Scaling LLM Agent Learning with Data Synthesis: A Comprehensive Survey

Hanrong Zhang[1,*], Yankai Chen[2,4,*,†], Shicheng Fan[1,‡], Dehai Min[1,‡], Shaowen Chen[3,‡], Huanhuan Ma[1,‡], Zhaofen Wu[3,‡], Jie Yang[1,‡], Bowei He[2,4,‡], Jikun Kang[4,‡], Kening Zheng[1,‡], Xi Chen[4,‡], Chunyu Miao[1,‡], Fulin Lin[3,‡], Wei-Chieh Huang[1,‡], Jiayu Zhou[5,‡], Haolun Wu[4], Liancheng Fang[1], Hong Kang[4], Langzhou He[1], Henry Peng Zou[1], Chengze Li[1], Jialong Wu[6], Haiwen Hong[3], Zhaorun Chen[7], Hanjun Luo[8], Linghe Kong[9], Hongwei Wang[3], Dawn Song[10], Philip S. Yu[1], Xue Liu[2,4]

[1]University of Illinois Chicago [2]MBZUAI [3]Zhejiang University [4]McGill University
[5]Columbia University [6]Peking University [7]University of Chicago
[8]New York University [9]Shanghai Jiao Tong University [10]University of California, Berkeley

Abstract

LLM agents are transforming language models from chatbots into interactive systems that operate through memory, tool use, and external environments. Training such agents requires coherent goals, multi-step trajectories, feedback, and outcomes derived from long-horizon interactions, yet these data are difficult to collect at scale because environments are costly, human oversight is limited, and execution traces are complex. Therefore, data synthesis offers a practical route to scaling agent learning by sampling supervision from source distributions such as teacher models, seed-conditioned generators, simulators, judges, or executable environments. Despite growing interest, no survey has systematically organized synthetic data for LLM agents. To bridge this gap, we present a comprehensive survey that classifies synthesis methods for LLM agents, by their outputs within the agent–environment loop, including task specifications, agent trajectories, feedback signals, and environments. We examine the lifecycle of these artifacts, from grounding and quality control to their utilization in learning, evaluation, and downstream applications, and identify key challenges for reliable and scalable agent-data synthesis.

1 Introduction

LLM agents are increasingly deployed to tackled complex and real-world tasks. By integrating planning, tool use, and memory with interactive environments, these systems can sustain the long-horizon engineering required for autonomous execution [143, 211, 300, 338]. Recent advancements have proved their effectiveness across diverse domains, spanning web navigation and coding to deep research and specialized medical or educational assistance [63, 181, 193, 226, 251, 295]. Crucially, this evolution represents a fundamental shift from passive question answering to active execution. By acting as autonomous interfaces, these agents can seamlessly manipulate external systems, transforming how we automate complex, end-to-end workflows.

Since agent tasks are often long-horizon and interactive, training such agents requires supervision that goes beyond isolated input–output pairs to capture the entire process of multi-step executions [17, 31, 215, 303, 313]. **Yet, gathering real-world interaction data is difficult**: recording live user logs is expensive, web and software environments change constantly, and critical failure cases may not be safely tested in production [205, 268, 301]. Therefore, the main challenge is not just checking the final outcome is right, but obtaining coherent and verifiable interaction traces. Moreover, recent studies show that an agent's capability is directly driven by the quality and variety of this interaction

*Hanrong Zhang (hzhan135@uic.edu) and Yankai Chen (yankaichen@acm.org) are co-first authors.
†Yankai Chen is the corresponding author.
‡Authors marked with ‡ are core contributors.

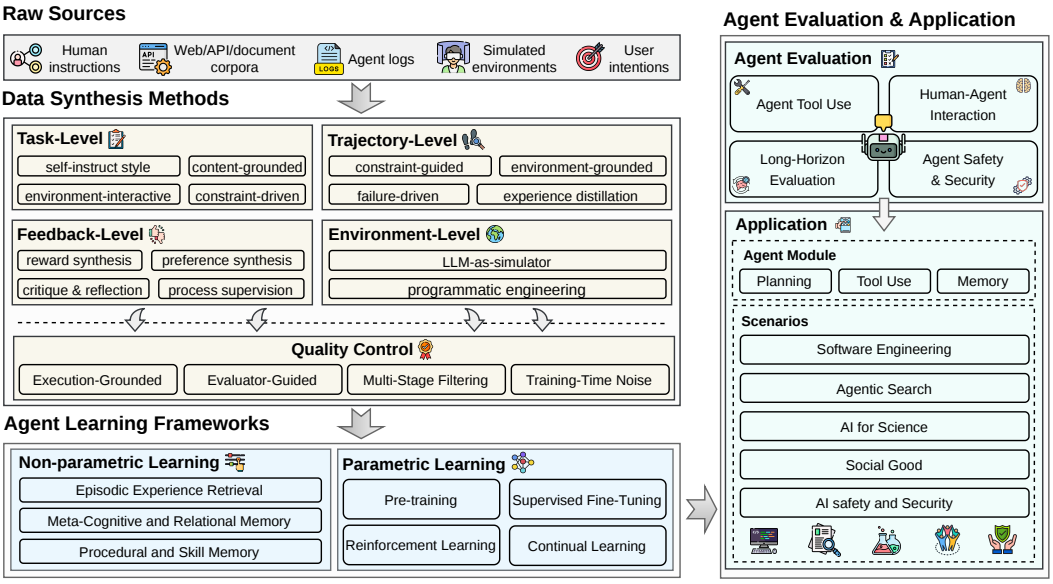

Figure 1: Overview of data synthesis for LLM agent learning, spanning data sources, synthetic artifacts, quality control, learning paradigms, evaluation, and downstream applications.

data, such that the abundant, diverse, and verifiable one is a key character to improve capability. This suggests that, scaling agent learning requires more than increasing model size: **agents should learn from massive, high-quality, and diverse interaction experiences** that expose them to various scenarios, environments, and unexpected failure modes [32, 183, 296, 345].

These challenges have motivated the use of **Agent Data Synthesis** as a scalable alternative to costly real-world data collection. Unlike traditional synthesis methods that relied on rigid templates, grammars, rules, or domain-specific simulators, recent LLM-driven approaches can simluate the interactive ecosystem, including autonomously generating complex tasks, evolving execution trajectories, and providing evaluative feedback, etc. [162, 165, 188, 255, 288, 293]. Viewed more generally, this multi-dimensional data synthesis functions similarly to a generalized form of knowledge distillation, where the agent samples its entire training experience from an underlying *source distribution* [77, 80]. Since agent tasks are inherently dynamic and require multi-step execution, this synthesis must continue to scale up data volume and scenario diversity. However, a critical issue is that as interactions extend over long horizons, ensuring the reliability of the data becomes much harder. The central challenge is thus two-fold: we need to expand the data scale, while ensuring that both the generative source and its internal verifiers remain dependable enough to guide complex target behaviors.

While recent surveys comprehensively cover LLM agent architectures, evaluation, and tool learning [93, 129, 147, 196, 247, 302], a dedicated survey focusing on *data synthesis for agent learning* remains absent. To address this, this survey provides a comprehensive and structured overview of data synthesis for LLM agent learning from an artifact-centered perspective. We organize existing methods around four core artifacts in the agent–environment loop—*task specifications*, *agent trajectories*, *feedback signals*, and *interactive environments*, which naturally capture varying levels of signal granularity and informational density required for agent training. Based on these artifacts, we then examine quality-control mechanisms in §3 and investigate how curated artifacts are applied to agent learning in §4. We further survey synthetic evaluation data in §5 and showcase downstream applications in §6. Finally, we outline open risks and future directions for reliable and scalable data synthesis in §7. Fig. 1 illustrates the overall framework of this survey.

