and 1 others. 2026. Skillnet: Create, evaluate, and connect ai skills. *arXiv preprint arXiv:2603.04448*.

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

Hongyi Liu, Haoyan Yang, Tao Jiang, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2026b. Skillsvote: Lifecycle governance of agent skills from collection, recommendation to evolution. *arXiv preprint arXiv:2605.18401*.

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

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

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*.

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. 2023. Memgpt: towards llms as operating systems.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2024. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. 2026. Toolrl: Reward is all tool learning needs. *Advances in Neural Information Processing Systems*, 38:105523–105553.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, volume 2024, pages 9695–9717.

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

Jie-Jing Shao, Haiyan Yin, Yueming Lyu, Xingrui Yu, Lan-Zhe Guo, Ivor Tsang, James Kwok, and Yu-Feng Li. 2026. Lifting traces to logic: Programmatic skill induction with neuro-symbolic learning for long-horizon agentic tasks. *arXiv preprint arXiv:2605.01293*.

Junhao Shen, Teng Zhang, Xiaoyan Zhao, and Hong Cheng. 2026. Dynamic skill lifecycle management for agentic reinforcement learning. *arXiv preprint arXiv:2605.10923*.

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

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

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

Shuzheng Si, Haozhe Zhao, Yu Lei, Qingyi Wang, Dingwei Chen, Zhitong Wang, Zhenhailong Wang, Kangyang Luo, Zheng Wang, Gang Chen, and 1 others. 2026. From context to skills: Can language models learn from context skillfully? *arXiv preprint arXiv:2604.27660*.

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*.

Yu Tian, Jiawei Chen, Lifan Zheng, Mingxiang Tao, Xinyi Zeng, Zhaoxia Yin, Hang Su, and Xian Sun. 2026. Skills-coach: A self-evolving skill optimizer via training-free grpo. *arXiv preprint arXiv:2604.27488*.

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

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*.

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, Xiaoxin Chen, and 1 others. 2026b. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents. *arXiv preprint arXiv:2604.10674*.

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.

Zijun Wu, Yongchang Hao, and Lili Mou. Tokmem: One-token procedural memory for large language models. In *The Fourteenth International Conference on Learning Representations*.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, and 1 others. 2026. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*.

Renjun Xu and Yang Yan. 2026. Agent skills for large language models: Architecture, acquisition, security, and the path forward. *arXiv preprint arXiv:2602.12430*.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, and 1 others. 2025. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*.

Yutao Yang, Junsong Li, Qianjun Pan, Bihao Zhan, Yuxuan Cai, Lin Du, Jie Zhou, Kai Chen, Qin Chen, Xin Li, and 1 others. 2026. Autoskill: Experience-driven lifelong learning via skill self-evolution. *arXiv preprint arXiv:2603.01145*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022a. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022b. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Tik Yu Yim, Wenting Tan, Sum Yee Chan, Tak-Wah Lam, and Siu Ming Yiu. 2026. Asda: Automated skill distillation and adaptation for financial reasoning. *arXiv preprint arXiv:2603.16112*.

Hanrong Zhang, Shicheng Fan, Henry Peng Zou, Yankai Chen, Zhenting Wang, Jiayu Zhou, Chengze Li, Wei-Chieh Huang, Yifei Yao, Kening Zheng, and 1 others. 2026a. Coevoskills: Self-evolving agent skills via co-evolutionary verification. *arXiv preprint arXiv:2604.01687*.

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

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, and 1 others. 2025. Aflow: Automating agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pages 34040–34077.

Mingda Zhang, Tiesunlong Shen, Haoran Luo, Wenjin Liu, Zikai Xiao, Erik Cambria, and Xiaoying Tang. 2026c. Skillflow: Flow-driven recursive skill evolution for agentic orchestration. *arXiv preprint arXiv:2605.14089*.

Guangxiang Zhao, Qilong Shi, Xusen Xiao, Xiangzheng Zhang, Tong Yang, and Lin Sun. 2026. Thinking with reasoning skills: Fewer tokens, more accuracy. *arXiv preprint arXiv:2604.21764*.

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

6 Appendix

Concept	Primary Function	Difference from Skills	Representative References
Tools	Expose atomic executable functions with defined inputs and outputs	Skills specify when, how, and in what sequence tools should be used	ToolFormer(Schick et al., 2023),ToolLLM(Qin et al., 2024),Gorilla(Patil et al., 2024),ToolRL(Qian et al., 2026)
Workflows	Encode predefined execution structures for recurring processes	Skills package procedural know-how that may instantiate, adapt, or compose workflows	ReAct(Yao et al., 2022b),Reflexion(Shinn et al., 2023),VOYAGER(Wang et al., 2023),AFLOW(Zhang et al., 2025),Data Interpreter(Hong et al., 2025)
Memory	Store past observations, interactions, or experiences	Skills abstract experience into reusable operational procedures	MemGPT(Packer et al., 2023),Memory-R1(Yan et al., 2025),Mem0(Chhikara et al., 2025)
Skills	Externalize reusable procedural knowledge for task execution	Serve as the focal abstraction in this survey	SkVM(Chen et al., 2026a),SkillRL(Xia et al., 2026),AutoSkill(Yang et al., 2026)

Table 4: Conceptual boundaries between skills and adjacent abstractions.

