Beyond Scaling: A Survey on Data-Efficient Agentic Learning

Yaqing Wang[1] , **Zhenlin Luo**[1,2] , **Peiyao Zhao**[1,3] , **Yunfeng Cai**[1] , **Quanming Yao**[4]
[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
{wangyaqing, luozhenlin, zhaopeiyao, caiyunfeng}@bimsa.cn, qyaoaa@tsinghua.edu.cn

Abstract

LLM-based agents are increasingly deployed across web and GUI automation, embodied decision making, and scientific workflows, yet their progress is often constrained by limited data and interaction. High-quality supervision is costly, and real-environment interactions are expensive, risky, and quickly invalidated by environment drift. This survey studies how to obtain and improve LLM-based agents with fewer samples, fewer labels, and fewer/ cheaper interactions. We view agentic learning as a closed-loop decision process where experience arises from both external supervision and online interactions, and data efficiency requires maximizing information yield per unit cost. We then introduce a unified agentic learning framework and organize the literature along three complementary dimensions: experience augmentation, agent structural design, and learning paradigms. This perspective connects design choices to where learning signals come from, how they are utilized, and how adaptation is performed under bounded budgets. We summarize representative benchmarks and synthesize key open challenges, aiming to clarify the emerging landscape and support future progress in data-efficient agentic learning.

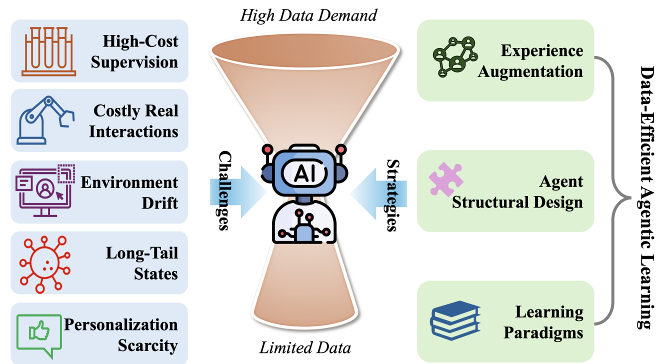

Figure 1: Conceptual overview of data-efficient agentic learning and its three complementary strategies.

1 Introduction

Large language model (LLM)-based agents are rapidly moving beyond prompt-only prototypes into closed-loop systems that can perceive, reason/plan, and act in dynamic environments. This shift marks a transition from "model-as-a-service" to "model-as-an-agent": success is no longer determined by one-shot generation quality, but by whether the agent can reliably acquire, verify, and refine behaviors while acting under dynamic feedback and long-horizon goals. Recent progress has enabled such agents to operate across a wide range of practical settings—from web and GUI automation to embodied decision making and scientific or medical workflows—where perception, reasoning, and action execution must be coordinated end-to-end [Zhou *et al.*, 2024c; Xie *et al.*, 2024; Shridhar *et al.*, 2021; Laurent *et al.*, 2024].

In these realistic deployments, the dominant bottleneck is increasingly data efficiency rather than model scaling. Downstream agent tasks are often intrinsically data-scarce: high-quality supervision may be unavailable (e.g., it is unclear how to label every intermediate decision in an interactive trajectory), or feasible but prohibitively expensive (e.g., step-level grounding labels, demonstrations, expert feedback, or verification). Meanwhile, online trial-and-error is not "free data": it consumes environment steps and tool calls, can require human verification, and may introduce safety or reliability risks, especially in high-stakes scenarios. Compounding the problem, interaction data can become stale under environment or interface drift—a common issue in software automation [Zhou *et al.*, 2024c; Xie *et al.*, 2024]—and similar constraints arise in embodied, scientific, and medical workflows where privacy, expert time, or experimental verification costs dominate [Chen *et al.*, 2024; Rein *et al.*, 2024; Laurent *et al.*, 2024]. As a result, practitioners often have to obtain and improve agents with whatever limited signal is available, making "how to learn/obtain a capable LLM-based agent in a data-efficient way" a first-order question.

Data efficiency has been a central theme in machine learning, spanning few-shot learning [Wang *et al.*, 2020] and sample-efficient reinforcement learning (RL) [Yu, 2018]. Yet agentic learning fundamentally broadens what "data" means and where efficiency comes from. Beyond labeled examples, agents learn from human annotations, trajectories, intermediate reasoning traces, tool-use patterns, and verification outcomes [Yao *et al.*, 2022; Shinn *et al.*, 2023]. Efficiency is therefore no longer solely a property of a learning algorithm; it emerges from the joint design of (i) experience and how it is generated, transformed, or simulated to reduce reliance on costly supervision, (ii) agent structure—including specialized perceivers and action executors—that reduces wasted interactions and localizes errors, and (iii) learning paradigms that maximize information gain from limited samples, labels, and interactions by governing whether and how model parameters are updated. Recent theoretical analyses further suggest that in-context adaptation can be understood as a form of implicit learning, helping explain strong few-shot generalization behaviors in modern LLMs [Wu *et al.*, 2025b].