2 Data Synthesis Methods

We organize the literature into four levels based on the artifact being synthesized: task-level (§2.1), trajectory-level (§2.2), feedback-level (§2.3), and environment-level (§2.4). Specifically, task synthesis samples problem specifications; trajectory synthesis samples conditional behaviors; feedback

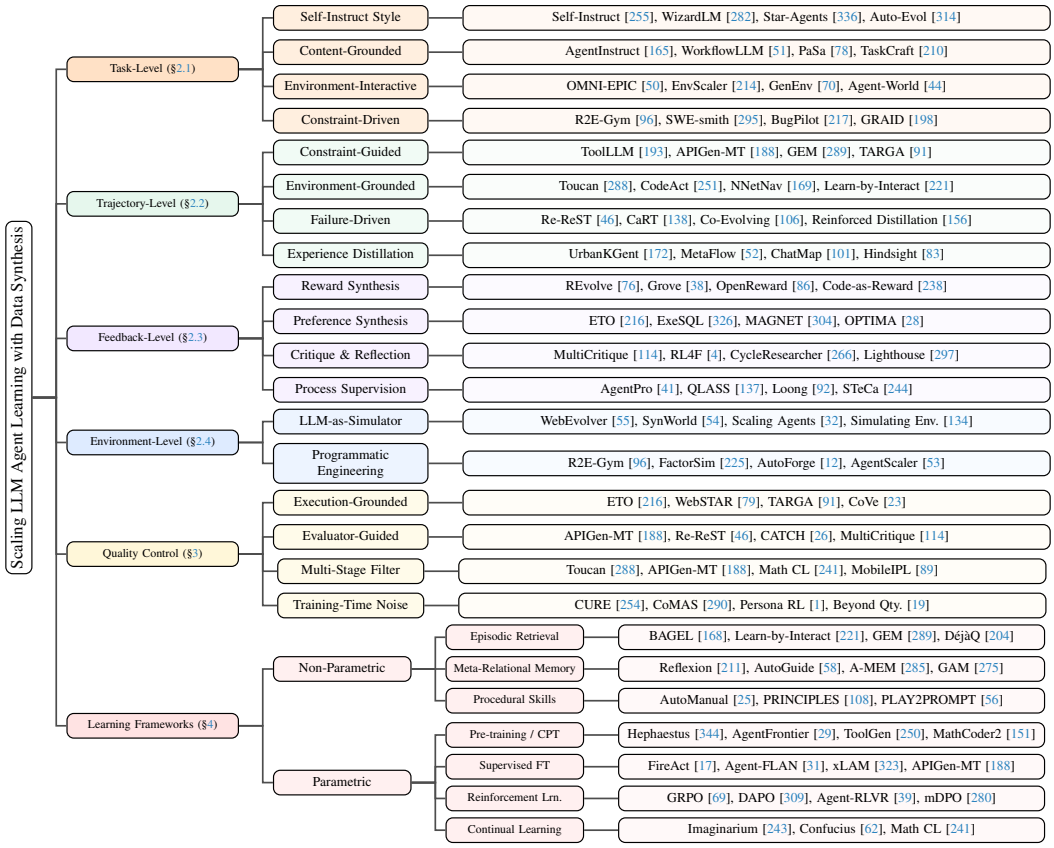

Figure 2: Taxonomy of data synthesis for LLM agent learning, covering synthesis methods (§2.1–§2.4), quality control (§3), and learning frameworks (§4).

synthesis samples evaluative signals; and environment synthesis constructs the substrate from which tasks, transitions, and rewards emerge. Rather than being isolated, these levels are hierarchically nested, where each higher level subsumes the lower ones to provide an increasingly dense and interactive stream of supervision. For instance, a trajectory requires a task, while a fully simulated environment can dynamically spawn tasks, trajectories, and feedback alike. To maintain conceptual clarity amid this nesting, we classify each work by the deepest and most complex level at which it performs data synthesis for agents. The taxonomy can be found in Fig. 2.

2.1 Task-Level Specification Synthesis

Task-level specification synthesis defines the problem an agent must solve. A task typically includes a goal, inputs, constraints, tool schemas, and, in some cases, a target answer or verifier. We focus on methods that synthesize task specifications, categorizing them according to their sources: seed prompts, grounded artifacts, live environments, and explicit constraints.

Self-Instruct-Style Synthesis Self-instruct-style methods scale up a small set of human-written seed instructions by prompting a language model to generate new ones in the same format. For instance, Self-Instruct [255] generates new instruction triples from human-written seeds and removes near-duplicates. Later work improves difficulty and teacher diversity through mutation prompts [282, 314], learned prompt rewriting [203], voting [118], role-specialized prompting [336], or reviewer-adjudicator designs [64]. These pipelines scale easily, but they remain weakly grounded, where the generated tasks may not reflect the actual distribution of instructions users issue in practice, in terms of topic, phrasing, or difficulty.

Content-Grounded Synthesis Rather than generating tasks from model-written seeds, content-grounded methods derive task specifications from real-world artifacts, e.g., documents, code fragments, API schemas, and similar resources harvested from existing corpora. Two common patterns emerge. (1) *Single-artifact grounding*: each task is constructed from a single harvested item, such as a document span, code fragment, API field, Python script, or Apple Shortcut [51, 131]. For example, AgentInstruct [165] transforms raw documents or code through a multi-agent pipeline. It first converting seeds into intermediate representations, then generating and iteratively refining tasks from them. DroidCall [279] and Hammer [136] instead ground tasks directly in tool schemas: the former generates natural language instructions paired with Android intent calls from predefined function definitions, while the latter masks function names during synthesis to force models to reason from parameter descriptions rather than memorized identifiers. (2) *Compositional-artifact grounding* scales task complexity by linking multiple artifacts rather than grounding each task in a single item. One line of work builds tasks from knowledge graphs: by sampling subgraphs of increasing traversal depth and deliberately obfuscating entity information, these methods generate tasks that require models to follow multi-hop evidence chains rather than recall memorized facts [34, 60, 78, 98, 122, 123, 141, 179, 324, 341]. Another line composes tasks from multi-step tool workflows: WorkflowLLM [51] collects real Apple Shortcuts and expands them into new task queries, while TaskCraft [210] starts from verified atomic tool tasks and combines them through depth- and width-based extensions to produce hierarchically complex challenges [191, 218].

Environment-Interactive Synthesis Unlike content-grounded methods that rely on static artifacts, this category synthesizes task specifications inside simulators or sandboxes, using execution outcomes as a verifier rather than relying on the artifact itself to guarantee task validity. A first wave of work instantiates this idea in specific domains: OMNI-EPIC [50], EnterpriseBench [240], and InvestAlign [246] each embed task generation inside a runnable environment that span generated game levels, enterprise access-control systems, and investment simulators, respectively. Then the following work EnvScaler [214], ScaleEnv [235], and AWM [258] further support the large-scale programmatic synthesis of environments and tool interfaces. Evolving beyond this, GenEnv [70], Agent-World [44] shift toward execution-driven reverse-engineering, synthesizing queries and rubrics directly from randomly executed tool trajectories. While offering stronger grounding than static artifacts, these methods inherently depend on domain-specific executable infrastructure that is costly to build and difficult to transfer across settings.

Constraint-Driven Synthesis Rather than sourcing tasks from seeds or environments, constraint-driven methods synthesize tasks by first defining an explicit, checkable property and then generating instances designed to satisfy or violate it. Specifically, Controlled-bug methods instantiate this by introducing code faults through mutations, commit reversals, or patch composition, using test execution to confirm that a bug exists and is meaningful [96, 217, 295]. Policy-violation methods specify forbidden behaviors through policy documents, classifiers, or privacy norms, and generate targets in which an agent must navigate or detect those violations [111–113, 139, 158, 198]. Domain-validator methods outsource verification to external tools that certify synthesized tasks against domain-specific correctness criteria, such as theorem provers, physics simulators, or DFT scores [85, 186, 192, 218]. The tradeoff is clear: every new target property requires a corresponding verifier, making these methods precise but also tends to limit the generality of these methods across domains.

2.2 Trajectory-Level Data Synthesis

While task-level synthesis defines what an agent should do, training agents also requires supervision over *how* they do it. A trajectory records the full sequence of an agent's actions, including reasoning steps, tool calls, and environment interactions, that leads to a final outcome. Thus, synthesizing such data is the focus of this section.

Constraint-Guided Reasoning Synthesis A central challenge in trajectory synthesis is ensuring that intermediate steps are coherent and grounded, rather than plausible-sounding but unverifiable. This category addresses it by anchoring each step to an external signal that constrains how the trajectory unfolds, rather than relying on free-form Chain-of-Thought [261]. Search-based methods enumerate candidate paths and retain those passing execution checks, utilizing DFS to explore APIs call sequences [193] or MCTS to navigate chemistry reaction spaces [271]. Outcome-anchored approaches work in the opposite direction: starting from a valid endpoint or executed trajectory,

they retroactively infer the reasoning path that could have produced it [125, 221]. Structural methods constrain trajectory shapes upfront using predefined standard operating procedures or task blueprints [26, 91, 135, 188]; GEM [289] extends this by mining such workflows directly from large text corpora rather than hand-authoring them. When execution feedback is unavailable, model-based verifiers supply proxy signals [226, 330], though at the cost of high path-exploration overhead and verifier bias.

Environment-Grounded Synthesis Constraint-guided methods produce trajectories that are internally coherent, but they do not necessarily reflect how an agent would behave in a real environment. Environment-grounded methods close this gap by using actual execution outcomes, e.g., success flags, error messages, and state transitions, as the primary supervision signal. Execute-and-filter methods run candidate actions and keep successful trajectories across Model Context Protocol (MCP) servers, data-science tasks, user-agent-tool loops, or Python-as-action formats [251, 293, 329]. Toucan [288] scales this paradigm across MCP servers with rule- and model-based admission checks, while FireFly [150] shows how real APIs can serve as grounded environments for large-scale verified tool-call data generation. Explore-then-summarize methods take a complementary approach: they first find successful action sequences through environment interaction, then retroactively name the tasks those sequences solve. For instance, NNetNav [169] prunes paths without sub-task annotations, and Learn-by-Interact [221] applies the same idea to interaction logs. Execution provides strong supervision, but only when exploration manages to reach useful states.