Library Structure	Organization Principle	Representative Works
Flat	Stores skills as independent entries, typically indexed by metadata, descriptions, tags, or embeddings for direct retrieval.	SkillFlow(Li et al., 2025),Reasoning-Skill(Zhao et al., 2026),SkillReducer(Gao et al., 2026)
Hierarchical	Organizes skills by generality, domain, task type, or granularity, enabling coarse-to-fine selection and reuse.	ASDA(Yim et al., 2026),SkilLRL(Xia et al., 2026),SKILL0(Lu et al., 2026)
Graph	Models relations among skills, such as similarity, dependency, prerequisite, and composition, to support structure-aware retrieval and multi-skill coordination.	SkillNet(Liang et al., 2026),Graph of Skills(Liu et al., 2026a),
Shared	Skills are shared across users, communities, marketplaces, repositories, or synchronized/evaluated as common skill ecosystems.	SkillClaw(Ma et al., 2026),SkillsVote(Liu et al., 2026b)

Table 5: Representative organization patterns of skill libraries.

Category	Acquisition Type	Source Signal	Acquisition Mechanism	Representative Works
Human-authored	Manual design	Expert knowledge, domain workflows, community-contributed procedures	Manually encodes procedural instructions, applicability conditions, constraints, and supporting resources into reusable skill artifacts.	SkillFlow(Li et al., 2025),SkillsBench(Li et al., 2026),CUA-Skill(Chen et al., 2026b),SkillNet(Liang et al., 2026)
Experience-derived	Policy-driven	Rollouts, rewards, policy optimization signals, utility estimates	Distills effective behaviors from agent-environment interaction and organizes them into skill libraries for future decision making and policy learning.	SkillRL(Xia et al., 2026),Skill0(Lu et al., 2026),Skill1(Shi et al., 2026),Skill-SD(Wang et al., 2026b),SkillOS(Ouyang et al., 2026),SkillFlow(Zhang et al., 2026c)
	Memory-centric	Long-term interactions, user-specific contexts, memory operations	Converts accumulated experience into procedural memory artifacts that can be selected, updated, and reused over time.	AutoSkill(Yang et al., 2026),Mem-Skill(Zhang et al., 2026b),SkillClaw(Ma et al., 2026)
	Trace-grounded	Successful trajectories, failed executions, execution traces, feedback	Extracts local lessons, failure causes, or reusable patches from concrete traces and consolidates them into transferable skill documents or validated skill folders.	Trace2Skill(Ni et al., 2026),Ctx2Skill(Si et al., 2026),EvoSkill(Alzubi et al., 2026),Co-EvoSkills(Zhang et al., 2026a)

Table 6: Taxonomy of skill acquisition methods.

Routing Type	Routing Signal	Routing Mechanism	Representative Works
Similarity-based	Task descriptions, skill metadata, natural-language descriptions, embeddings	Retrieves candidate skills by matching the current task with skill descriptors, often using Top-K semantic or lexical retrieval over a skill bank.	SkillFLow(Li et al., 2025),SkillRL(Xia et al., 2026),SRA(Su et al., 2026),AutoSkill(Yang et al., 2026),SkillReducer(Gao et al., 2026)
Context-aware	Current observations, execution state, retrieved memories, arguments, intermediate results	Selects or activates skills according to the agent's evolving execution context rather than only the initial task description.	CUA-Skill(Chen et al., 2026b),MemSkill(Zhang et al., 2026b),SkillsVote(Liu et al., 2026b), NSI(Shao et al., 2026)
Structure-aware	Skill taxonomies, dependency graphs, prerequisite relations, composition relations	Uses explicit relations among skills to recover dependencies, compose multi-skill bundles, or select execution-ready skill sets.	SkillNet(Liang et al., 2026),Graph of Skills(Liu et al., 2026a),ASDA(Yim et al., 2026)
Adaptive	Utility estimates, retrieval statistics, rewards, policy signals, historical success	Treats routing as a learned or utility-aware decision process, selecting skills based on both relevance and expected downstream usefulness.	D2Skill(Tu et al., 2026),Skill0(Lu et al., 2026),Skill1(Shi et al., 2026),SkillOS(Ouyang et al., 2026)

Table 7: Taxonomy of skill routing types