Beyond Scaling: A Survey of Data-Efficient Learning for LLM Agents

Yaqing Wang[1], **Zhenlin Luo**[1,2], **Peiyao Zhao**[1,3], **Yunfeng Cai**[1], **Quanming Yao**[4,5,6*]
[1]Beijing Institute of Mathematical Sciences and Applications
[2]Institute of Statistics and Big Data, Renmin University of China
[3]Yau Mathematical Sciences Center, Tsinghua University
[4]Department of Electronic Engineering, Tsinghua University
[5]Beijing National Research Center for Information Science and Technology
[6]State Key Laboratory of Space Network and Communications
{wangyaqing, luozhenlin, zhaopeiyao, caiyunfeng}@bimsa.cn, qyaoaa@tsinghua.edu.cn

Abstract

LLM-based agents perceive, reason, act, and adapt through interaction. While scaling remains important, agentic progress also depends on extracting more learning signal from limited experience, including demonstrations, feedback, reasoning traces, tool-use records, and interaction trajectories. This survey develops an agent-centric view of data-efficient learning. We formulate an agentic learning loop, define data efficiency as capability improvement without proportional increases in supervision or real-environment trial-and-error, and synthesize methods into experience augmentation, agent structural design, and learning paradigms. We also summarize representative application domains and open challenges in experience reuse and cost-aware evaluation. Overall, the agentic era requires smarter ways to acquire, structure, reuse, and learn from limited experience.

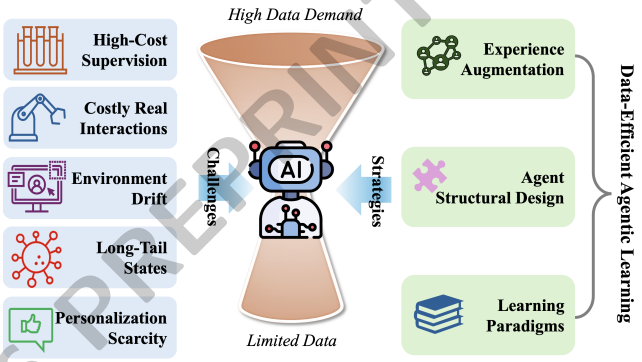


Figure 1: Conceptual overview of data-efficient agentic learning and its three complementary strategies.

1 Introduction

Large language model (LLM)-based agents are moving beyond prompt-only prototypes into closed-loop systems that perceive, reason, plan, and act in dynamic environments. Their success depends not only on one-shot generation quality, but also on whether they can acquire, verify, and refine behaviors under feedback and long-horizon goals. Such agents now appear in web and GUI automation, embodied decision making, and scientific or medical workflows, where perception, reasoning, and action execution must be coordinated end-to-end [Shridhar *et al.*, 2021; Zhou *et al.*, 2024c; Xie *et al.*, 2024; Laurent *et al.*, 2024].

Realistic deployment exposes a complementary bottleneck to scaling: agents must improve under limited and costly experience. Supervision such as grounding labels, demonstrations, expert feedback, or verification can be expensive; online trial-and-error consumes environment steps, tool calls, and human oversight, and may introduce safety or reliability risks. Interaction-derived experience may also become stale under environment or interface drift [Zhou *et al.*, 2024c; Xie *et al.*, 2024], while embodied, scientific, and medical workflows face privacy, expert-time, or experimental costs [Chen *et al.*, 2024a; Rein *et al.*, 2024; Laurent *et al.*, 2024]. Thus, building capable LLM agents in a data-efficient way becomes a first-order question.

Data efficiency has long been studied in few-shot learning [Wang *et al.*, 2020], robust learning [Han *et al.*, 2018], and sample-efficient reinforcement learning [Yu, 2018]. LLM agents, however, broaden both what counts as data and where efficiency comes from: beyond labels, they use demonstrations, trajectories, reasoning traces, tool-use records, feedback, and verification outcomes [Yao *et al.*, 2022; Shinn *et al.*, 2023]. We use *agentic data* to cover both external supervision and interaction-derived experience, and use *agentic experience* when emphasizing trajectories, feedback, memory, and reuse within the agent loop. Data efficiency therefore depends not only on a learning algorithm, but also on how limited experience is acquired, structured, reused, and transformed.

This survey provides a unified synthesis of data-efficient learning for LLM agents. We formulate an agentic learning loop, organize methods into experience augmentation, agent structural design, and learning paradigms, and summarize representative application domains and open challenges across web, GUI, embodied, medical, and scientific settings.

*Corresponding author.

Difference with existing surveys. Despite the rapidly growing literature on LLM-based agents, existing surveys mainly organize the field by architectures, components, applications, or general capabilities. Broad surveys cover agent construction, evaluation, and deployment [Wang *et al.*, 2024b; Liu *et al.*, 2025a]; recent work discusses the shift toward model-native agentic AI [Sang *et al.*, 2025]; and specialized surveys focus on multi-agent collaboration [Guo *et al.*, 2024; Wu *et al.*, 2025a], feedback [Liu *et al.*, 2025b], memory [Zhang *et al.*, 2025], and planning [Torreno *et al.*, 2017]. However, bounded supervision, costly interaction, and limited agentic experience have not been treated as a unifying lens. This survey fills this gap by synthesizing experience generation and reuse, structural agent design, and learning paradigms under the perspective of data-efficient learning. The LLM-agent setting makes this lens distinctive because the reusable units are not only labeled examples or transitions, but also reasoning traces, tool-use histories, verifier signals, contextual memories, and prompt-time demonstrations; the relevant budgets also include human/expert feedback, tool calls, real-environment executions, and inference-time orchestration.

2 Overview

Background. We study how to obtain and improve LLM-based agents in a data-efficient way. An LLM-based agent can be viewed as a closed-loop decision-making system that repeatedly perceives the environment, reasons and plans with an LLM, executes actions, and receives feedback. At time step t, the environment state is $s_t \in \mathcal{S}$ and the interaction loop in Figure 2 is

$$o_t = P(s_t), \qquad (g_t, a_t) = L_\theta(o_t, m_{t-1}),$$
$$m_t = M(m_{t-1}, g_t), \qquad a'_t = E(a_t),$$
$$s_{t+1} = T(s_t, a'_t).$$

Here, $\mathcal{S}$ denotes the environment state space, P denotes a perceiver that maps the environment state s_t to an observation o_t, L_θ is the LLM that produces intermediate reasoning outputs g_t and selects the next action a_t, M is a memory module that maintains and updates the agent state m_t, E is an action executor that converts the selected action into an executable form a'_t, and T is the environment transition function, which maps the current state and executable action to the next environment state. Each step may produce within-loop experience, such as observations, reasoning traces, actions, rewards, tool outputs, or verification signals; we denote this collection by $\mathcal{D}_{\text{loop}}$. The agent may also use external data or supervision $\mathcal{D}_{\text{ext}}$, such as demonstrations, labels, preferences, expert annotations, or verified outcomes. Their union is the available agentic data: $\mathcal{D} = \mathcal{D}_{\text{loop}} \cup \mathcal{D}_{\text{ext}}$.