Failure-Driven Synthesis Both constraint-guided and environment-grounded methods largely discard failed rollouts, retaining only trajectories that succeed. Failure-driven synthesis takes the opposite stance: failures carry information about where agents go wrong, and retaining them enables models to learn recovery strategies and discriminative behaviors. Specifically, repair-based methods generate failure-reflection-correction triples by having a teacher model identify early-stage errors and rewrite the trajectory into a successful one, or by perturbing failed rollouts into corrected alternatives [110, 156]. Negative-construction methods take a different angle: rather than repairing failures, they preserve them as contrastive or preference supervision, exposing models to plausible-but-wrong behavior so they learn to discriminate [106, 148]. CaRT [138] applies this idea specifically to termination decisions, constructing minimal counterfactuals about when an agent should stop. For these methods, failure-derived supervision offers a signal that success-only approaches cannot, that it can teach agents not just what to do, but how to recover when things go wrong. However, the quality of this supervision is bounded by what the generator or teacher can produce: if failures are too random or too rare, the resulting signal offers little meaningful learning.

Experience Distillation and Rewriting The preceding methods all synthesize trajectories from scratch or from live interactions. Experience distillation takes a different angle: rather than generating new trajectories, it transforms existing logs, rollouts, or task traces into more reusable training artifacts. One approach compiles experience into structured programs or templates that can be retrieved and adapted to new tasks. For example, UrbanKGent builds a reusable geospatial instruction set from interaction logs, and MetaFlow extracts retrievable task templates from past rollouts [52, 172]. Another approach distills experience into natural language, capturing behavioral patterns that fixed schemas often fail to represent: hindsight relabeling mines failed episodes for corrective signal, ChatMap distills reusable service strategies from customer-service dialogues, and AgentTrek reconstructs web trajectories by replaying steps described in online tutorials [83, 101, 287]. Structured representations are easier to retrieve and reuse, while natural-language forms are more flexible but depend more heavily on model judgment to extract meaningful patterns.

2.3 Feedback-Level Data Synthesis

Trajectory-level synthesis (§2.2) builds the behavior agents train on, while feedback-level synthesis builds the *evaluative signals* used to judge that behavior. These signals, including scalar rewards, pairwise preferences, natural-language critiques, and step-level process labels, are materialized as offline feedback artifacts that support reward modeling, preference optimization, critic training, or feedback-based adaptation, rather than serving as transient trajectory filters. Some methods use rollouts, search, or interaction during data construction; we include them here only when these procedures produce reusable feedback artifacts rather than serving solely as online rewards during policy optimization.

Reward Synthesis Reward synthesis transforms sparse outcome signals into reusable reward artifacts over collected trajectories, including scalar labels, reward models, and executable reward functions. One line of work annotates offline trajectories through consistency checks or hindsight labels, while others train tool-augmented reward models from synthesized judgments [86, 119, 187]. A second line synthesizes executable reward functions as reusable artifacts, validating or refining them through automated code verification, human-guided evolutionary search, or VLM-based feedback [38, 76, 238]. A third family derives reward signals from external priors or task-specific consistency checks, such as game manuals or persona-consistency constraints [1, 274]. Across all three, the central risk is grounding: a dense reward helps only if it tracks the real task rather than artifacts of the annotator, verifier, or external prior.

Preference Synthesis Because scalar rewards require absolute calibration, which foundation-model annotators often struggle to provide, preference-based methods synthesize relative judgments, comparing which of two behaviors is better without requiring calibrated scores. Execution provides one of the most reliable sources of preference pairs: ETO compares successful and failed trajectories directly, and ExeSQL verifies text-to-SQL queries against execution results [216, 326]. When execution is unavailable or only partially informative, model judges or verifiers construct preference pairs from agent rollouts. MAGNET [304] models multi-turn function call interactions as a graph, traversing signature paths to generate positive and negative trajectory pairs for preference optimization. Methods like DecEx [120] and others construct pairs from inference trees or step-level samples, using model-based judgment to rank candidate behaviors [27, 126, 281]. Multi-agent settings open up a further source of preference signal: Optima [28] treats conversation turns as tree nodes and uses MCTS to explore diverse interaction paths, while other methods derive preferences from cross-agent debates, self-play, user edits, or communication games [57, 59, 220, 241, 339, 343]. To summarize, preferences are easier to elicit than calibrated scores, but they discard magnitude information and inherit biases from their underlying source.

Critique and Reflection Preferences indicate which behavior is better but not why. Therefore, critique synthesis metods fill this gap by producing natural-language feedback artifacts, such as error descriptions, revision advice, or critique-revision pairs [132]. MultiCritique [114] aggregates flaw-specific critiques from multiple LLMs and retains only those that improve downstream revision, while RL4F [4] trains a small critique generator to maximize downstream repair by a large frozen model. Another line turns generator-evaluator loops into reusable critique-revision traces: the evaluator identifies errors or weak decisions, and the generator revises its next attempt using that feedback, producing paired artifacts that capture both the failure mode and the corrective response [82, 266, 297]. In general, critique synthesis produces the interpretable form of feedback among the three categories, making it particularly useful for tasks where understanding failure modes matters as much as correcting them. However, natural-language feedback is harder to verify than scalar signals and prone to inheriting the evaluator's blind spots.

Process Supervision Preceding three forms primarily operate at the trajectory level, requiring the agent to infer which intermediate steps contributed to success or failure. Process supervision addresses this credit-assignment problem by attaching feedback directly to individual reasoning or action steps. Specifically, tree-search and value-estimation methods generate step labels by propagating outcome signals backward through exploration trees. For instance, QLASS [137] builds an exploration tree from agent rollouts and estimates Q-values at each node, producing step-level annotations that reflect long-term utility rather than immediate outcomes. Other methods in this group assign step labels from one-step lookahead or transition statistics [41, 133, 161]. A second group derives step-level feedback without tree search: STeCa [244] identifies suboptimal actions through step-level reward comparison during exploration and uses LLM-driven reflection to construct calibrated replacement trajectories. Other methods in this group derive step signals from reflection-based repair, paired code-and-test generation, or independent verification checks [92, 254, 310, 318]. A key frontier for agentic settings is exposing denser intermediate signals, e.g., state diffs, tool validity checks, subgoal progress, and recovery indicators, that make partial progress measurable before a final outcome is observed. Across all these approaches, step-level feedback is most valuable on long-horizon tasks, but reliable only when labels track genuine progress rather than surface correlates of success.

2.4 Environment-Level Data Synthesis

The previous levels synthesize tasks, trajectories, and feedback; environment-level synthesis builds the interactive substrate from which those artifacts can emerge. In this sense, an environment is not only a task container but also a source distribution and verifier: it generates reachable states, constrains possible actions, and returns feedback about whether an interaction changed the world in the intended way. Recent environment-scaling work similarly views environments as producers of experience through generation, execution, and feedback rather than passive containers for agent activity [93].

LLM-as-Simulator LLM-as-simulator methods use an LLM as a text-based environment that returns observations and rewards. This gives broad coverage because a single backbone can simulate websites, APIs, dialogue partners, or workflows. Frozen LLM simulators improve throughput but not fidelity [32, 134]. WebEvolver [55] raises fidelity by co-evolving a world-model LLM for training-time simulation and inference-time lookahead. Other work adds MCTS or tabular Q-learning over imagined transitions [14, 54]. However, the weakness is that the simulator is still bound by the LLM's world knowledge, so hallucinated dynamics can become reward hacking.

Programmatic Environment Engineering Programmatic environment engineering builds environments with code, so execution defines dynamics and reward. Software engineering is the cleanest case because codebases and tests already act as environments; R2E-Gym [71, 96] combines native test suites with synthesized checks. For tool-use agents, recent systems synthesize stateful mock services with executable tools, database-backed state, and verification code, extending environment scaling from abstract tool interfaces to executable approximations of real software systems [12, 53, 214, 258]. This line of work also extends beyond tool APIs to broader computer use environments. MobileGym provides programmable mobile simulators with reset, rollout forking, and state verification, while WebGym scales visual web training with realistic tasks, rubric-based evaluation, and efficient rollout collection [6, 269]. CLI-Anything complements these systems by exposing real software through command line harnesses with explicit state, backend execution, and programmatic feedback [298]. The same programmatic paradigm also appears in evaluation-oriented and industrial pipelines, which generate task specifications, tool interfaces, scoring rubrics, or training environments for large-scale agent tuning [130, 231, 233, 249]. Outside software and tool use, methods convert static resources or high-level specifications into interactive environments for domains such as embodied manipulation and procedural simulation [24, 40, 48, 107, 223, 312]; FactorSim [225] builds simulations from natural language with factored POMDPs, while AutoEnv [325] shows that more environments help less without an adaptive curriculum.