Despite the rapidly growing literature, existing reviews of LLM-based agents [Wang *et al.*, 2024b; Sang *et al.*, 2025; Liu *et al.*, 2025a] typically emphasize broad architectural scope or focus on individual components such as multi-agent architectures [Wu *et al.*, 2025a; Guo *et al.*, 2024], feedback mechanisms [Liu *et al.*, 2025b], memory designs [Zhang *et al.*, 2025b], or planning patterns [Torreno *et al.*, 2017]. They have not explicitly centered bounded supervision and interaction budgets as the organizing principle that connects techniques across experience acquisition, agent structure, and learning dynamics. At the same time, the growing scale and diversity of recent work make a unified, agent-centric synthesis along this dimension both timely and valuable.

In this survey, we provide a review of data-efficient agentic learning (Figure 1). Concretely, we make the following contributions: We introduce a unified agentic learning framework and a data-efficiency criterion grounded in limited samples, labels, and interactions. We organize the literature into a taxonomy along three complementary dimensions: (i) experience augmentation, (ii) agent structural design, and (iii) learning paradigms, connecting where learning signals originate, how they are utilized, and how adaptation is performed under bounded budgets. Finally, we summarize representative benchmarks across application domains and discuss open challenges that shape future progress.

2 Overview

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 (often via tools), and receives feedback through interaction. At time step t, the environment is in state $s_t \in \mathcal{S}$ and the agent follows the interaction loop shown in Figure 2:

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

Here P denotes a perceiver that maps the environment state to an observation, 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 internal state m_t, and E is an action executor that converts the selected action into an executable form. The interaction yields online experience $\mathcal{D}_o = \{(s_t, o_t, a_t, \ldots)\}_{t=1}^{T}$, while the agent may also leverage external experience $\mathcal{D}_e$ collected outside its own interaction loop (e.g., demonstrations, labels, preference feedback, or verified outcomes). We denote the available experience by $\mathcal{D} = \mathcal{D}_o \cup \mathcal{D}_e$.

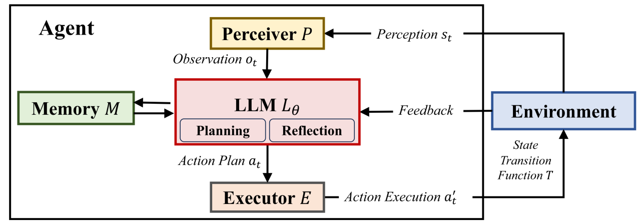

Figure 2: Agentic learning loop with core components.

Then, we define data-efficient agentic learning as follows.

Definition 1 (Data-Efficient Agentic Learning)**.** *Data-efficient agentic learning studies how to obtain and improve LLM-based agents that operate in interactive decision-making settings under limited available experience $\mathcal{D}$.*

In this survey, we emphasize three coupled aspects. First, agentic interaction refers to a closed-loop process in which an agent repeatedly perceives the environment, reasons and plans with an LLM, executes actions, and incorporates feedback over time. Second, learning signals arise from both online interactions (trajectories, feedback, verification outcomes) and external supervision (demonstrations, labels, preference feedback, curated data). Third, an approach is data-efficient if its performance gains do not rely on collecting large amounts of new supervision or extensive real-environment trial-and-error, but instead improve the information yield per unit data and interaction.

Definition 1 highlights that data-efficiency bottlenecks stem from both costly supervision in $\mathcal{D}_e$ and costly real-environment interaction in $\mathcal{D}_o$. Accordingly, this survey organizes existing methods along three complementary design levers that act on different parts of the agentic loop. Experience augmentation focuses on expanding the effective experience $\mathcal{D}$ without proportional increases in real interaction (Section 3.1). Agent structural design reorganizes the internal modules and execution protocol (e.g., perceiver, memory, planning, reflection, and action executor) so that interactions become more directed, verifiable, and reusable, reducing redundant trial-and-error (Section 3.2). Learning paradigms characterize how agents are adapted from limited data and interaction (Section 3.3). Section 3 then elaborates this taxonomy and reviews representative methods in each category.

3 Taxonomy

We now elaborate the taxonomy motivated by Definition 1 and Figure 2. The following three subsections review three complementary aspects of data-efficient agentic learning introduced in Section 2. For each aspect, we summarize its core

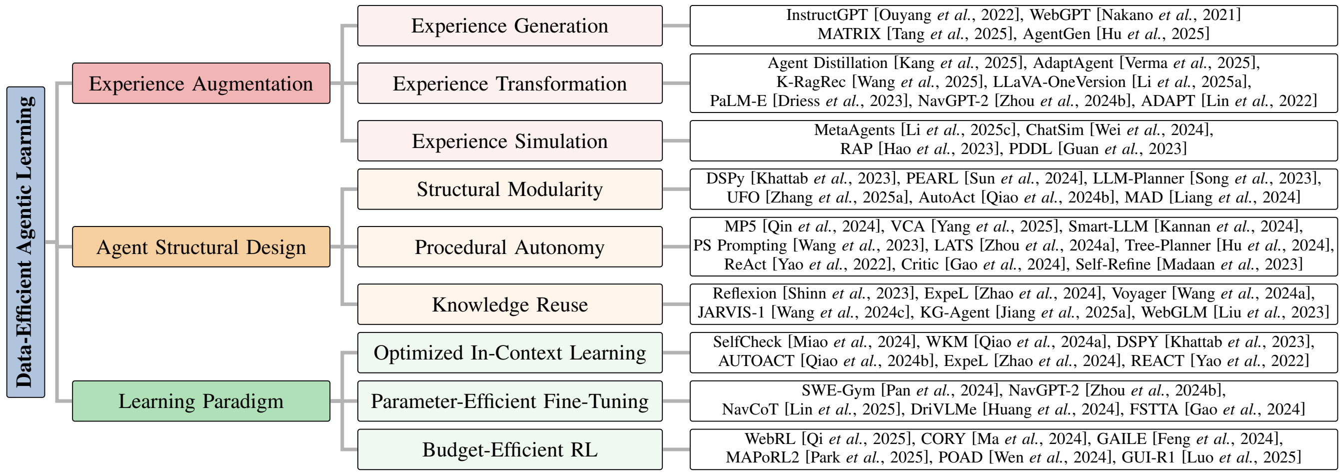

Figure 3: Taxonomy of data-efficient agentic learning.