Definition. We define *data-efficient agentic learning* as the study of how to obtain and improve LLM-based agents in interactive decision-making settings using limited agentic data $\mathcal{D}$. An approach is data-efficient when its gains do not rely on large amounts of new supervision or extensive real-environment trial-and-error, but instead improve the information content, reliability, and reuse value of limited within-loop experience and external supervision.

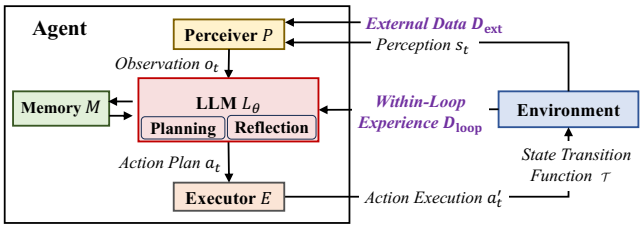


Figure 2: Agentic learning loop with core components and agentic data sources.

Framework. Data-efficiency bottlenecks arise from both costly supervision in $\mathcal{D}_{\text{ext}}$ and costly interaction needed to obtain $\mathcal{D}_{\text{loop}}$. We organize methods into three coupled design levers. *Experience augmentation* expands or improves the effective experience pool through generation, simulation, or transformation (Section 3.1). *Agent structural design* reorganizes modules such as P, M, L_θ, planners, verifiers, and E so that each observation and feedback signal is processed, stored, and reused more effectively (Section 3.2). *Learning paradigms* adapt or condition the LLM or policy from bounded $\mathcal{D}$, ranging from inference-time reuse to lightweight parameter updates and budgeted reinforcement learning (RL) (Section 3.3). These levers interact: for example, environment drift may require refreshed trajectories, memory verification to avoid stale reuse, and test-time or budget-efficient adaptation under bounded interaction.

3 Taxonomy

The following three subsections review three complementary aspects of data-efficient agentic learning introduced in Section 2. For each aspect, we summarize its core idea, the type of data scarcity it addresses (samples, labels, or interactions), and representative works, noting that practical systems often combine multiple aspects.

3.1 Experience Augmentation

In data-efficient agentic learning, performance is often constrained by the availability, cost, and quality of agentic data rather than model capacity alone. Real-world trajectories rarely cover long-tail states, human supervision is costly, and online interaction incurs substantial time, safety, and resource overhead. Under such constraints, naive trial-and-error yields low information gain and poor generalization.

Experience augmentation addresses this bottleneck by expanding and strengthening the effective experience pool under bounded budgets. Rather than simply collecting more data, its goal is to reduce the cost of acquiring experience while increasing the informativeness of the collected or synthesized experience, and reusability of existing experience. We organize existing approaches into two categories (Table 1): *experience generation* and *experience transformation*.

Experience Generation. This category expands the effective experience pool beyond what is directly collected from the real environment (Figure 4(a)), including two strategies: *non-interactive generation and interactive simulation.*

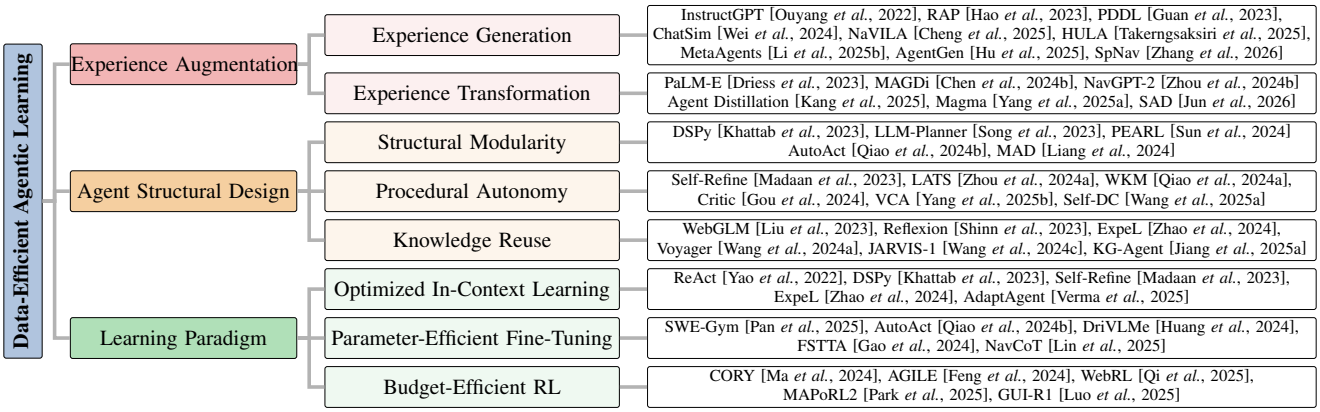


Figure 3: Taxonomy of data-efficient agentic learning.

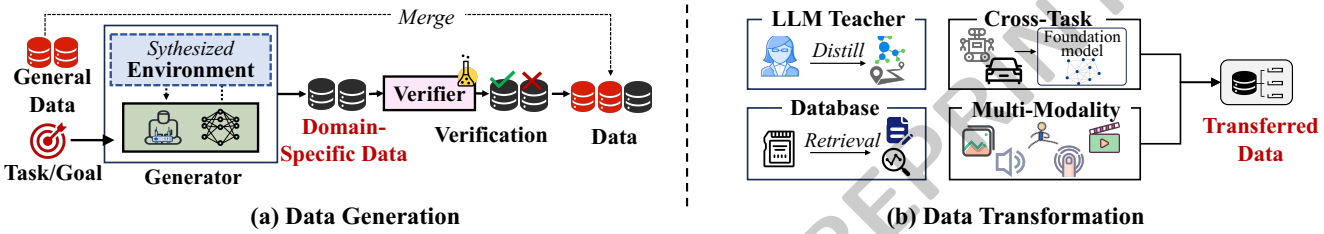


Figure 4: Illustration of experience augmentation for data-efficient agentic learning. (a) Experience generation creates additional training experience either through direct synthesis or lower-cost simulated/surrogate interaction. (b) Experience transformation enriches and restructures limited existing experience into more reusable training signals.

Non-Interactive Generation. This category produces trajectory data without interacting with any environment. In LLM-based agents, experience generation often starts from *human-centric* designs, where a small amount of carefully curated demonstrations or preference feedback is used to concentrate supervision on effective behaviors and reduce exploration waste, as exemplified by InstructGPT [Ouyang *et al.*, 2022] and HULA [Takerngsaksiri *et al.*, 2025]. It can further scale through *model-centric* generation, where LLMs/VLMs synthesize trajectories from scratch or convert visual observations into step-wise supervision at low marginal cost, relying on verification to improve data quality; representative examples include action sequence construction in NaVILA [Cheng *et al.*, 2025] and spatial navigation instructions in SpNav [Zhang *et al.*, 2026]. These methods improve data efficiency by creating additional training signals to expand coverage under limited supervision.