3 Quality Control for Synthetic Data

Building on the taxonomy in §2, quality control examines and controls synthesized tasks, trajectories, feedback signals, and environments before they are used for learning or evaluation. Its core issue is verifier reliability: executable checks provide precise signals when tests, constraints, tools, or state changes are available, while human and model judges cover less formal criteria but introduce calibration bias [20, 335]. We organize existing work into execution-grounded filtering, evaluator-guided scoring, multi-stage pipelines, and training-time noise handling.

Execution-Grounded Filtering Execution-grounded filtering uses tools, environments, tests, or task constraints as automatic checkers. A synthetic trajectory can then be kept, rejected, or relabeled according to whether execution reaches the expected state. The simplest form assigns only final success or failure labels [216]. More fine-grained methods inspect intermediate behavior, which is important for long trajectories because a single late error should not erase evidence that earlier steps were valid. DeepAnalyze verifies data-science outputs against executable constraints [329], WebSTAR filters GUI rollouts at the action level [79], TARGA validates structured reasoning steps [91], and CoVe folds explicit task constraints into both generation and verification [23]. The same principle also appears before and after trajectory generation: tool-oriented pipelines screen APIs or MCP environments before synthesis begins, while search-tree methods explore candidate paths and label branches by downstream success [27, 41, 120, 193, 288]. Execution provides the cleanest quality signal among these mechanisms, but only when the relevant notion of correctness

has already been made executable. Open-ended tasks remain difficult because success may depend on preferences, strategies, or partial progress that cannot yet be reduced to tests or state predicates.

Evaluator-Guided Scoring When execution cannot decide quality, evaluator-guided scoring uses a model or human judge to assess synthetic artifacts. The evaluator may assign an absolute score, compare candidates, or produce feedback that guides revision. Because a single judge can be unstable, many systems reduce variance through reviewer committees, ensembled judgments, or disagreement-aware critics [119, 188, 241]. Others evaluate the feedback itself, for example by checking whether a critique improves the downstream actor that uses it [114]. This mechanism can also make quality control corrective rather than purely selective: Re-ReST revises trajectories using environment feedback [46], while CATCH and APIGen-MT rewrite low-scoring samples according to quality scores and reviewer comments [26, 188]. Human review remains useful for criteria that model judges often miss, including SOP compliance and reward-function design [76, 135]. The central risk is correlated error: if the generator and evaluator share model families, training data, or assumptions, both may overlook the same defect. Rubrics reduce this risk only when they express the quality dimensions that users actually care about.

Multi-Stage Filtering Pipelines Since objective and subjective checks fail in different ways, stronger pipelines usually chain them. A typical pipeline first removes format errors or non-executable samples, then applies model or human review, and finally regenerates or reranks the remaining data. Toucan follows this structure for tool-use data by combining automatic checks on format and executability with model-based quality review [288]. APIGen-MT, Math CL, and MobileIPL adopt similar staged designs for multi-turn APIs, mathematical reasoning, and mobile-screen control [89, 188, 241]. These systems are best understood as data-engineering pipelines rather than single filtering algorithms. Their effectiveness depends less on the number of gates than on whether the gates cover complementary failure modes. A pipeline that stacks several model judges may still preserve the same blind spot, whereas a pipeline that combines schema checks, execution, rubric review, and targeted regeneration can expose errors at different points in the synthetic data lifecycle.

Training-Time Noise Handling The preceding mechanisms try to remove weak data before training. Training-time noise handling makes a different choice: it keeps imperfect data, but changes the learning signal so that weaker samples receive less influence. CURE uses co-evolved unit testers as reward signals [254], SPORT refines preference labels across iterations [126], and Persona RL and CoMAS use multi-turn reinforcement learning (RL) to reduce the effect of noisy interactions [1, 290]. A related direction controls quality through data composition. For code agents, trajectory diversity can matter more than raw volume, and plausible negative samples can be more useful than random wrong answers [19, 43]. This approach is attractive when neither the source distribution nor the verifier is clean enough to support hard filtering. It also links quality control directly to the learning frameworks in the next section: a sample that is too noisy for imitation may still be useful as a ranked comparison, a negative example, or a low-reward trajectory. Its main limitation is the instability of this tradeoff, since training objectives may absorb moderate synthetic noise in some settings but amplify repeated verifier errors in others.

4 Agent Learning Frameworks

After synthesis and quality control, synthetic artifacts become learning material. They may remain outside the model as retrievable experience, or they may be converted into gradients that update the model weights. This distinction shapes both the benefit and the risk of synthetic data. External memories are editable and can track changing tools or environments, but they depend on retrieval and reuse at inference time. Parametric training produces behavior that is available without retrieval, but errors can become embedded in the policy. We therefore organize agent learning into non-parametric and parametric frameworks.

4.1 Non-parametric Learning

Non-parametric learning treats synthesized experience as an external resource. The stored artifact can be close to the original interaction, as in an episodic trace, or it can be compressed into a lesson, relation, procedure, or skill. Moving from episodes to skills improves reuse across tasks, but it also

increases the burden on synthesis: the system must decide which details are incidental and which are generally useful.

Episodic Experience Retrieval Episodic experience retrieval stores solved interactions and reuses them as demonstrations for similar future tasks. This is the least abstract form of non-parametric learning: the synthetic artifact is still a concrete task trace, often including the instruction, actions, observations, and final outcome. Self-Generated In-Context Examples stores successful traces for few-shot retrieval [208], while Learn-by-Interact grounds its traces in documentation, tutorials, and executable interaction so that retrieved examples remain tied to the environment they came from [221]. Other systems improve the store by refining exploration into instruction-trace pairs, mining tool-use workflows from text, rewriting failures into successful cases, evolving verifiable problems, or indexing trajectories by skill [75, 168, 204, 264, 289]. This directness makes episodic retrieval easy to refresh without weight updates, but it also limits transfer: if a new task is phrased differently, requires a different tool composition, or departs from the stored state distribution, the retrieved episode may provide misleading guidance rather than useful supervision.

Meta-Cognitive and Relational Memory Meta-cognitive memory moves one level above episodic reuse by storing what the agent should learn from an interaction rather than the interaction itself. MetaReflection distills trial and error into reusable instructions [74], AutoGuide selects context-aware guidelines [58], and Reflexion stores verbal self-reflections after failures [211]. A-MEM links each new note to earlier memories, allowing the memory base to grow as a connected record rather than an unordered collection [285]. Relational memory adds temporal or causal structure through contextual replay, chronological timelines, or hierarchical graphs [144, 174, 275]. These abstractions are useful because agent tasks often recur in form even when their surface details change: a lesson about checking tool preconditions can transfer across many tools, whereas a stored trajectory may apply only to one API state. The risk is overgeneralization, since a synthesized reflection or relation is valuable only if it captures a stable pattern, not a detail that happened to matter in one episode.

Procedural and Skill Memory Procedural and skill memory compresses experience into assets that can be invoked directly during future interaction. AutoManual compiles environment rules into instruction manuals [25]. PRINCIPLES learns strategy memory from self-play dialogues, and PLAY2PROMPT converts tool play into prompting templates [56, 108]. CoEvoSkills adds a verifier to the synthesis loop, co-evolving the checker and the skill generator so that skill construction still receives feedback when no external test is available [321]. This is the most compact non-parametric use of synthetic data, turning many trajectories into a smaller procedural interface that is easier to inspect and edit than a large retrieval store. The same compression raises the cost of errors: a flawed procedure can be reused across many tasks, so procedural memory needs either strong verification during synthesis or later mechanisms that detect when an invoked skill is no longer appropriate.

4.2 Parametric Learning

Parametric learning uses synthetic artifacts as training signals that update the model itself. The relevant artifact changes across stages. Pre-training needs broad corpora that expose agent-like patterns. Supervised fine-tuning needs high-quality trajectories that can be imitated. Reinforcement learning needs rewards, preferences, or verifiers that score behavior. Continual learning needs a cycle that can synthesize new data as tasks, tools, and environments evolve. Because the learned behavior persists after training, parametric methods are more sensitive to systematic noise than external-memory methods.

Pre-training Pre-training applies self-supervised objectives to synthetic agent corpora before task-specific adaptation [162]. Agentic continual pre-training exposes the model to planning, tool-use, code, and instruction-style sequences so that later training starts from a model already biased toward interactive behavior [29, 36, 151, 203, 222, 344]. AgentFounder frames this stage as a bridge between general pre-training and fine-tuning [222], while Hephaestus-Forge and AgentFrontier study how agent, code, and text mixtures should be separated from later tool-augmented post-training [29, 344]. ToolGen takes a different route by assigning tools virtual tokens, making tool identity part of the model vocabulary [250]. The role of synthesis at this stage is to shape a broad prior rather than to teach one verified behavior. Scale and coverage therefore matter, but so does the mixture of agentic

and non-agentic text. The interaction between these broad synthetic priors and later supervised fine-tuning or reinforcement learning remains under-specified, especially when later stages use narrower but more strongly verified data.