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 and quality of external experience rather than model capacity. 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 collecting more data, its goal is to increase the density of task-relevant learning signal and reusable behavioral structure per unit of real experience. We organize existing approaches into three categories (Table 1): *experience generation*, *experience transformation*, and *experience simulation*.

Category	Core Idea	Data Type	Representative Works
Experience Generation	Create new high-quality data	S, L, I	InstructGPT [Ouyang *et al.*, 2022]; WebGPT [Nakano *et al.*, 2021]; MATRIX [Tang *et al.*, 2025]; AgentGen [Hu *et al.*, 2025]
Experience Transformation	Increase information density	S, L	Agent Distillation [Kang *et al.*, 2025]; AdaptAgent [Verma *et al.*, 2025]; PaLM-E [Driess *et al.*, 2023]; NavGPT-2 [Zhou *et al.*, 2024b]
Experience Simulation	Shift interaction to cheaper surrogates	S, I	MetaAgents [Li *et al.*, 2025c]; ChatSim [Wei *et al.*, 2024]; RAP [Hao *et al.*, 2023]; PDDL [Guan *et al.*, 2023]

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

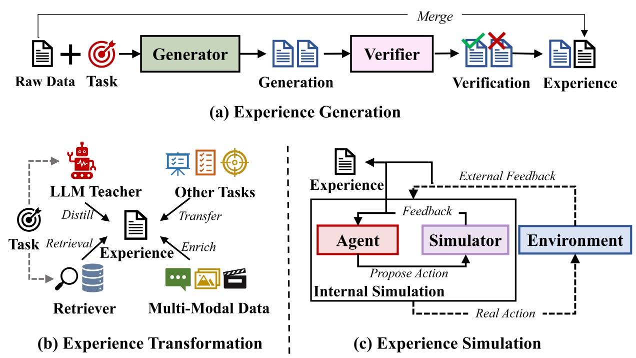

Figure 4: Illustration of experience augmentation strategy for data-efficient agentic learning. (a) Experience generation synthesizes additional training experience to expand coverage under limited interaction budgets. (b) Experience transformation enriches and restructures limited real experience into more reusable training signals. (c) Experience simulation replaces costly real-world interaction with simulated or modeled environments.

Experience Generation. This category expands the effective experience pool beyond what is directly collected from the real environment, targeting coverage gaps and recurring failure modes under limited interaction budgets (Figure 4(a)). Instead of unconstrained data synthesis, experience generation focuses on producing high-quality trajectories that expose diverse error cases while maintaining reliable supervision. 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], WebGPT [Nakano *et al.*, 2021]. It can further scale through *model-centric* generation, where agents synthesize additional trajectories at low marginal cost and rely on verification or structured feedback to control error propagation; representative examples include MATRIX [Tang *et al.*, 2025], which generates interaction data via structured multi-agent simulation, and AgentGen [Hu *et al.*, 2025], which expands coverage through environment-conditioned trajectory synthesis without additional real-world interaction. These methods improves data efficiency by creating additional training signals to expand coverage under limited supervision.

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 learning signals without additional interaction cost (Figure 4(b)). This is achieved by integrating real experience with external supervision or structure, followed by systematic reprocessing such as filtering, rewriting, relabeling, or compression. Existing approaches span multiple mechanisms: *knowledge distillation* transfers reasoning traces or interactive trajectories from stronger teachers (e.g., Agent Distillation [Kang *et al.*, 2025]); *experience retrieval* reuses relevant demonstrations or structured knowledge as in-context guidance (e.g., AdaptAgent [Verma *et al.*, 2025], K-RagRec [Wang *et al.*, 2025]); *cross-task transfer* enables data-scarce tasks to benefit from skills learned in data-rich domains (e.g., PaLM-E [Driess *et al.*, 2023], LLaVA-OneVision [Li *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.

Experience Simulation. This category reduces reliance on expensive real-world interaction by shifting exploration and failure discovery to cheaper surrogate environments (Figure 4(c)). 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. Systems such as MetaAgents [Li *et al.*, 2025c] and ChatSim [Wei *et al.*, 2024] demonstrate the utility of controllable simulated environments for generating targeted and rare interaction scenarios, while 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. In sum, they improves data efficiency by substituting costly real-world interaction with lower-cost interaction sources.

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.