Interactive Simulation. This category reduces reliance on expensive real-world interaction by shifting exploration and failure discovery to cheaper surrogate environments (Figure 4). Instead of real trial-and-error, agents explore within explicit simulators or learned world models to obtain diverse trajectories and feedback at lower cost. The objective is not perfect realism, but sufficient diversity and structural fidelity to complement scarce real experience. World-model-based approaches such as RAP [Hao *et al.*, 2023] and symbolic planning frameworks like PDDL [Guan *et al.*, 2023] enable agents to simulate outcomes and validate plans without repeated external execution. Systems such as ChatSim [Wei *et al.*, 2024], MetaAgents [Li *et al.*, 2025b] and AgentGen [Hu *et al.*, 2025] demonstrate the utility of controllable simulated environments for generating targeted and rare interaction scenarios. In sum, they improve data efficiency by substituting costly real-world interaction with lower-cost interaction sources.

Category	Core Idea	Data Type	Representative Works
Experience Generation	Generate new experience via direct synthesis or cheaper surrogate interaction	S, L, I	InstructGPT [Ouyang *et al.*, 2022]; RAP [Hao *et al.*, 2023]; PDDL [Guan *et al.*, 2023]; ChatSim [Wei *et al.*, 2024]. NaVILA [Cheng *et al.*, 2025]; HULA [Takerngsaksiri *et al.*, 2025]; MetaAgents [Li *et al.*, 2025b]; AgentGen [Hu *et al.*, 2025]; SpNav [Zhang *et al.*, 2026]
Experience Transformation	Increase information density and reuse value of existing experience	S, L	ADAPT [Lin *et al.*, 2022]; PaLM-E [Driess *et al.*, 2023]; MAGDi [Chen *et al.*, 2024b]; NavGPT-2 [Zhou *et al.*, 2024b]; Agent Distillation [Kang *et al.*, 2025]; K-RagRec [Wang *et al.*, 2025b]; Magma [Yang *et al.*, 2025a]; SAD [Jun *et al.*, 2026]

Table 1: Experience augmentation strategies for data-efficient agentic learning. S/L/I denote sample/label/interaction.

Experience Transformation. This category improves data efficiency by enriching and restructuring limited real trajectories with complementary information, allowing each experience item to carry stronger and more reusable learn-

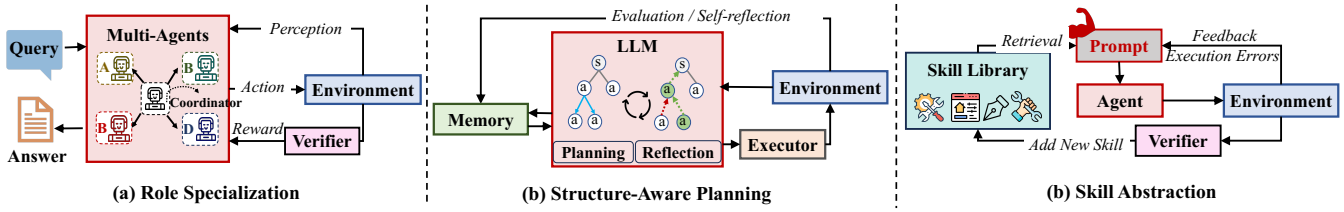


Figure 5: Illustrative examples of agent structural design for data-efficient agentic learning. (a) Role specialization, a representative instantiation of *structural modularity*, where the agent is decomposed into coordinated sub-roles with verifier feedback. (b) Structure-aware planning, a representative instantiation of *procedural autonomy*, where explicit state–action structure with memory, planning, and reflection guides decisions to reduce trial-and-error. (c) Skill abstraction, a representative instantiation of *knowledge reuse*, where the agent retrieves, composes, and verifies reusable skills from a library to avoid repeated low-level interactions.

ing signals without additional interaction cost (Figure 4(b)). This is achieved by integrating real experience with external supervision or structure. Existing approaches span multiple mechanisms: *knowledge distillation* transfers reasoning traces or interactive trajectories from stronger teachers (e.g., MAGDi [Chen *et al.*, 2024b], Agent Distillation [Kang *et al.*, 2025], SAD [Jun *et al.*, 2026]); *experience retrieval* reuses relevant demonstrations or structured knowledge as in-context guidance (e.g., K-RagRec [Wang *et al.*, 2025b]); *cross-task transfer* enables data-scarce tasks to benefit from skills learned in data-rich domains (e.g., PaLM-E [Driess *et al.*, 2023], Magma [Yang *et al.*, 2025a]); and *modality enrichment* aligns multimodal signals to make supervision more explicit and informative (e.g., ADAPT [Lin *et al.*, 2022], NavGPT-2 [Zhou *et al.*, 2024b]). These methods transform existing experience to increase information density and reuse, thereby improving data efficiency.

Category	Core Idea	Core Modules	Representative Works
Structural Modularity	Decompose decision structure	Planner; Action Executor; Critic	DSPy [Khattab *et al.*, 2023]; LLM-Planner [Song *et al.*, 2023]; PEARL [Sun *et al.*, 2024]; AutoAct [Qiao *et al.*, 2024b]; MAD [Liang *et al.*, 2024]
Procedural Autonomy	Constrain execution procedure	Perceiver; Planner; Controller; Verifier	Self-Refine [Madaan *et al.*, 2023]; LATS [Zhou *et al.*, 2024a]; WKM [Qiao *et al.*, 2024a]; Critic [Gou *et al.*, 2024]; VCA [Yang *et al.*, 2025b]; Self-DC [Wang *et al.*, 2025a]
Knowledge Reuse	Reuse prior experience	Memory; Skill library; External KB	WebGLM [Liu *et al.*, 2023]; Reflexion [Shinn *et al.*, 2023]; ExpeL [Zhao *et al.*, 2024]; Voyager [Wang *et al.*, 2024a]; JARVIS-1 [Wang *et al.*, 2024c]; KG-Agent [Jiang *et al.*, 2025a]

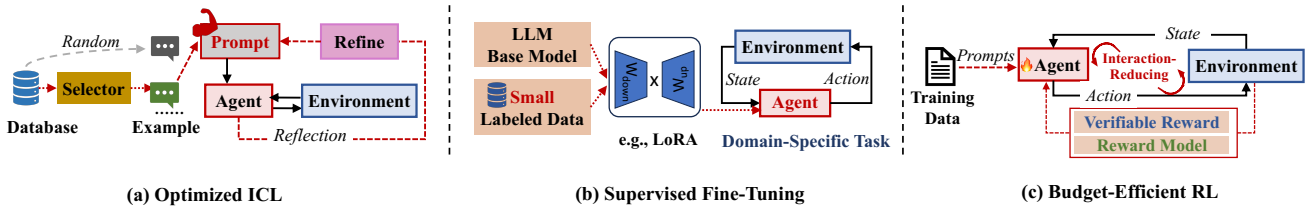
Table 2: Agent structural design for data-efficient agentic learning.