Supervised Fine-Tuning Supervised fine-tuning (SFT) turns synthesized trajectories into imitation targets. The model observes states, reasoning traces, actions, tool outputs, and final answers, and learns to reproduce similar behavior on related tasks. Early agent-tuning work used diverse trajectories, corrective signals, and modular annotations to inject interactive behavior into the model weights [17, 31, 303, 313]. Recent work focuses more directly on the tension between trajectory quality and task coverage. MAGNET constructs function-call paths over dependency graphs and distills both positive and negative trajectories from a teacher [304]. APIGen-MT starts from verified task blueprints with ground-truth actions and then converts them into realistic multi-turn interactions [188]. Other pipelines expand coverage through multi-source training, execution-verified API traces, or workflows mined from text [142, 193, 289, 323]. SFT is therefore where the quality-control decisions from §3 become most visible: a useful trajectory should cover realistic tool use without teaching invalid calls or brittle reasoning. At scale, SFT often provides the initial tool-use competence that later reward-driven training refines [122, 231].

Reinforcement Learning Reinforcement learning changes the role of synthetic data from imitation target to scoring signal. The agent samples new behavior, and synthesized tasks, environments, rewards, or preferences determine which behavior is reinforced. Recent agent training often uses Group Relative Policy Optimization (GRPO) because it estimates relative advantages within a group of sampled responses without a separate value model [69], with later variants adjusting clipping, normalization, baseline estimation, or tree-structured rollouts [3, 100, 146, 309]. For this survey, the optimizer is secondary to the source of the reward. In code and tool-use settings, verifiable rewards are especially important because correctness can be checked by execution, unit tests, or environment-state comparisons [33, 39]. Data synthesis supplies the task distribution and runnable environments that make such rewards available; if the synthesized tasks are too narrow, RL discovers narrow behavior, and if the verifier is incomplete, RL may exploit it. Other settings use denser process reward models, outcome reward models, or preference comparisons [4, 61, 114, 115, 126, 154, 199, 248, 280, 318]. These signals connect back to feedback-level synthesis in §2.3: the value of RL depends on whether synthesized feedback tracks real task progress rather than artifacts of the checker or judge.

Continual Learning Continual learning closes the loop by repeating synthesis, filtering, and training as tasks, tools, and environments change. Imaginarium and Confucius learn from simulated trial and error or self-generated curricula [62, 243]. Math CL cycles through fine-tuning, fresh synthesis, preference optimization, and replay so that new synthetic data improves current performance without erasing earlier skills [241]. This setting makes the lifecycle view explicit: the agent's current behavior changes the data that will train its future behavior. The main challenge is therefore not only generating new cases, but also curating them across rounds so that synthetic noise is corrected rather than accumulated.

5 Synthetic Evaluation Data for Agents

Synthetic data also serves as evaluation infrastructure. The previous sections treated synthetic artifacts as training material; here, similar artifacts make agent behavior observable under controlled conditions. A benchmark must specify the task, tools, initial state, hidden user goal, interaction environment, scoring rule, and often a simulated user or adversary. These components are difficult to obtain from live deployments because APIs change, websites drift, user state is private, and long trajectories can mix planning, tool use, memory, recovery, and safety failures in a single final score. Synthetic evaluation therefore becomes a form of verifier construction: it determines which parts of the agent–environment loop are observable, which states can be reset, which failures can be diagnosed, and which risky situations can be tested without harming real users or systems. Tab. 1 summarizes representative component-level and scenario-level benchmarks discussed in this section and §6.

Type	Area	Representative benchmarks
Agent Module	Agent Planning	PlanBench [237]; TravelPlanner [278]; TripCraft [16]; BioPlanner [175];
	Agent Memory	LoCoMo [160]; LongMemEval [268]; MemoryAgentBench [84]; MemSim [332]
	Agent Tool Use	API-Bank [124]; BFCL [182]; ACEBench [18]; τ-bench [301]; τ^2-bench [8]; MCP-Universe [155]; MCPMark [276]; MCP-Atlas [7]; Toolathlon [121]
Scenario	Software Engineering	HumanEval+ [140]; LiveCodeBench [95]; BigCodeBench [347]; SWE-bench Verified [102]; SWE-Gym [178]; MLE-bench [15]
	Agentic Search	FRAMES [109]; AssistantBench [306]; WebWalkerQA [270]; HotpotQA [299]
	AI for Science	DiagBench [195]; 3MDbench [227]; ReasonMed [226]; ChemToolBench [271]
	Social Good	Agent4Edu [63]; SADAS [87]; CharacterCraft [305]; ESC-Judge [159]
	AI Safety and Security	FinRpt-Gen [103]; AIDSAFE [112]; ASTRA [286]; ASB [320]

Table 1: Representative agent benchmarks discussed in §5 and §6, grouped by agent module and application scenario.

5.1 Synthetic Evaluation Benchmarks

Agent Tool Use Tool benchmarks evaluate whether an agent can choose the right tool, fill its arguments, chain dependent calls, interpret outputs, and leave the backend in the intended state. Early benchmarks emphasized call-level correctness. Gorilla grades the structure of generated API calls rather than only matching surface text [181], API-Bank extends tool use to multi-turn API dialogues [124], and ToolBench adds executable trajectories constructed over dependent tool calls [193]. Later benchmarks add state and protocol complexity. BFCL checks backend-state changes for mock APIs [182], while ToolSandbox, StableToolBench, and DICE-Bench introduce simulated users, stable virtual backends, or explicit tool-dependency graphs [73, 97, 149]. The most recent benchmarks move from static or mocked interfaces toward realistic software environments: MCP-Universe evaluates agents over real MCP servers [155], and later MCP benchmarks scale this setting to larger tool ecosystems, live connectors, diagnostic states, and long-horizon workflows [7, 121, 166, 259, 276]. This progression improves ecological validity, but it also makes reproducibility harder because live-like benchmarks must stabilize backend state, tool versions, and scoring rules.

Human-Agent Interaction Human-agent interaction benchmarks add a variable that tool benchmarks usually omit: the goal is partially hidden inside another actor. The benchmark must therefore synthesize private goals, policies, personas, role backgrounds, payoff structures, or organizational state, and the agent must infer them through interaction [348]. Customer-service benchmarks make this setting concrete by pairing user simulators with synthetic backends. τ-bench scores whether a retail or airline agent reaches the correct final database state [301], while τ^2-bench makes both sides act under partial observability [8]. Persona and role-play benchmarks fix hidden profiles and judge consistency across long conversations [45, 190, 207, 236, 257, 337, 340]. Multi-agent and organizational benchmarks extend the same hidden-state idea to bargaining, coordination, mixed collaboration-competition, and enterprise work [2, 9, 283, 284, 342]. The main risk is simulator fidelity: an LLM-based user may be too cooperative, reveal private information too easily, or share biases with the evaluated agent. In that case, the benchmark may measure compatibility with the simulator rather than competence with real users, so strong interaction benchmarks need calibration against human behavior and real deployment failures in addition to diverse synthetic personas.

Long-Horizon Agent Evaluations Long-horizon benchmarks address a diagnostic problem that becomes visible only after many steps: a final score often hides where the agent failed. Synthetic evaluation makes intermediate behavior measurable by planting facts, updates, conflicts, timestamps, subgoals, milestones, tests, or rubrics. Memory benchmarks plant ground-truth facts and later test whether the agent preserves them under distraction or update. LongMemEval uses gold anchors and

distractor sessions [268], LoCoMo builds event-graph timelines [160], MemSim controls fact distributions with a Bayesian simulator [332], and MemoryAgentBench isolates updates and conflicts [84]. Process-oriented benchmarks expose partial progress through solution paths, subgoal sequences, or milestones [5, 66, 153, 157, 193]. User-intent-shift benchmarks extend this idea by changing the target mid-trajectory and measuring whether a web agent can be interrupted and redirected [349]. Open-ended professional and research benchmarks use generated tests, rubric trees, agentic judges, or executable checks when no single gold path exists [15, 21, 35, 67, 99, 104, 170, 219, 232, 267, 307, 346]. These signals are more diagnostic than final success, but they can also overconstrain valid alternative strategies if the gold path or rubric is too narrow.

Agent Safety and Security Safety and security benchmarks synthesize risky situations that should not be tested directly on live systems [22]. Attack benchmarks create sandboxes in which unsafe behavior can be observed without real harm. ToolEmu uses a language-model-emulated tool environment to reveal unsafe tool actions [205], ASTRA automates spatial and temporal red-teaming for coding assistants [286], and ASB formalizes common agent-security attacks and defenses [320]. Robustness benchmarks perturb missions, user goals, or vulnerable codebases to test whether the agent recovers and stays within policy [116, 202, 308]. Social-risk benchmarks simulate adversarial harassment attacks [176]. These benchmarks should report task completion and safety together: broad refusal can look safe while failing the task, and policy-blind completion can look capable while violating constraints. Synthetic evaluation makes such tradeoffs measurable under controlled conditions, but the verdict remains trustworthy only when the scenarios, user simulators, and scoring rules are calibrated against real incidents and human judgment.