Category	Core Idea	Core Modules	Representative Works
Structural Modularity	Decompose decision structure	Planner; Action Executor; Critic	DSPy [Khattab *et al.*, 2023]; PEARL [Sun *et al.*, 2024]; LLM-Planner [Song *et al.*, 2023]; UFO [Zhang *et al.*, 2025a]; AutoAct [Qiao *et al.*, 2024b]; MAD [Liang *et al.*, 2024]
Procedural Autonomy	Constrain execution procedure	Perceiver; Planner; Controller; Verifier	MP5 [Qin *et al.*, 2024]; VCA [Yang *et al.*, 2025]; Plan-and-Solve [Wang *et al.*, 2023]; LATS [Zhou *et al.*, 2024a]; ReAct [Yao *et al.*, 2022]; Self-Refine [Madaan *et al.*, 2023]
Knowledge Reuse	Reuse prior experience	Memory; Skill library; External KB	Reflexion [Shinn *et al.*, 2023]; ExpeL [Zhao *et al.*, 2024]; Voyager [Wang *et al.*, 2024a]; JARVIS [Wang *et al.*, 2024c]; WebGLM [Liu *et al.*, 2023]; KG-Agent [Jiang *et al.*, 2025a]

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

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 UFO [Zhang *et al.*, 2025a]; 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 cascading errors before costly external actions. **Active perception**, which treats perception as a decision policy over what

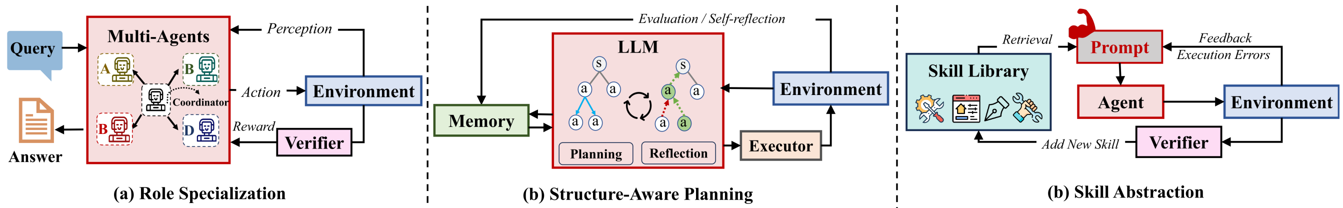

Figure 5: Illustrative examples of agent structural design for data-efficient agentic learning. (a) Role specialization, a representative instantiation of *structural modularity*, where the agent is decomposed into coordinated sub-roles with verifier feedback. (b) Structure-aware planning, a representative 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.

and when to observe, selectively acquires task-relevant information under limited perception budgets, as in MP5 [Qin *et al.*, 2024] and VCA [Yang *et al.*, 2025]; **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 or trees, reduce blind trial-and-error by enabling backtracking and reuse of partial solutions, as in Plan-and-Solve Prompting [Wang *et al.*, 2023], Smart-LLM [Kannan *et al.*, 2024], LATS [Zhou *et al.*, 2024a], and Tree-Planner [Hu *et al.*, 2024]; and **execution control and self-verification**, which gate action execution through intermediate checks and critique, prevent error propagation before costly external actions, as in ReAct [Yao *et al.*, 2022], 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**, where agents retrieve factual or structural priors on demand from the Web or knowledge graphs, reduce 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 Paradigm

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.

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.

Paradigm	Core Idea	Budget Type	Representative Works
Optimized ICL	Adapt without parameter updates	Inference-only	ReAct [Yao *et al.*, 2022]; SelfCheck [Miao *et al.*, 2024]; DSPy [Khattab *et al.*, 2023]; ExpeL [Zhao *et al.*, 2024]; WKM [Qiao *et al.*, 2024a]; AutoAct [Qiao *et al.*, 2024b]
PEFT	Partial parameter adaptation	Limited training data	SWE-Gym [Pan *et al.*, 2024]; NavGPT-2 [Zhou *et al.*, 2024b]; NavCoT [Lin *et al.*, 2025]; DriVLMe [Huang *et al.*, 2024]; FSTTA [Gao *et al.*, 2024]
Budget-Efficient RL	Policy learning via a few interaction	Budgeted interaction	WebRL [Qi *et al.*, 2025]; CORY [Ma *et al.*, 2024]; GAILE [Feng *et al.*, 2024]; MAPoRL2 [Park *et al.*, 2025]; POAD [Wen *et al.*, 2024]; GUI-R1 [Luo *et al.*, 2025]

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

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

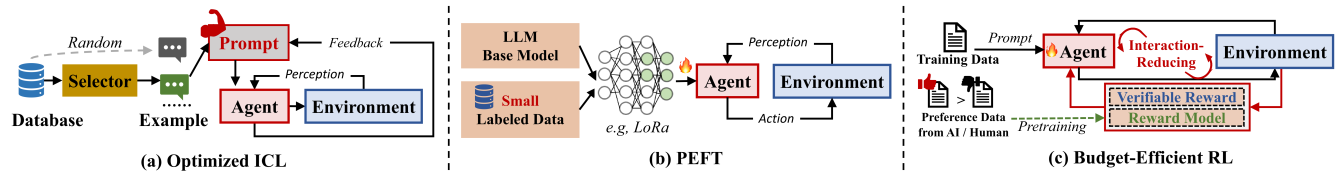

Figure 6: Illustration of learning paradigms for data-efficient agentic learning. (a) Optimized ICL adapts behavior at inference time without parameter updates. (b) PEFT enables persistent adaptation with limited demonstrations via lightweight updates. (c) Budget-efficient RL improves policies under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