3.2 Agent Structural Design

Agent structural design studies how to reorganize an LLM-based agent's internal structure and execution protocol while holding data sources fixed, so as to increase the utility of each supervision signal or interaction. Rather than acquiring new experience, it focuses on how the agent perceives, reasons, plans, and executes actions through structured internal modules. The goal is to reduce unnecessary trial-and-error and environment steps by localizing supervision to the most informative stages. This often trades additional inference-time computation for fewer costly external interactions. From a data-efficiency perspective, structural design improves performance by shrinking the decision space, preventing costly error propagation, and enabling reuse of plans, skills, and memories. As a result, performance gains increasingly arise from structured internal self-improvement rather than additional external supervision or interaction. We organize existing designs into three categories (Table 2): *structural modularity*, *procedural autonomy*, and *knowledge reuse*, which respectively emphasize modular composition, controlled decision procedures, and reuse of prior knowledge to avoid redundant exploration.

Structural Modularity. This line of work introduces explicit boundaries and interfaces into an agent's internal workflow, transforming an entangled end-to-end reasoning–action process into coordinated components. From a data-efficiency perspective, modularity reduces global trial-and-error by localizing failures, enabling targeted supervision, and promoting reuse of intermediate artifacts. **Function decoupling**, which factorizes monolithic reasoning into planning, execution, and verification modules, allows errors to be corrected locally without restarting the entire decision loop, as exemplified by DSPy [Khattab *et al.*, 2023] and PEARL [Sun *et al.*, 2024]; **hierarchical organization**, which separates high-level subgoal planning from low-level execution, compresses long-horizon decision-making via reusable action executors, as in LLM-Planner [Song *et al.*, 2023]; and **role specialization** (Figure 5 (a)), where different agents or components take stable functional roles and exchange structured feedback, internalizes verification and coordination within the system rather than relying on external supervision, as demonstrated by AutoAct [Qiao *et al.*, 2024b] and MAD [Liang *et al.*, 2024] . These design strategies show that modular composition can substantially reduce external interaction cost and improve sample reuse under fixed data budgets.

Procedural Autonomy. This line of work constrains agent behavior through explicit, reusable decision procedures, replacing unconstrained autoregressive generation with controlled iterative workflows. By deciding what to observe, how to decompose goals, when to act, and when to verify, procedural designs reduce wasted exploration and prevent cascad-



(a) Optimized ICL (b) Supervised Fine-Tuning (c) Budget-Efficient RL

Figure 6: Illustration of learning paradigms for data-efficient agentic learning. (a) Optimized ICL adapts behavior at inference time without parameter updates. (b) PEFT enables persistent adaptation with limited demonstrations via lightweight updates. (c) Budget-efficient RL improves policies under constrained interaction budgets by enhancing learning signals rather than scaling rollouts.

ing errors before costly external actions. **Active perception**, which treats perception as a decision policy over what and when to observe, selectively acquires task-relevant information under limited perception budgets, as in VCA [Yang *et al.*, 2025b]; **task decomposition and structure-aware planning** (Figure 5 (b)), which break long-horizon goals into verifiable substeps and restrict the search space via explicit plans, trees, or divide-and-conquer procedures, reducing blind trial-and-error by enabling backtracking and reuse of partial solutions, as in LATS [Zhou *et al.*, 2024a], WKM [Qiao *et al.*, 2024a] and Self-DC [Wang *et al.*, 2025a]; and **execution control and self-verification**, which gate action execution through intermediate checks and critiques and prevent error propagation before costly external actions, as in Self-Refine [Madaan *et al.*, 2023] and Critic [Gou *et al.*, 2024]. Collectively, these procedural constraints shift performance gains toward structured internal self-improvement rather than additional external supervision or interaction.

Knowledge Reuse. This line of work enables agents to avoid re-learning from scratch by converting available priors—past interactions, acquired skills, and external knowledge—into callable and transferable resources for future decisions. From a data-efficiency perspective, it shifts cost from repeated trial-and-error and human correction to reuse of compact representations that generalize across instances. **Memory compression**, which distills long and context-heavy interaction traces into concise, retrievable summaries or rules, helps agents avoid previously encountered failures without repeating costly exploration, as in Reflexion [Shinn *et al.*, 2023] and ExpeL [Zhao *et al.*, 2024]; **skill abstraction** (Figure 5 (c)), which transforms recurring behavior patterns into reusable subroutines or goal-conditioned controllers, enables compositional reuse of action structure across tasks, as in Voyager [Wang *et al.*, 2024a] and JARVIS-1 [Wang *et al.*, 2024c]; and **external knowledge bases**, which allow agents to retrieve factual or structural priors on demand from the Web or knowledge graphs, thereby reducing reliance on parametric memory and additional supervision, as in WebGLM [Liu *et al.*, 2023] and KG-Agent [Jiang *et al.*, 2025a]. Together, these reuse mechanisms substantially reduce redundant exploration and task-specific supervision by amortizing learning signal across time and tasks.

3.3 Learning Paradigms

When experience augmentation and structural design alone are insufficient, learning becomes unavoidable for improving agent performance. However, naive learning in agentic settings often incurs prohibitive supervision and interaction costs, violating the data-efficiency objective. We therefore regard a learning paradigm as data-efficient only when its gains do not rely on large-scale new supervision or extensive real-environment trial-and-error per task, but instead operate under strictly bounded data and interaction budgets.

Paradigm	Core Idea	Budget Type	Representative Works
Optimized ICL	Adapt without parameter updates	Inference-only	ReAct [Yao *et al.*, 2022]; DSPy [Khattab *et al.*, 2023]; Self-Refine [Madaan *et al.*, 2023]; ExpeL [Zhao *et al.*, 2024]; AdaptAgent [Verma *et al.*, 2025]
PEFT	Partial parameter adaptation	Limited training data	SWE-Gym [Pan *et al.*, 2025]; AutoAct [Qiao *et al.*, 2024b]; DriVLMe [Huang *et al.*, 2024]; FSTTA [Gao *et al.*, 2024]; NavCoT [Lin *et al.*, 2025]
Budget-Efficient RL	Policy learning via a few interactions	Budgeted interaction	CORY [Ma *et al.*, 2024]; AGILE [Feng *et al.*, 2024]; WebRL [Qi *et al.*, 2025]; MAPoRL2 [Park *et al.*, 2025]; GUI-R1 [Luo *et al.*, 2025];

Table 3: Learning paradigms for data-efficient agentic learning.

Under this criterion, we organize existing approaches into three paradigms (Table 3): *optimized in-context learning (ICL)*, which enables inference-time adaptation without parameter updates; *parameter-efficient fine-tuning (PEFT)*, which achieves persistent adaptation via lightweight parameter updates; and *budget-efficient reinforcement learning (RL)*, which improves policies under constrained interaction and credit-assignment budgets. Together, they span a spectrum from transient to persistent adaptation, capturing key trade-offs among learning permanence, data efficiency, and interaction cost.