6 Applications

Having traced how synthetic artifacts are produced (§2), filtered for quality (§3), and consumed in learning (§4) and evaluation (§5), we now turn to where these pipelines are actually deployed. We organize the landscape along two complementary cuts: the core agent modules that synthesis must support (§6.1), and the end-to-end application scenarios in which those modules are exercised under domain-specific pressures (§6.2). Across both, one pattern recurs: as tasks grow more interactive and long-horizon, useful synthesis shifts away from isolated input–output pairs toward generating the users, environments, and feedback that make complex behaviors both learnable and verifiable.

6.1 Agent Module

An agent's competence rests on a handful of reusable capabilities: deciding what to do, acting on the world through tools, and retaining state across time. Each of them poses a distinct synthesis bottleneck, which we examine in turn.

Agent Planning Agentic planning turns a user goal into executable subgoals or tool calls, so it is a natural bottleneck for data synthesis. Planning benchmarks cover classical PDDL planning, constrained itinerary planning, and diagnostic long-horizon tasks: PlanBench [237] tests Blocksworld-style domains, TravelPlanner [278] and TripCraft [16] test hard travel constraints, and BioPlanner [175] plus HeroBench [5] remain mostly diagnostic. Synthesis work concentrates on two of these settings, classical and itinerary planning, which differ in where validity comes from and which we take in turn.

For classical planning, early work fine-tunes on synthetic symbolic plans, while later work improves diversity, search traces, and validation. Plansformer [177] distills Fast-Downward problem-plan pairs, CMDS [127] reduces data cost through graph-based diversity selection, and Bohnet et al. [11] show that scaling Blocksworld data still leaves out-of-distribution generalization brittle. System-1.x [206] synthesizes A* traces with subgoal-level mode labels, PDDL-INSTRUCT [239] uses logical traces and VAL feedback for obfuscated PDDL, Plan2Evolve [88] self-evolves new domains, and Correa et al. [37] replace hand-designed rewards with iterative filtering.

For itinerary planning, where feasibility hinges on hard external constraints rather than symbolic preconditions, synthesis mainly supplies feedback and rewards. FAFT [30] labels synthetic plans with an oracle checker, TTG [105] maps natural language to MILP constraints, DeepTravel [173] combines cold-start trajectories, RL samples, and a spatio-temporal verifier, and TripScore [197] uses

a continuous feasibility-and-quality reward. IMAGINE [331] distills GPT-4o multi-agent traces into an 8B model, while STAR [272] studies which parts of a Synthesis-SFT-RL pipeline actually matter. Across both planning lines, gains mostly come from richer synthetic supervision, but many pipelines still depend on large teachers.

Agent Tool Use If planning fixes what an agent intends to do, tool use governs whether those intentions survive contact with real interfaces. Tool-use applications show a shift from static function-call data to multi-turn trajectories and simulated interaction. API-Bank [124], BFCL [182], ACEBench [18], τ-bench [301], and τ^2-bench [8] now span multi-turn API dialogues, single- and multi-turn function calling, ambiguous or incomplete instructions, customer-service tool-agent-user interaction, and dual-control settings. Recent methods differ mainly in how they synthesize interaction: PARL-MT [13] adds progress-aware summaries, MUA-RL [333] trains with simulated users, ToolACE-MT [316] and ToolACE-R [315] use iterative refinement, TRUSTEE [229] uses a local model to simulate tasks, users, tools, and rewards, ToolMind [293] uses function-graph sampling and turn-level filtering, APIGen-MT/xLAM-2 [188] builds verified blueprints and simulated human-agent trajectories, and Tool-N1 [328] applies RL over structural and functional rewards. The common pattern is that realistic tool use requires synthetic users, state transitions, and interaction-level feedback, not only schema-level function-call correctness.

The same trend holds at larger scale, where model and dataset reports converge on the same recipe. APIGen-MT/xLAM-2 [188] centers on executable trajectory blueprints, ToolMind [293] mixes open-source data with graph-sampled synthetic traces, Kimi K2 [231] integrates large-scale agentic synthesis with joint RL in real and synthetic environments, and TRUSTEE [229] shows that self-simulated tool environments can provide a lower-cost training signal.

Agent Memory Planning and tool use both operate within a single episode; memory is what sustains competence across episodes. Agentic memory requires agents to decide what to write, read, and update across sessions [90]. Benchmarks probe cross-session recall, fact updates, conflict resolution, temporal reasoning, and multi-platform state tracking through LoCoMo [160], PerLTQA [49], LongMemEval [268], MemoryAgentBench [84], MEMTRACK [42], TReMu [65], and MemSim [332]. Training work focuses mostly on LoCoMo and LongMemEval, where synthetic trajectories and rewards are available, and splits into two complementary lines.

One line treats memory operations as learnable actions. Memory-R1 [291] labels ADD / UPDATE / DELETE / NOOP decisions with a teacher model, MEM-α [256] scales synthetic multi-turn streams and long contexts, MemLoRA [10] distills memory skills into small adapters, MemBuilder [209] adds dense session-level rewards, and MemSkill [322] expands the skill bank during training. A second line synthesizes long-term conversations around planted facts: LongMemEval [268] provides distractor sessions and query expansion, RMM [228] trains retrieval with LLM-cited evidence, UMA [327] builds Ledger-QA for end-to-end state maintenance, and InfMem [253] uses teacher trajectories over very long contexts. AtomMem [94] and Mem-T [311] reinforce the same shift toward learnable memory operations. The remaining gap is coverage: several memory benchmarks still lack synthesis-trained entries.

6.2 Application Scenarios

The three modules rarely act in isolation; deployed agents weave them together under the constraints of a concrete domain. We therefore move from capabilities to scenarios, where planning, tool use, and memory are exercised jointly and where each domain imposes its own standard for what counts as a valid trajectory and a trustworthy verifier.

Software Engineering Among these scenarios, software engineering offers the cleanest grounding, since repositories and their test suites already behave as executable verifiers. Software engineering agents translate natural-language intents into code edits, shell commands, and test executions across large repositories, making them a stress test for long-horizon trajectory synthesis. Recent benchmarks span function-level generation, repository-level issue resolution, and end-to-end ML engineering: HumanEval+ [140] hardens the canonical function-synthesis baseline with stronger test suites, while LiveCodeBench [95] and BigCodeBench [347] introduce contamination-resistant, time-stamped contests and library-rich function calls; SWE-bench [102] and its human-validated SWE-bench Verified subset have become the de facto evaluation for repository-level agents such as SWE-agent [294],

OpenHands [252], Agentless [277], Moatless, and SWE-RL [265], with SWE-Gym [178] providing the paired training environment and Commit0 [334] pushing toward from-scratch library reconstruction; SWT-bench [167] isolates reproduction-test generation; and MLE-bench [15] probes full ML engineering pipelines over 75 Kaggle competitions.

Because that grounding is executable, software-engineering agents use synthesis to turn repositories into tasks, trajectories, tests, and rewards. SWE-smith [295] generates bug-triggering task instances from codebases, COAST [145] synthesizes debugging dialogues with problem-setter, learner, and teacher agents, CURE [254] co-evolves code and unit-test generators through pairwise rewards, and SWE-Dev [245] combines training and inference scaling for software-engineering agents. Tool- and API-oriented pipelines also matter for coding agents: ToolBench [193] builds DFS tool-use trajectories, APIGen-MT [188] uses task blueprints, ToolACE [142] self-evolves function-calling data, ToolMind [293] adds reasoning chains, and MAGNET [304] translates dependency graphs into multi-turn interactions. Execution-grounded work validates trajectories by running them, as in CodeAct [251] and CodeContests+ [260]. Open frameworks such as SWE-agent, OpenHands, and Agentless show that strong SWE-bench Verified performance is possible without proprietary infrastructure, while commercial tools now split between IDE-embedded copilots and more autonomous coding agents.

Agentic Search If software engineering gives synthesis a closed world with built-in checkers, agentic search faces the opposite condition: an open, drifting web where ground truth is dispersed and must be reconstructed rather than executed. Deep research and agentic search agents are evaluated along two axes: open-web browsing under realistic latency and multi-hop reasoning over retrieved evidence, and recent benchmarks have shifted decisively toward harder, hint-free settings. BrowseComp [263] has emerged as the dominant frontier evaluation for browsing agents such as OpenAI Deep Research, WebSailor [123], DeepDive [324], and Fathom-DeepResearch [72], with the text-only subset of Humanity's Last Exam [185] increasingly used to stress-test long-horizon retrieval and reasoning; FRAMES [109] and AssistantBench [306] probe factuality and realistic web-task completion, and WebWalkerQA [270] targets multi-hop navigation specifically. For training and shorter-horizon evaluation, multi-hop QA datasets including HotpotQA [299], 2WikiMultihopQA [81], MuSiQue [234], and Bamboogle [189] remain the workhorse testbeds for synthesized search trajectories used in Search-R1, R1-Searcher, and DeepResearcher, with SimpleQA [262] serving as a short-form factuality calibration baseline.