agents to incorporate environment feedback without learning. Subsequent work improves data efficiency by optimizing context quality rather than quantity: SelfCheck [Miao *et al.*, 2024] and DSPy [Khattab *et al.*, 2023] introduce verification and compilation mechanisms to refine prompts based on feedback signals, while knowledge-based approaches (e.g., ExpeL [Zhao *et al.*, 2024], WKM [Qiao *et al.*, 2024a]) reuse distilled trial-and-error experience and inject task knowledge as in-context guidance to mitigate blind exploration and hallucinated actions. AutoAct [Qiao *et al.*, 2024b] further demonstrates that agents can bootstrap high-quality contextual trajectories through self-instruction and structured reflection under minimal human supervision. Overall, optimized in-context learning 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 Fine-Tuning (PEFT). This paradigm enables persistent task adaptation under limited supervision by updating only a small subset of parameters, reducing data demand, optimization cost, and the risk of catastrophic forgetting compared with full fine-tuning (Figure 6(b)). In agentic systems, PEFT allows specialization from a small number of demonstrations or interaction traces while preserving the generality of large pretrained models. Across embodied navigation, autonomous driving, and software agents, works such as NavGPT-2 [Zhou *et al.*, 2024b], NavCoT [Lin *et al.*, 2025], DriVLMe [Huang *et al.*, 2024], and SWE-Gym [Pan *et al.*, 2024] demonstrate that adopting lightweight adapters or projections can achieve strong performance with orders of magnitude fewer labeled trajectories. Recent extensions further show that PEFT can be applied 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. Fine-grained credit assignment methods such as POAD [Wen *et al.*, 2024] further accelerate learning by yielding finer-grained credit assignment without additional interactions. 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.

4 Applications and Benchmarks

We highlight five representative application domains—Web, GUI, Embodied AI, Medical, and Science—that capture the primary real-world settings in which data-efficient agentic learning is most critical. These domains span common agent interaction modalities, from text-based and vision-language interfaces to long-horizon decision making in physical and scientific workflows. Across all five domains, agents face similar structural constraints: effective interaction trajectories are costly to obtain, fine-grained grounding or expert supervision is expensive, environments and user interfaces evolve over time, and safety, privacy, or experimental considerations restrict large-scale trial-and-error. As a result, agent performance in these settings depends not only on task competence, but on how efficiently learning signals are acquired, reused, and transferred under limited supervision and interaction.

Table 4 lists a small set of widely used benchmarks selected to provide empirical grounding for these challenges. Rather than offering an exhaustive benchmark survey, we focus on benchmarks that (i) adopt relatively stable evaluation protocols, (ii) instantiate explicit perception–decision–action loops, and (iii) expose dominant sources of data scarcity, including interaction cost, labeling or expert supervision cost, environment or interface drift, and regulatory or safety constraints. These benchmarks therefore serve as representative testbeds for studying how different data-efficient mechanisms—experience augmentation, agent structural design, and learning paradigms—translate into practical gains across application domains.

5 Open Challenges

Data-efficient agentic learning departs from classical notions of sample efficiency because agent data is interactive, sequential, and heterogeneous across tasks, environments, and users. Each step can generate not only observations and rewards, but also reasoning traces, tool-use patterns, and verification signals that affect future decisions. When interaction, supervision, verification, and personalization all carry real cost, several challenges become central to making data-efficient agents practical and reliable.

Benchmark	Modality	Task	Link	Year
Web *(High interaction cost and rapid environment drift.)*				
WebArena [Zhou *et al.*, 2024c]	L	Web Navigation	link	2023
Mind2Web [Deng *et al.*, 2023]	V–L	Web Navigation	link	2023
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 Automation	link	2024
ScreenSpot [Li *et al.*, 2025b]	V–L	GUI Grounding	link	2025
Embodied AI *(Costly and safety-constrained real-world interaction.)*				
Franka-Kitchen [Gupta *et al.*, 2020]	3D	Manipulation	link	2019
ALFWorld [Shridhar *et al.*, 2021]	L	Embodied Task Execution	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.*, 2024]	L	Clinical Prediction	link	2024
MedRAX [Fallahpour *et al.*, 2025]	V–L	Clinical Diagnosis	link	2025
MedAgentBench [Jiang *et al.*, 2025b]	L	Clinical Decision Making	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 benchmarks for data-efficient agentic learning across application domains. L denotes language-only (text), V–L denotes vision–language, and 3D denotes embodied observations (3D state/environment).

Long-Horizon Learning. Many agentic tasks require long-horizon decision making, where errors compound and the burden of verification, credit assignment, and exploration escalates quickly [Lin *et al.*, 2025]. Although PEFT and budget-efficient RL reduce parameter-update cost, long-horizon settings often still incur substantial interaction and supervision overhead. A key direction is to make horizon a first-class factor in agent design and learning: agents should reason with checkpoints, reuse intermediate artifacts, and allocate verification strategically, rather than relying on naive rollout scaling.

Generalization and Drift. General-purpose agents must transfer across tasks, tools, environments, and deployment conditions while relying on limited data and interactions. Since exhaustive coverage is infeasible, robust generalization hinges on learning reusable abstractions of reasoning, planning, and interaction rather than fitting individual trajectories [Zhao *et al.*, 2024]. This brings forward practical questions about what should be abstracted (e.g., decomposition patterns and tool-use strategies), how sparse interactions should trigger adaptation, and how to remain stable under distribution shift and environment/UI drift.