Optimized In-Context Learning (ICL). This paradigm enables inference-time adaptation by reusing and restructuring contextual information, eliminating the need for parameter updates (Figure 6(a)). Early agentic prompting frameworks such as ReAct [Yao *et al.*, 2022] show that interleaving reasoning, action, and observation within context allows agents to incorporate environment feedback without learning. Subsequent work improves data efficiency by optimizing context quality rather than quantity: Self-Refine [Madaan *et al.*, 2023] and DSPy [Khattab *et al.*, 2023] introduce verification and compilation mechanisms to refine prompts based

on feedback signals, while ExpeL [Zhao *et al.*, 2024] allows the agent to retrieve the top-K relevant examples from its experience pool to aid its decision-making, by reusing trial-and-error experience to mitigate blind exploration. AdaptAgent [Verma *et al.*, 2025] also injects up to two multimodal human demonstrations (e.g., visual snapshots, HTML elements) directly into the prompt of proprietary models like GPT-4o, enabling rapid adaptation to unseen websites. Overall, optimized ICL enables efficient inference-time adaptation of LLM-based agents without parameter updates; recent theoretical analyses further suggest that such behavior can be understood as implicit learning, exhibiting strong few-shot generalization [Wu *et al.*, 2025b].

Parameter-Efficient Supervised Fine-Tuning (PEFT). This paradigm enables persistent supervised adaptation under limited demonstrations or interaction traces. Unlike full-parameter SFT, PEFT updates only a small set of task-specific parameters, such as LoRA modules, adapters, or projection layers, thereby reducing optimization cost and mitigating overfitting and catastrophic forgetting in small-data regimes (Figure 6(b)). In agentic systems, PEFT allows agents to specialize to particular tasks or environments while preserving the generality of large pretrained models. AutoAct [Qiao *et al.*, 2024b] automatically synthesizes trajectories from limited examples and fine-tunes differentiated sub-agents via LoRA, eliminating the need for large-scale annotated data. Across embodied navigation, autonomous driving, and software agents, works such as DriVLMe [Huang *et al.*, 2024], NavCoT [Lin *et al.*, 2025] and SWE-Gym [Pan *et al.*, 2025] show that lightweight supervised adaptation can achieve strong performance with substantially fewer labeled trajectories. Recent extensions further show that such lightweight adaptation can be performed online or at test time [Gao *et al.*, 2024], enabling agents to adapt under strict data budgets without large-scale retraining.

Budget-Efficient Reinforcement Learning (RL). This paradigm improves agent policies under strictly limited interaction budgets, expensive environment access, and sparse or delayed feedback, where conventional RL that scales rollouts becomes impractical (Figure 6(c)). Rather than increasing interactions, recent approaches enhance data efficiency by restructuring experience, shaping rewards, and improving credit assignment. For example, WebRL [Qi *et al.*, 2025] and MAPoRL2 [Park *et al.*, 2025] demonstrate that self-evolving curriculum reuse, verifier-based rewards, and collaborative training can yield substantial gains with far fewer rollouts. AGILE [Feng *et al.*, 2024] frames agent construction as RL over interaction with memory, tools, and expert feedback; CORY [Ma *et al.*, 2024] extends this view to cooperative multi-agent co-evolution for more stable fine-tuning; GUI-R1 [Luo *et al.*, 2025] applies rule-guided RL adaptation to data-limited GUI control. These strategies have proven particularly effective for GUI agents [Luo *et al.*, 2025], highlighting that budget-efficient RL is less about scaling interaction and more about extracting maximal learning signal from minimal experience.

Representative Examples	Modality	Task	Link	Year
Web *(High interaction cost and rapid environment drift.)*				
Mind2Web [Deng *et al.*, 2023]	V–L	Web Navigation	link	2023
WebArena [Zhou *et al.*, 2024c]	L	Web Navigation	link	2024
WebVoyager [He *et al.*, 2024]	V–L	Web Navigation	link	2024
GUI *(High annotation cost and UI drift across versions/devices.)*				
OSWorld [Xie *et al.*, 2024]	V–L	Desktop Automation	link	2024
AndroidWorld [Rawles *et al.*, 2025]	V–L	Mobile App Auto.	link	2025
ScreenSpot-Pro [Li *et al.*, 2025a]	V–L	GUI Grounding	link	2025
Embodied AI *(Costly and safety-constrained real-world interaction.)*				
Franka-Kitchen [Gupta *et al.*, 2020]	3D	Manipulation	link	2020
ALFWorld [Shridhar *et al.*, 2021]	L	Embodied Task Execu.	link	2021
Meta-World [McLean *et al.*, 2025]	3D	Multi-task Manipulation	link	2025
Medical *(Privacy-restricted data and costly expert supervision.)*				
ClinicalBench [Chen *et al.*, 2024a]	L	Clinical Prediction	link	2024
MedRAX [Fallahpour *et al.*, 2025]	V–L	Clinical Diagnosis	link	2025
MedAgentBench [Jiang *et al.*, 2025b]	L	Clinical Decision	link	2025
Science *(Expert-labeled data and expensive experiments.)*				
GPQA [Rein *et al.*, 2024]	L	Scientific Reasoning	link	2024
LAB-Bench [Laurent *et al.*, 2024]	V–L	Biology Research	link	2024
DiscoveryWorld [Jansen *et al.*, 2024]	V–L	Scientific Discovery	link	2024

Table 4: Representative application domains and examples for data-efficient agentic learning. L denotes language-only (text), V–L denotes vision–language, and 3D denotes embodied observations (3D state/environment).

4 Application Domains

We highlight five common application domains: Web, GUI, Embodied AI, Medical, and Science. These domains cover major settings in which LLM-based agents operate through text-based interfaces, vision-language interfaces, physical interaction, expert decision support, and scientific workflows. Across these settings, agent performance depends on both task competence and the ability to acquire, reuse, and transfer learning signals under limited supervision and interaction.

Table 4 summarizes representative examples from these domains to illustrate where data-efficiency bottlenecks arise in practice. Web agents face high interaction cost and rapid environment drift, since navigation trajectories are expensive to collect and websites change over time. GUI agents further depend on fine-grained grounding annotation and robust transfer across layouts, app versions, and device-specific workflows. Embodied agents face slow, costly, and safety-sensitive trial-and-error in physical or simulated environments. Medical agents are constrained by privacy-restricted data and scarce expert verification, while scientific agents often require expert-labeled data, costly experiments, and delayed feedback. These examples therefore serve as representative testbeds for studying how different data-efficient mechanisms, namely experience augmentation, agent structural design, and learning paradigms, translate into practical gains across application domains.