With no executable oracle to lean on, search synthesis must build its own validity into each trajectory. Deep research agents need multi-hop search [164], source selection, and long-form synthesis trajectories that are complex but still verifiable. Structured-knowledge methods synthesize search paths from knowledge graphs, MCTS over KBs, urban KG construction, abductive evidence-to-query paths, or factorized deep-search evaluation [125, 152, 163, 172, 213, 324]. Self-evolving methods score, filter, and retrain on search trajectories for web search, paper search, long reasoning, and open-source deep-research tasks [72, 78, 92, 319, 341]. Feedback-driven RAG methods synthesize trajectories or preferences for retrieval, reranking, generation, peer review, and executable scientific tasks [128, 131, 184, 266, 317]. Open systems such as STORM and MindSearch show that the same agentic search loop can be built outside closed products; the key bottleneck is increasingly reasoning over retrieved evidence, not retrieval alone.

AI for Science The same premium on verifiable reasoning over imperfect evidence carries into science and medicine, where a further obstacle dominates: the data itself is hard to obtain. Healthcare and scientific discovery use synthesis because real data is private, expensive, or tied to complex workflows. Several generated benchmarks leverage the conversational and reasoning abilities of LLMs, including DiagBench for multi-turn diagnostic interaction trajectories [195], 3MDbench for multimodal medical dialogues [227], and ReasonMed for advanced medical reasoning [226].

On the method side, work clusters by sub-domain. In medical diagnostics, DiagGym and DiagAgent [195] create virtual clinical environments for RL over diagnostic trajectories, while ReasonMed [226] distills and refines large numbers of medical reasoning paths. In mental health, PsyCoTalk [242] synthesizes psychiatrist-validated diagnostic dialogues from clinical protocols, CATCH [26] uses progressive dialogue synthesis for counseling, PACE [117] builds panic-support data from first-person narratives, and PsyDial [194] reconstructs real dialogues in a privacy-preserving form. In chemistry and science, CheMatAgent [271] integrates chemical tools and HE-MCTS plan-

ning through ChemToolBench, while MIMIR [230] uses role-play to generate agent-tuning data for scientific tasks. The shared theme is that synthesis supplies interactive, verifiable practice where real supervision is hard to collect.

Social Good When the object of synthesis is human behavior rather than verifiable facts, the difficulty shifts in kind: the challenge is less about acquiring data than about convincingly standing in for the people an agent must serve. Human-facing agents use synthesis to simulate learners (e.g., Agent4Edu with generated learner responses [63]), collaborators and negotiators (e.g., SADAS with synthetic business negotiations [87]), personas (e.g., CharacterCraft for character dialogues from novels [305]), and emotional-support seekers (e.g., ESC-Judge with generated help-seeker conversations [159]).

These roles recur across three broad settings. In education, Agent4Edu [63] simulates students with learner profiles and memory, while DataEnvGym [107] treats data generation as a teacher-agent problem guided by student-model feedback. For collaboration, Assistive LLM Agents [87] use role-play and remediation in negotiation, Coral [171] uses self-play to create collaborative preference data, and DiscussLLM [180] teaches agents when to stay silent or intervene in group discussion. For role-play and support, personality-scale data [201], CharacterCraft [305], and SimsChat/SimsConv [292] improve character consistency, ESC-Judge [159] synthesizes help-seeker roles for theory-grounded evaluation, and SAPIENT [47] uses MCTS for conversational planning. These domains benefit from controllable simulation, but fidelity to real social behavior remains the hard part.

AI Safety and Security Fidelity matters most when the behaviors being modeled are dangerous in themselves. High-stakes domains use synthesis to create professional tasks and controlled risk scenarios, such as FinRpt-Gen for equity research [103], ExCyTIn-Bench for cyber threat investigation [273], ASB for agent security attacks and defenses [320], and AIDSAFE [112] for safety reasoning.

Concretely, FinRpt-Gen [103] builds multi-agent equity-research-report data from several financial sources, and Beyond Reactive Safety [224] uses long-horizon simulations to reveal indirect harms from model advice. Security work synthesizes vulnerable projects, red-team probes, cyber-investigation questions, and agent attack-defense tasks through SEC-bench [116], ASTRA [286], ExCyTIn-Bench [273], and ASB [320]. Multi-turn jailbreak work creates both attacks and defenses: X-Teaming [200] optimizes collaborative multi-turn attacks and releases XGuard-Train, while AIDSAFE [112] synthesizes policy-embedded reasoning data for safer responses. Across these frontiers, synthetic data is valuable because the target failures are rare, risky, or too costly to collect directly.

Across both the modules and the scenarios, the same tension recurs in domain-specific dress: progress depends less on producing more synthetic data than on whether the synthesized source distribution covers the behaviors that matter and whether its verifier can be trusted. Such two questions organize the open challenges that we discuss in §7.

7 Challenges and Future Directions

Data synthesis can scale agent learning, but it also introduces risks from weak grounding, noisy feedback, long-horizon interaction, and simulated environments. These risks are manifestations of two deeper questions: whether the source distribution provides supervision that matches the target behavior, and whether the verifier provides feedback that is valid enough to filter, rank, or improve that supervision. A third, more basic question underlies both: whether the model tasked with synthesis is itself capable of producing such sources and verifiers reliably. We organize the discussion around these three axes: source-distribution reliability (§7.1), verifier reliability (§7.2), and synthesis capability (§7.3), pairing each diagnosis with concrete directions for future work.

7.1 Source Distribution Reliability

Data synthesis and distillation can be viewed as sampling supervision from a source distribution conditioned on seeds such as prompts, tasks, demonstrations, tools, environments, or evaluation harnesses. Agent synthesis enriches this process because the sampled object is not only an input–output pair, but a joint structure over goals, tool calls, observations, state transitions, trajectories, feedback, and sometimes memory. The central risk is whether this source distribution preserves the

factors that determine target behavior, rather than surface style, simulator artifacts, or prompt-template regularities.

Several common concerns in agent synthesis can be viewed as manifestations of this source-distribution problem. For example, limited task coverage means the source distribution may miss important goals, tools, user states, or failure modes, causing downstream agents to learn only a narrow slice of the target behavior. Sandbox training may overrepresent simplified APIs or unrealistic user behavior, so behavior learned in the synthetic environment may not transfer to deployment. Recursive self-improvement can also repeatedly sample from and reinforce the same biased source distribution across training rounds [32, 68, 212, 338].

What unifies these concerns is that the field still lacks ways to tell a well-targeted source distribution from a merely large one. Future work should move beyond reporting dataset size or task diversity and characterize which features of a source distribution actually predict downstream agent improvement. These features include coverage over goals, tools, user states, environment dynamics, failure modes, and recovery paths; grounding in executable constraints and valid state transitions; and calibration of task difficulty, simulator behavior, and feedback signals to deployment conditions. A useful direction is to develop diagnostics that identify whether failures come from missing task modes, unrealistic environment assumptions, invalid trajectory dynamics, or difficulty distributions that are mismatched to the policy being trained.

7.2 Verifier Reliability

A reliable source distribution is necessary but not sufficient: even well-chosen supervision is only as good as the feedback used to filter, rank, and reward it. The second question is whether verifier feedback is a faithful proxy for task success. Objective verifiers are strongest when success can be decomposed into rules, tests, executable constraints, or state changes, but such decomposition is difficult when the relevant factors of success are not yet fully understood, isolated, or quantifiable. Subjective verifiers, such as LLM-as-a-judge or learned reward models, cover broader tasks but replace tacit human preferences with an explicit rubric or learned judge, which may fail to align with human expectations beyond its calibration data. For agents, the most grounded verifier is the environment itself: whether actions produce the intended effects under real constraints.

Future work should therefore treat verifier design as a problem of proxy alignment: which aspects of task success the verifier captures, which aspects it omits, and under what distribution shifts its judgments remain valid. This requires audits that compare synthetic feedback with executable checks, human judgments, and real task outcomes, rather than assuming that judge scores or reward-model labels are reliable training targets. Long-horizon error propagation is a concrete example: a small hallucinated tool call, invalid state update, or wrong observation can contaminate later steps if no process-level verifier catches it early. Synthesis-time validation is useful only if intermediate checks reveal whether failures arise from task construction, trajectory generation, feedback modeling, or data filtering. Dynamic evaluation raises a related issue: scoring should use state changes and support valid alternative trajectories, rather than relying only on fixed references.

Catching such errors early, before they propagate, requires verification that does not wait for the final outcome. Future harnesses should also turn sparse final outcomes into dense process-level verifiability through intermediate state changes, tool-call validity, subgoal completion, rollback signals, memory consistency, and environment feedback. The long-term goal is for agents to learn not only from final success or failure, but also to discover and use intermediate checks that make their own execution process more observable, debuggable, and correctable.

7.3 Synthesis Capability

Both the source distribution and the verifier are, in the end, produced by a model, which raises a question upstream of either: is that model actually able to synthesize them well? It remains unclear how a model's problem-solving capability translates into its ability to synthesize useful agent training data. A strong model may solve a task correctly, but this does not guarantee that it can generate diverse tasks, faithful trajectories, reliable feedback, or training examples that improve other agents. Data synthesis therefore depends not only on the generator model's raw capability, but also on how the synthesis process elicits, verifies, filters, and diversifies the generated artifacts.