Personalization and User-Centric Learning. Many real-world agents operate in personalized settings, where behavior must adapt to individual users, preferences, and constraints [Nie *et al.*, 2025]. Personalization is intrinsically data-scarce: each user induces a distinct interaction distribution, and feedback is often implicit, noisy, or delayed. Core challenges include leveraging population-level structure to reduce per-user data needs, maintaining long-term user models under privacy constraints, and ensuring personalization does not erode generalization or safety.

System-Level Efficiency across Deployments. Most agents are still improved in isolation, causing interaction and supervision costs to scale linearly with the number of deployment instances [Ma *et al.*, 2024]. A promising direction is to treat data efficiency as a system objective: transform trajectories, skills, and verification outcomes into structured representations that can be selectively shared and reused across instances. Achieving this requires principled selection and aggregation, safeguards against error amplification, and evaluation protocols that measure marginal utility of shared experience rather than isolated task performance.

Self-Evolving Agents. A long-term goal is open-ended agents that continually improve through interaction with environments, humans, and data [Yao *et al.*, 2022], blurring the boundary between training-time and test-time learning. In deployment, however, unconstrained self-evolution is unrealistic because interaction, verification, and computation are budgeted. The challenge is to make self-evolution deliberate and economical: agents should decide when to explore, when to verify, and what to retain or reuse so improvement is sustainable and does not regress under drift.

Interaction-Centric Evaluation. Most benchmarks emphasize final task success and implicitly treat interaction as free, which hides inefficiencies in learning and adaptation. More informative evaluation should report not only success rates, but also interaction steps, tool calls, verification frequency, supervision cost, and performance gain per unit interaction [Shinn *et al.*, 2023]. Such protocols would enable principled comparison of methods that trade computation, verification, and interaction differently.

High-Stakes, Data-Scarce Domains. Interaction data is costly, sparse, or high-risk rather than simply "limited" in many applications. This is evident in embodied decision making [Qin *et al.*, 2024], scientific discovery [Swanson *et al.*, 2025], and medical decision support, where unsafe or excessive trial-and-error is unacceptable. These domains call for data-efficient agents that integrate domain priors, rely on verifiable signals, and allocate scarce human supervision where it has the highest leverage.

6 Conclusion

This survey presented an agent-centric view of data-efficient agentic learning, focusing on how to obtain and improve LLM-based agents when supervision and real-world interactions are scarce, expensive, or risky. We framed the design space along three complementary dimensions: experience augmentation, agent structural design, and learning paradigms, which together aim to maximize information yield per unit cost, often trading additional inference-time computation and verification for fewer external interactions and less human supervision. We also summarized representative benchmarks across Web, GUI, embodied, medical, and scientific domains, and discussed open challenges in long-horizon learning under tight budgets, generalization beyond data coverage, user-centric personalization under sparse feedback, and interaction-centric evaluation. We hope this survey helps clarify the emerging landscape and supports the development of robust and deployable data-efficient agents.

References

[Chen *et al.*, 2024] Canyu Chen, Jian Yu, Shan Chen, et al. Clinicalbench: Can LLMs beat traditional ML models in clinical prediction? In *AAAI Workshop*, 2024.

[Deng *et al.*, 2023] Xiang Deng, Yu Gu, Boyuan Zheng, et al. Mind2Web: Towards a generalist agent for the web. In *NeurIPS*, 2023.

[Driess *et al.*, 2023] Danny Driess, Fei Xia, Mehdi SM Sajjadi, et al. PaLM-E: An embodied multimodal language model. In *ICML*, 2023.

[Fallahpour *et al.*, 2025] Adibvafa Fallahpour, Jun Ma, Alif Munim, et al. MedRAX: Medical reasoning agent for chest X-ray. In *ICML*, 2025.

[Feng *et al.*, 2024] Peiyuan Feng, Yichen He, Guanhua Huang, et al. AGILE: A novel reinforcement learning framework of LLM agents. In *NeurIPS*, 2024.

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

[Gou *et al.*, 2024] Zhibin Gou, Zhihong Shao, Yeyun Gong, et al. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *ICLR*, 2024.

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

[Guo *et al.*, 2024] Taicheng Guo, Xiuying Chen, Yaqi Wang, et al. Large language model based multi-agents: A survey of progress and challenges. In *IJCAI*, 2024.

[Gupta *et al.*, 2020] Abhishek Gupta, Vikash Kumar, Corey Lynch, et al. Relay policy learning: Solving long-Horizon tasks via imitation and reinforcement learning. In *CoRL*, 2020.

[Hao *et al.*, 2023] Shibo Hao, Yi Gu, Haodi Ma, et al. Reasoning with language model is planning with world model. In *EMNLP*, 2023.

[He *et al.*, 2024] Hongliang He, Wenlin Yao, Kaixin Ma, et al. WebVoyager: Building an end-to-end web agent with large multimodal models. In *ACL*, 2024.

[Hu *et al.*, 2024] Mengkang Hu, Yao Mu, Xinmiao Yu, et al. Tree-Planner: Efficient close-loop task planning with large language models. In *ICLR*, 2024.

[Hu *et al.*, 2025] Mengkang Hu, Pu Zhao, Can Xu, et al. AgentGen: Enhancing planning abilities for large language model based agent via environment and task generation. In *KDD*, 2025.