Practical implications. The choice of data-efficient strategy depends on the dominant deployment bottleneck. When labels or demonstrations are scarce but offline synthesis is feasible, experience generation, experience transformation, and PEFT are natural first choices. When real-environment interaction is expensive, unsafe, or slow, interactive simulation, procedural autonomy, and verifier-gated execution are preferable because they reduce external trial-and-error. When

tasks recur or user preferences persist over time, memory, skill reuse, and optimized ICL can amortize limited experience across episodes. When environments or interfaces drift, agents should combine refreshed experience collection or simulation with memory verification and test-time adaptation, rather than relying on stored trajectories alone. These choices reveal a common trade-off: experience augmentation reduces supervision or interaction cost but requires quality control; structural design reduces environment steps but increases inference-time computation; learning paradigms provide stronger or more persistent adaptation but consume training or rollout budget.

5 Open Challenges

Data-efficient learning for LLM agents raises challenges beyond classical sample efficiency because experience is interactive, sequential, heterogeneous, and often costly or stale. Progress requires understanding how limited interaction supports adaptation, how experience can be reused safely, and how agents should be evaluated under realistic supervision, interaction, verification, computation, and personalization budgets. These challenges are tightly coupled: better mechanisms should inform better reuse strategies, while better evaluations should reveal whether reuse truly improves agent behavior under limited resources. Addressing them requires moving from isolated algorithmic improvements toward a more systematic view of agent learning under bounded experience and bounded execution cost.

Mechanisms of Efficient Adaptation. The first challenge is to explain when limited agentic experience yields reliable adaptation. Unlike standard few-shot or sample-efficient reinforcement learning settings [Yu, 2018; Wang *et al.*, 2020], LLM agents reuse closed-loop experience through context, memory, feedback, and test-time adaptation. Existing views of in-context learning as implicit learning [Wu *et al.*, 2025b] do not fully cover long horizons, sparse feedback, drift, and compounding errors [Lin *et al.*, 2025]. Future theory should characterize which trajectories, reasoning structures, memories, or skills remain transferable, and why some experience improves future decisions while other experience induces brittle shortcuts. It is also important to connect external adaptation behavior with internal mechanisms such as attention, gradient flow, numerical precision, and communication structure, which may affect the information yield and stability of limited experience [Zhao *et al.*, 2024; Chen *et al.*, 2026a; Qiu and Yao, 2026; Wu *et al.*, 2026]. Such understanding would help identify when an agent should trust prior experience, seek new interaction, or rely on more conservative reasoning.

Methods for Experience Reuse. The second challenge is to convert raw trajectories, tool-use records, feedback, reflections, and verification outcomes into reliable reusable structures. Memories, skills, transition graphs, task constraints, user models, and shared repositories can reduce repeated exploration, but agents must decide what to store, retrieve, revise, compress, transfer, or forget [Shinn *et al.*, 2023; Zhao *et al.*, 2024]. Reuse is especially difficult under long horizons, personalization, privacy constraints, and multi-agent deployments [Ma *et al.*, 2024; Nie *et al.*, 2025]. Moreover, accumulated or generated experience may contain hallucinated actions, stale knowledge, biased feedback, simulator artifacts, or environment-specific shortcuts. Future methods should therefore combine reuse with filtering, verification, uncertainty estimation, and utility-aware selection, so that stored experience is reused only when it is reliable, transferable, and beneficial for the current task. A further direction is to design reusable experience representations that are compact enough for efficient inference but structured enough to support planning, adaptation, and diagnosis.

Cost-Aware Evaluation in Deployment. The third challenge is to evaluate whether agents improve efficiently, not only whether they eventually succeed. Benchmarks should report interaction steps, tool calls, supervision and verification cost, memory footprint, inference latency, reuse rate, and gain per unit cost, in addition to success rate. This matters for web, GUI, embodied, scientific, medical, and personalized agents, where trial-and-error, expert feedback, privacy, safety, or user-specific criteria make data expensive [Zhou *et al.*, 2024c; Xie *et al.*, 2024; Chen *et al.*, 2024a; Rein *et al.*, 2024; Laurent *et al.*, 2024; Rawles *et al.*, 2025; Swanson *et al.*, 2025; Nie *et al.*, 2026; Chen *et al.*, 2026b]. Evaluation should also diagnose failures such as stale memory reuse, unsafe exploration, premature termination, over-reliance on flawed feedback, or excessive human verification. More realistic protocols should account for latency, energy, memory, and robustness under resource-limited execution, so that progress reflects deployable data efficiency rather than benchmark-specific optimization. This will also make it easier to compare agents that trade off computation, memory, interaction, and supervision in different ways.

6 Conclusion

This survey presents an agent-centric view of data-efficient learning for LLM agents. While scaling remains central, realistic deployment exposes a complementary bottleneck: agents must improve under limited supervision, costly interaction, and changing environments. We broadened data from static labeled examples to agentic data that includes both external supervision and interaction-derived experience, such as demonstrations, trajectories, feedback, verification signals, reasoning traces, and tool-use records, and organized existing methods into three directions: experience augmentation, agent structural design, and learning paradigms. By summarizing representative application domains and open challenges, we argue that the agentic era requires not only larger models, but also smarter mechanisms for acquiring, structuring, reusing, and learning from limited experience.

Acknowledgments

Q. Yao is supported by Beijing Science and Technology Program (No.Z251100008125003). Y. Wang is sponsored by Beijing Nova Program.

References

[Chen *et al.*, 2024a] Canyu Chen, Jian Yu, Shan Chen, Che Liu, Zhongwei Wan, Danielle Bitterman, Fei Wang, and Kai Shu. ClinicalBench: Can LLMs beat traditional ML models in clinical prediction? In *Association for the Advancement of Artificial Intelligence Workshop*, 2024.

[Chen *et al.*, 2024b] Justin Chih-Yao Chen, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. MAGDi: Structured distillation of multi-agent interaction graphs improves reasoning in smaller language models. In *International Conference on Machine Learning*, 2024.

[Chen *et al.*, 2026a] Yihong Chen, Zhouchen Lin, and Quanming Yao. Attention sinks induce gradient sinks: Massive activations as gradient regulators in transformers. *arXiv preprint*, 2026.

[Chen *et al.*, 2026b] Yihong Chen, Shuai Wang, Yaqing Wang, and Quanming Yao. A survey on benchmarks of LLM-based GUI agents. *Transactions on Machine Learning Research*, 2026.

[Cheng *et al.*, 2025] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged robot vision-language-action model for navigation. In *Robotics: Science and Systems*, 2025.

[Deng *et al.*, 2023] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *Neural Information Processing Systems*, 2023.

[Driess *et al.*, 2023] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, et al. PaLM-E: An embodied multimodal language model. In *International Conference on Machine Learning*, 2023.

[Fallahpour *et al.*, 2025] Adibvafa Fallahpour, Jun Ma, Alif Munim, Hongwei Lyu, and BO WANG. MedRAX: Medical reasoning agent for chest X-ray. In *International Conference on Machine Learning*, 2025.

[Feng *et al.*, 2024] Peiyuan Feng, Yichen He, Guanhua Huang, Yuan Lin, Hanchong Zhang, Yuchen Zhang, and Hang Li. AGILE: A novel reinforcement learning framework of LLM agents. In *Neural Information Processing Systems*, 2024.