[Huang *et al.*, 2024] Yidong Huang, Jacob Sansom, Ziqiao Ma, et al. DriVLMe: Enhancing LLM-based autonomous driving agents with embodied and social experiences. In *IROS*, 2024.

[Jansen *et al.*, 2024] Peter Jansen, Marc-Alexandre Côté, Tushar Khot, et al. DiscoveryWorld: A virtual environment for developing and evaluating automated scientific discovery agents. In *NeurIPS*, 2024.

[Jiang *et al.*, 2025a] Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, et al. KG-Agent: An efficient autonomous agent framework for complex reasoning over knowledge graph. In *ACL*, 2025.

[Jiang *et al.*, 2025b] Yixing Jiang, Kameron C Black, Gloria Geng, et al. MedAgentBench: A virtual EHR environment to benchmark medical LLM agents. *NEJMAI*, 2025.

[Kang *et al.*, 2025] Minki Kang, Jongwon Jeong, Seanie Lee, et al. Distilling LLM agent into small models with retrieval and code tools. In *NeurIPS*, 2025.

[Kannan *et al.*, 2024] Shyam Sundar Kannan, Vishnunandan LN Venkatesh, and Byung-Cheol Min. Smart-LLM: Smart multi-agent robot task planning using large language models. In *IROS*, 2024.

[Khattab *et al.*, 2023] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, et al. DSPy: Compiling declarative language model calls into self-improving pipelines. In *NeurIPS*, 2023.

[Laurent *et al.*, 2024] Jon M Laurent, Joseph D Janizek, Michael Ruzo, et al. LAB-Bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362*, 2024.

[Li *et al.*, 2025a] Bo Li, Yuanhan Zhang, Dong Guo, et al. LLaVA-OneVision: Easy visual task transfer. *TMLR*, 2025.

[Li *et al.*, 2025b] Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, et al. ScreenSpot-Pro: GUI grounding for professional high-resolution computer use. In *MM*, 2025.

[Li *et al.*, 2025c] 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, et al. Encouraging divergent thinking in large language models through multi-agent debate. In *EMNLP*, 2024.

[Lin *et al.*, 2022] Bingqian Lin, Yi Zhu, Zicong Chen, et al. ADAPT: Vision-language navigation with modality-aligned action prompts. In *CVPR*, 2022.

[Lin *et al.*, 2025] Bingqian Lin, Yunshuang Nie, Ziming Wei, et al. NavCoT: Boosting LLM-based vision-and-language navigation via learning disentangled reasoning. *TPAMI*, 2025.

[Liu *et al.*, 2023] Xiao Liu, Hanyu Lai, Hao Yu, et al. WebGLM: Towards an efficient web-enhanced question answering system with human preferences. In *KDD*, 2023.

[Liu *et al.*, 2025a] Bang Liu, Xinfeng Li, Jiayi Zhang, 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, et al. A survey on the feedback mechanism of LLM-based AI agents. In *IJCAI*, 2025.

[Luo *et al.*, 2025] Run Luo, Lu Wang, Wanwei He, et al. 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, et al. Coevolving with the other you: Fine-tuning LLM with sequential cooperative multi-agent reinforcement learning. In *NeurIPS*, 2024.

[Madaan *et al.*, 2023] Aman Madaan, Niket Tandon, Prakhar Gupta, et al. Self-Refine: Iterative refinement with self-feedback. In *NeurIPS*, 2023.

[McLean *et al.*, 2025] Reginald McLean, Evangelos Chatzaroulas, Luc McCutcheon, et al. Meta-World+: An improved, standardized, RL benchmark. In *NeurIPS*, 2025.

[Miao *et al.*, 2024] Ning Miao, Yee Whye Teh, and Tom Rainforth. SelfCheck: Using LLMs to zero-shot check their own step-by-step reasoning. In *ICLR*, 2024.

[Nakano *et al.*, 2021] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

[Nie *et al.*, 2025] Hongyi Nie, Yaqing Wang, Mingyang Zhou, et al. Adaptive preference arithmetic: Modeling dynamic preference strengths for LLM agent personalization. In *NeurIPS*, 2025.

[Ouyang *et al.*, 2022] Long Ouyang, Jeffrey Wu, Xu Jiang, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[Pan *et al.*, 2024] Jiayi Pan, Xingyao Wang, Graham Neubig, et al. Training software engineering agents and verifiers with SWE-Gym. In *ICML*, 2024.

[Park *et al.*, 2025] Chanwoo Park, Seungju Han, Xingzhi Guo, et al. MAPoRL: Multi-agent post-co-training for collaborative large language models with reinforcement learning. In *ACL*, 2025.

[Qi *et al.*, 2025] Zehan Qi, Xiao Liu, Iat Long Iong, et al. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *ICLR*, 2025.

[Qiao *et al.*, 2024a] Shuofei Qiao, Runnan Fang, Ningyu Zhang, et al. Agent planning with world knowledge model. In *NeurIPS*, 2024.

[Qiao *et al.*, 2024b] Shuofei Qiao, Ningyu Zhang, Runnan Fang, et al. AutoAct: Automatic agent learning from scratch for QA via self-planning. In *ACL*, 2024.

[Qin *et al.*, 2024] Yiran Qin, Enshen Zhou, Qichang Liu, et al. Mp5: A multi-modal open-ended embodied system in minecraft via active perception. In *CVPR*, 2024.