[Gao *et al.*, 2024] Junyu Gao, Xuan Yao, and Changsheng Xu. Fast-slow test-time adaptation for online vision-and-language navigation. In *International Conference on Machine Learning*, 2024.

[Gou *et al.*, 2024] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024.

[Guan *et al.*, 2023] Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. In *Neural Information Processing Systems*, 2023.

[Guo *et al.*, 2024] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. In *International Joint Conference on Artificial Intelligence*, 2024.

[Gupta *et al.*, 2020] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-Horizon tasks via imitation and reinforcement learning. In *Conference on Robot Learning*, 2020.

[Han *et al.*, 2018] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Neural Information Processing Systems*, 2018.

[Hao *et al.*, 2023] Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Empirical Methods in Natural Language Processing*, 2023.

[He *et al.*, 2024] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. WebVoyager: Building an end-to-end web agent with large multimodal models. In *Association for Computational Linguistics*, 2024.

[Hu *et al.*, 2025] Mengkang Hu, Pu Zhao, Can Xu, Qingfeng Sun, Jian-Guang Lou, Qingwei Lin, Ping Luo, and Saravan Rajmohan. AgentGen: Enhancing planning abilities for large language model based agent via environment and task generation. In *Knowledge Discovery and Data Mining*, 2025.

[Huang *et al.*, 2024] Yidong Huang, Jacob Sansom, Ziqiao Ma, Felix Gervits, and Joyce Chai. DriVLMe: Enhancing LLM-based autonomous driving agents with embodied and social experiences. In *Intelligent Robots and Systems*, 2024.

[Jansen *et al.*, 2024] Peter Jansen, Marc-Alexandre Côté, Tushar Khot, Erin Bransom, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Oyvind Tafjord, and Peter Clark. DiscoveryWorld: A virtual environment for developing and evaluating automated scientific discovery agents. In *Neural Information Processing Systems*, 2024.

[Jiang *et al.*, 2025a] Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, Yang Song, Chen Zhu, Hengshu Zhu, and Ji-Rong Wen. KG-Agent: An efficient autonomous agent framework for complex reasoning over knowledge graph. In *Association for Computational Linguistics*, 2025.

[Jiang *et al.*, 2025b] Yixing Jiang, Kameron C. Black, Gloria Geng, Danny Park, James Zou, Andrew Y. Ng, and Jonathan H. Chen. MedAgentBench: A virtual EHR environment to benchmark medical LLM agents. *New England Journal of Medicine-Artificial Intelligence*, 2025.

[Jun *et al.*, 2026] Liu Jun, Kong Zhenglun, Dong Peiyan, Yang Changdi, Li Tianqin, Xie Yanyue, Gong Yifan, Shen Xuan, Zhao Pu, Tang Hao, et al. Structured agent distillation for large language model agents. In *Autonomous Agents and Multiagent Systems*, 2026.

[Kang *et al.*, 2025] Minki Kang, Jongwon Jeong, Seanie Lee, Jaewoong Cho, and Sung Ju Hwang. Distilling LLM agent into small models with retrieval and code tools. In *Neural Information Processing Systems*, 2025.

[Khattab *et al.*, 2023] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, et al. DSPy: Compiling declarative language model calls into self-improving pipelines. In *Neural Information Processing Systems*, 2023.

[Laurent *et al.*, 2024] Jon M. Laurent, Joseph D. Janizek, Michael Ruzo, Michaela M. Hinks, Michael J. Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D. White,

and Samuel G. Rodriques. LAB-Bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362*, 2024.

[Li *et al.*, 2025a] Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. ScreenSpot-Pro: GUI grounding for professional high-resolution computer use. In *International Conference on Multimedia*, 2025.

[Li *et al.*, 2025b] Yuan Li, LiChao Sun, and Yixuan Zhang. MetaAgents: Large language model based agents for decision-making on teaming. *HCI*, 2025.

[Liang *et al.*, 2024] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Empirical Methods in Natural Language Processing*, 2024.

[Lin *et al.*, 2022] Bingqian Lin, Yi Zhu, Zicong Chen, Xiwen Liang, Jianzhuang Liu, and Xiaodan Liang. ADAPT: Vision-language navigation with modality-aligned action prompts. In *Computer Vision and Pattern Recognition*, 2022.

[Lin *et al.*, 2025] Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. NavCoT: Boosting LLM-based vision-and-language navigation via learning disentangled reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[Liu *et al.*, 2023] Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. WebGLM: Towards an efficient web-enhanced question answering system with human preferences. In *Knowledge Discovery and Data Mining*, 2023.

[Liu *et al.*, 2025a] Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, et al. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*, 2025.

[Liu *et al.*, 2025b] Zhipeng Liu, Xuefeng Bai, Kehai Chen, Xinyang Chen, Xiucheng Li, Yang Xiang, Jin Liu, Hong-Dong Li, Yaowei Wang, et al. A survey on the feedback mechanism of LLM-based AI agents. In *International Joint Conference on Artificial Intelligence*, 2025.

[Luo *et al.*, 2025] Rui Luo, Lu Wang, Wanwei He, Longze Chen, Jiaming Li, and Xiaobo Xia. GUI-R1: A generalist R1-style vision-language action model for GUI agents. *arXiv preprint arXiv:2504.10458*, 2025.

[Ma *et al.*, 2024] Hao Ma, Tianyi Hu, Zhiqiang Pu, Boyin Liu, Xiaolin Ai, Yanyan Liang, and Min Chen. Coevolving with the other you: Fine-tuning LLM with sequential cooperative multi-agent reinforcement learning. In *Neural Information Processing Systems*, 2024.

[Madaan *et al.*, 2023] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, et al. Self-refine: Iterative refinement with self-feedback. In *Neural Information Processing Systems*, 2023.

[McLean *et al.*, 2025] Reginald McLean, Evangelos Chatzaroulas, Luc McCutcheon, Frank Röder, Tianhe Yu, Zhanpeng He, K.R. Zentner, Ryan Julian, J K Terry, Isaac Woungang, et al. Meta-World+: An improved, standardized, RL benchmark. In *Neural Information Processing Systems*, 2025.

[Nie *et al.*, 2025] Hongyi Nie, Yaqing Wang, Mingyang Zhou, Feiyang Pan, Quanming Yao, and Zhen Wang. Adaptive preference arithmetic: Modeling dynamic preference strengths for LLM agent personalization. In *Neural Information Processing Systems*, 2025.

[Nie *et al.*, 2026] Hongyi Nie, Xunyuan Liu, Yudong Bai, Yaqing Wang, Yang. Liu, Quanming Yao, and Zhen Wang. PSPA-Bench: A personalized benchmark for smartphone GUI agent. *arXiv preprint*, 2026.

[Ouyang *et al.*, 2022] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, et al. Training language models to follow instructions with human feedback. In *Neural Information Processing Systems*, 2022.