[Rawles *et al.*, 2025] Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, et al. AndroidWorld: A dynamic benchmarking environment for autonomous agents. In *ICLR*, 2025.

[Rein *et al.*, 2024] David Rein, Betty Li Hou, Asa Cooper Stickland, et al. GPQA: A graduate-level Google-proof Q&A benchmark. In *COLM*, 2024.

[Sang *et al.*, 2025] Jitao Sang, Jinlin Xiao, Jiarun Han, et al. Beyond pipelines: A survey of the paradigm shift toward model-native agentic AI. *J. ACM*, 2025.

[Shinn *et al.*, 2023] Noah Shinn, Federico Cassano, Ashwin Gopinath, et al. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

[Shridhar *et al.*, 2021] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, et al. ALFWorld: Aligning text and embodied environments for interactive learning. In *ICLR*, 2021.

[Song *et al.*, 2023] Chan Hee Song, Jiaman Wu, Clayton Washington, et al. LLM-Planner: Few-shot grounded planning for embodied agents with large language models. In *ICCV*, 2023.

[Sun *et al.*, 2024] Simeng Sun, Yang Liu, Shuohang Wang, et al. PEARL: Prompting large language models to plan and execute actions over long documents. In *EACL*, 2024.

[Swanson *et al.*, 2025] Kyle Swanson, Wesley Wu, Nash L Bulaong, et al. The virtual lab of AI agents designs new SARS-CoV-2 nanobodies. *Nature*, 646(8085):716–723, 2025.

[Tang *et al.*, 2025] Shuo Tang, Xianghe Pang, Zexi Liu, et al. Synthesizing post-training data for LLMs through multi-agent simulation. In *ACL*, 2025.

[Torreno *et al.*, 2017] Alejandro Torreno, Eva Onaindia, et al. Cooperative multi-agent planning: A survey. *CSUR*, 2017.

[Verma *et al.*, 2025] Gaurav Verma, Rachneet Kaur, Nishan Srishankar, et al. AdaptAgent: Adapting multimodal web agents with few-shot learning from human demonstrations. In *ACL*, 2025.

[Wang *et al.*, 2020] Yaqing Wang, Quanming Yao, James T Kwok, et al. Generalizing from a few examples: A survey on few-shot learning. *CSUR*, 53(3):1–34, 2020.

[Wang *et al.*, 2023] Lei Wang, Wanyu Xu, Yihuai Lan, et al. Plan-and-Solve Prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *ACL*, 2023.

[Wang *et al.*, 2024a] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, et al. Voyager: An open-ended embodied agent with large language models. *TMLR*, 2024.

[Wang *et al.*, 2024b] Lei Wang, Chen Ma, Xueyang Feng, et al. A survey on large language model based autonomous agents. *Front. Comput. Sci.*, 18(6):186345, 2024.

[Wang *et al.*, 2024c] Zihao Wang, Shaofei Cai, Anji Liu, et al. JARVIS-1: Open-world multi-task agents with memory-augmented multimodal language models. *TPAMI*, 2024.

[Wang *et al.*, 2025] Shijie Wang, Wenqi Fan, Yue Feng, et al. Knowledge graph retrieval-augmented generation for LLM-based recommendation. In *ACL*, 2025.

[Wei *et al.*, 2024] Yuxi Wei, Zi Wang, Yifan Lu, et al. Editable scene simulation for autonomous driving via collaborative LLM-agents. In *CVPR*, 2024.

[Wen *et al.*, 2024] Muning Wen, Ziyu Wan, Jun Wang, et al. Reinforcing LLM agents via policy optimization with action decomposition. In *NeurIPS*, 2024.

[Wu *et al.*, 2025a] Di Wu, Xian Wei, Guang Chen, et al. Generative multi-agent collaboration in embodied AI: A systematic review. In *IJCAI*, 2025.

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

[Xie *et al.*, 2024] Tianbao Xie, Danyang Zhang, Jixuan Chen, et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *NeurIPS*, 2024.

[Yang *et al.*, 2025] Zeyuan Yang, Delin Chen, Xueyang Yu, et al. VCA: Video curious agent for long video understanding. In *ICCV*, 2025.

[Yao *et al.*, 2022] Shunyu Yao, Jeffrey Zhao, Dian Yu, et al. ReAct: Synergizing reasoning and acting in language models. In *ICLR*, 2022.

[Yu, 2018] Yang Yu. Towards sample efficient reinforcement learning. In *IJCAI*, pages 5739–5743, 2018.

[Zhang *et al.*, 2025a] Chaoyun Zhang, Liqun Li, Shilin He, et al. UFO: A UI-focused agent for Windows OS interaction. In *NAACL*, 2025.

[Zhang *et al.*, 2025b] Zeyu Zhang, Quanyu Dai, Xiaohe Bo, et al. A survey on the memory mechanism of large language model-based agents. *TOIS*, 43(6):1–47, 2025.

[Zhao *et al.*, 2024] Andrew Zhao, Daniel Huang, Quentin Xu, et al. ExpeL: LLM agents are experiential learners. In *AAAI*, 2024.

[Zhou *et al.*, 2024a] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, et al. Language agent tree search unifies reasoning, acting, and planning in language models. In *ICML*, 2024.

[Zhou *et al.*, 2024b] Gengze Zhou, Yicong Hong, Zun Wang, et al. 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, et al. WebArena: A realistic web environment for building autonomous agents. In *ICLR*, 2024.