[Pan *et al.*, 2025] Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. Training software engineering agents and verifiers with SWE-Gym. In *International Conference on Machine Learning*, 2025.

[Park *et al.*, 2025] Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman E. Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. MAPoRL: Multi-agent post-co-training for collaborative large language models with reinforcement learning. In *Association for Computational Linguistics*, 2025.

[Qi *et al.*, 2025] Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, et al. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, 2025.

[Qiao *et al.*, 2024a] Shuofei Qiao, Runnan Fang, Ningyu Zhang, Yuqi Zhu, Xiang Chen, Shumin Deng, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Agent planning with world knowledge model. In *Neural Information Processing Systems*, 2024.

[Qiao *et al.*, 2024b] Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo, Wangchunshu Zhou, Yuchen Eleanor Jiang, Chengfei Lv, and Huajun Chen. AutoAct: Automatic agent learning from scratch for QA via self-planning. In *Association for Computational Linguistics*, 2024.

[Qiu and Yao, 2026] Haiquan Qiu and Quanming Yao. Why low-precision transformer training fails: An analysis on flash attention. In *International Conference on Learning Representations*, 2026.

[Rawles *et al.*, 2025] Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William E Bishop, Wei Li, et al. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In *International Conference on Learning Representations*, 2025.

[Rein *et al.*, 2024] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *COLM*, 2024.

[Sang *et al.*, 2025] Jitao Sang, Jinlin Xiao, Jiarun Han, Jilin Chen, Xiaoyi Chen, Shuyu Wei, Yongjie Sun, and Yuhang Wang. Beyond pipelines: A survey of the paradigm shift toward model-native agentic ai, 2025.

[Shinn *et al.*, 2023] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Neural Information Processing Systems*, 2023.

[Shridhar *et al.*, 2021] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J.

Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

[Song *et al.*, 2023] Chan Hee Song, Brian M. Sadler, Jiaman Wu, Wei-Lun Chao, Clayton Washington, and Yu Su. LLM-Planner: Few-shot grounded planning for embodied agents with large language models. In *International Conference on Computer Vision*, 2023.

[Sun *et al.*, 2024] Simeng Sun, Yang Liu, Shuohang Wang, Dan Iter, Chenguang Zhu, and Mohit Iyyer. PEARL: Prompting large language models to plan and execute actions over long documents. In *European Chapter of the Association for Computational Linguistics*, 2024.

[Swanson *et al.*, 2025] Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak, and James Zou. The virtual lab of AI agents designs new SARS-CoV-2 nanobodies. *Nature*, 2025.

[Takerngsaksiri *et al.*, 2025] Wannita Takerngsaksiri, Jirat Pasuksmit, Patanamon Thongtanunam, Chakkrit Tantithamthavorn, Ruixiong Zhang, Fan Jiang, Jing Li, Evan Cook, Kun Chen, et al. Human-in-the-loop software development agents. In *Software Engineering: Software Engineering in Practice*, 2025.

[Torreno *et al.*, 2017] Alejandro Torreno, Eva Onaindia, Antonín Komenda, and Michal Štolba. Cooperative multi-agent planning: A survey. *ACM Computing Surveys*, 2017.

[Verma *et al.*, 2025] Gaurav Verma, Rachneet Kaur, Nishan Srishankar, Zhen Zeng, Tucker Balch, and Manuela Veloso. AdaptAgent: Adapting multimodal web agents with few-shot learning from human demonstrations. In *Association for Computational Linguistics*, 2025.

[Wang *et al.*, 2020] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys*, 2020.

[Wang *et al.*, 2024a] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024.

[Wang *et al.*, 2024b] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 2024.

[Wang *et al.*, 2024c] Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Zhaofeng He, Zilong Zheng, et al. JARVIS-1: Open-world multi-task agents with memory-augmented multimodal language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[Wang *et al.*, 2025a] Hongru Wang, Boyang Xue, Baohang Zhou, Tianhua Zhang, Cunxiang Wang, Huimin Wang, Guanhua Chen, and Kam-Fai Wong. Self-DC: When to reason and when to act? self divide-and-conquer for compositional unknown questions. In *North American Chapter of the Association for Computational Linguistics*, 2025.

[Wang *et al.*, 2025b] Shijie Wang, Wenqi Fan, Yue Feng, Shanru Lin, Xinyu Ma, Shuaiqiang Wang, and Dawei Yin. Knowledge graph retrieval-augmented generation for LLM-based recommendation. In *Association for Computational Linguistics*, 2025.

[Wei *et al.*, 2024] Yuxi Wei, Zi Wang, Yifan Lu, Chenxin Xu, Changxing Liu, Hao Zhao, Siheng Chen, and Yanfeng Wang. Editable scene simulation for autonomous driving via collaborative LLM-agents. In *Computer Vision and Pattern Recognition*, 2024.

[Wu *et al.*, 2025a] Di Wu, Xian Wei, Guang Chen, Hao Shen, and Bo Jin. Generative multi-agent collaboration in embodied AI: A systematic review. In *International Joint Conference on Artificial Intelligence*, 2025.

[Wu *et al.*, 2025b] Shiguang Wu, Yaqing Wang, and Quanming Yao. Why in-context learning models are good few-shot learners? In *International Conference on Learning Representations*, 2025.

[Wu *et al.*, 2026] Shiguang Wu, Yaqing Wang, and Quanming Yao. Language model networks: Supervision-efficient learning through dense communication. In *International Conference on Machine Learning*, 2026.

[Xie *et al.*, 2024] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Jing Hua Toh, Zhoujun Cheng, Dongchan Shin, et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Neural Information Processing Systems*, 2024.

[Yang *et al.*, 2025a] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. In *Conference on Computer Vision and Pattern Recognition*, 2025.

[Yang *et al.*, 2025b] Zeyuan Yang, Delin Chen, Xueyang Yu, Maohao Shen, and Chuang Gan. VCA: Video curious agent for long video understanding. In *International Conference on Computer Vision*, 2025.

[Yao *et al.*, 2022] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2022.

[Yu, 2018] Yang Yu. Towards sample efficient reinforcement learning. In *International Joint Conference on Artificial Intelligence*, 2018.

[Zhang *et al.*, 2025] Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 2025.

[Zhang *et al.*, 2026] Lingfeng Zhang, Haoxiang Fu, Xiaoshuai Hao, Shuyi Zhang, Qiang Zhang, Rui Liu, Long Chen, and Wenbo Ding. What you see is what you reach: Towards spatial navigation with high-level human instructions. *Association for the Advancement of Artificial Intelligence*, 2026.

[Zhao *et al.*, 2024] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Association for the Advancement of Artificial Intelligence*, 2024.

[Zhou *et al.*, 2024a] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *International Conference on Machine Learning*, 2024.

[Zhou *et al.*, 2024b] Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. NavGPT-2: Unleashing navigational reasoning capability for large vision-language models. In *ECCV*, 2024.

[Zhou *et al.*, 2024c] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, et al. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024.