Future work should study capability-aware synthesis pipelines that assign different roles to models with different strengths. For example, stronger models could be used to plan task distributions, guide trajectory construction, identify missing coverage, and verify generated samples, while smaller models could support large-scale generation, paraphrasing, perturbation, and preliminary filtering under executable or human-grounded supervision. This also requires evaluating synthesis capability directly: whether a model can produce tasks with controlled difficulty, trajectories that preserve causal state changes, feedback that agrees with reliable verifiers, and data mixtures that improve a target policy rather than merely resembling high-quality demonstrations. Such evaluations would separate the ability to solve tasks from the ability to construct useful supervision for other agents.

Across all three axes, the requirement for trustworthy synthesis is the same: a source distribution that samples the behaviors that matter, a verifier that judges them faithfully, and a generator capable of producing both, and progress on any one is ultimately bounded by the other two.

8 Conclusion

This survey frames data synthesis for LLM agents as a workflow built around agent artifacts. We organize artifact synthesis into four levels: tasks, trajectories, feedback signals, and environments. We then examine how these artifacts are controlled for quality, used in agent learning, turned into evaluation infrastructure, and applied across agent modules and application scenarios. Across these levels, the central challenge is not merely generating more data, but identifying reliable source distributions and constructing verifiers that expose dense, environment-grounded learning signals.

References

[1] Marwa Abdulhai, Ryan Cheng, Donovan Clay, Tim Althoff, Sergey Levine, and Natasha Jaques. Consistently simulating human personas with multi-turn reinforcement learning. *arXiv preprint arXiv:2511.00222*, 2025.

[2] Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. LLM-Coordination: Evaluating and analyzing multi-agent coordination abilities in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025. URL https://aclanthology.org/2025.findings-naacl.448/.

[3] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, 2024.

[4] Afra Feyza Akyurek, Ekin Akyurek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. RL4F: Generating natural language feedback with reinforcement learning for repairing model outputs. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7716–7733, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.427. URL https://aclanthology.org/2023.acl-long.427/.

[5] Petr Anokhin, Roman Khalikov, Stefan Rebrikov, Viktor Volkov, Artyom Sorokin, and Vincent Bissonnette. Herobench: A benchmark for long-horizon planning and structured reasoning in virtual worlds. *arXiv preprint arXiv:2508.12782*, 2025.

[6] Hao Bai, Alexey Taymanov, Tong Zhang, Aviral Kumar, and Spencer Whitehead. Webgym: Scaling training environments for visual web agents with realistic tasks. *arXiv preprint arXiv:2601.02439*, 2026.

[7] Chaithanya Bandi, Razvan-Gabriel Dumitru, Ben Hertzberg, Divyansh Agarwal, Geobio Boo, Tejas Polakam, Sami Hassaan, Jeff Da, HiJae Kim, Vipul Gupta, Manasi Sharma, Andrew Park, Martin Dimakis, Ernesto Gabriel Hernandez Montoya, Dan Rambado, Ivan Salazar, Rafael Cruz, MohammadHossein Rezaei, Chetan Rane, Ben Levin, Daniel Yue Zhang, Brad Kenstler, and Bing Liu. MCP-Atlas: A large-scale benchmark for tool-use competency with real MCP servers, 2026. URL https://arxiv.org/abs/2602.00933.

[8] Victor Barres et al. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

[9] Federico Bianchi, Patrick John Chia, Mert Yuksekgonul, Jacopo Tagliabue, Dan Jurafsky, and James Zou. How well can LLMs negotiate? NegotiationArena platform and analysis. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 3935–3951, 2024. URL https://proceedings.mlr.press/v235/bianchi24a.html.

[10] Massimo Bini, Ondrej Bohdal, Umberto Michieli, Zeynep Akata, Mete Ozay, and Taha Ceritli. Memlora: Distilling expert adapters for on-device memory systems. *arXiv preprint arXiv:2512.04763*, 2025.

[11] Bernd Bohnet, Azade Nova, Aaron T Parisi, Kevin Swersky, Katayoon Goshvadi, Hanjun Dai, Dale Schuurmans, Noah Fiedel, and Hanie Sedghi. Exploring and benchmarking the planning capabilities of large language models. *arXiv preprint arXiv:2406.13094*, 2024.

[12] Shihao Cai, Runnan Fang, Jialong Wu, Baixuan Li, Xinyu Wang, Yong Jiang, Liangcai Su, Liwen Zhang, Wenbiao Yin, Zhen Zhang, et al. Autoforge: Automated environment synthesis for agentic reinforcement learning. *arXiv preprint arXiv:2512.22857*, 2025.

[13] Huacan Chai, Zijie Cao, Maolin Ran, Yingxuan Yang, Jianghao Lin, Pengxin, Hairui Wang, Renjie Ding, Ziyu Wan, Muning Wen, Weiwen Liu, Weinan Zhang, Fei Huang, and Ying Wen. PARL-MT: Learning to call functions in multi-turn conversation with progress awareness. *arXiv preprint arXiv:2509.23206*, 2025.

[14] Jiajun Chai, Sicheng Li, Yuqian Fu, Dongbin Zhao, and Yuanheng Zhu. Empowering llm agents with zero-shot optimal decision-making through q-learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[15] Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=6s5uXNWGIh.

[16] Soumyabrata Chaudhuri, Pranav Purkar, Ritwik Raghav, Shubhojit Mallick, Manish Gupta, Abhik Jana, and Shreya Ghosh. Tripcraft: A benchmark for spatio-temporally fine grained travel planning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17035–17064, 2025.

[17] Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*, 2023.

[18] Chen Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Yuefeng Huang, Wulong Liu, Xinzhi Wang, Defu Lian, Baoqun Yin, Yasheng Wang, and Wu Liu. ACEBench: Who wins the match point in tool usage? *arXiv preprint arXiv:2501.12851*, 2025.

[19] Guhong Chen, Chenghao Sun, Cheng Fu, Qiyao Wang, Zhihong Huang, Chaopeng Wei, Guangxu Chen, and Feiteng Fang. Beyond quantity: Trajectory diversity scaling for code agents, 2026. URL https://arxiv.org/abs/2602.03219.

[20] Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or llms as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, 2024.

[21] Hui Chen, Miao Yuan, Zhicheng Wang, Heng Ji, and Jiebo Luo. MLR-Bench: Evaluating AI agents on open-ended machine learning research, 2025. URL https://arxiv.org/abs/2505.19955.

[22] Huiyi Chen, Jiawei Peng, Dehai Min, Changchang Sun, Kaijie Chen, Yan Yan, Xu Yang, and Lu Cheng. Mvi-bench: A comprehensive benchmark for evaluating robustness to misleading visual inputs in lvlms. *arXiv preprint arXiv:2511.14159*, 2025.

[23] Jinpeng Chen, Cheng Gong, Hanbo Li, Ziru Liu, Zichen Tian, Xinyu Fu, Shi Wu, Chenyang Zhang, Wu Zhang, Suiyun Zhang, Dandan Tu, and Rui Liu. Cove: Training interactive tool-use agents via constraint-guided verification, 2026. URL https://arxiv.org/abs/2603.01940.

[24] Junting Chen, Haotian Liang, Lingxiao Du, Weiyun Wang, Mengkang Hu, Yao Mu, Wenhai Wang, Jifeng Dai, Ping Luo, Wenqi Shao, et al. Owmm-agent: Open world mobile manipulation with multi-modal agentic data synthesis. *Advances in Neural Information Processing Systems*, 38:130020–130049, 2026.

[25] Minghao Chen, Yihang Li, Yanting Yang, Shiyu Yu, Binbin Lin, and Xiaofei He. Automanual: Constructing instruction manuals by llm agents via interactive environmental learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 589–631. Curran Associates, Inc., 2024. doi: 10.52202/079017-0019. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/0142921fad7ef9192bd87229cdafa9d4-Paper-Conference.pdf.

[26] Mingyu Chen, Jingkai Lin, Zhaojie Chu, Xiaofen Xing, Yirong Chen, and Xiangmin Xu. CATCH: A novel data synthesis framework for high therapy fidelity and memory-driven planning chain of thought in AI counseling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10254–10286, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.543. URL https://aclanthology.org/2025.findings-emnlp.543/.

[27] Sijia Chen, Yibo Wang, Yi-Feng Wu, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Lijun Zhang. Advancing tool-augmented large language models: Integrating insights from errors in inference trees. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ZIpdu0cHYu.

[28] Weize Chen, Jiarui Yuan, Chen Qian, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Optima: Optimizing effectiveness and efficiency for LLM-based multi-agent system. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11534–11557, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.601. URL https://aclanthology.org/2025.findings-acl.601/.

[29] Xuanzhong Chen, Zile Qiao, Guoxin Chen, Liangcai Su, Zhen Zhang, Xinyu Wang, Pengjun Xie, Fei Huang, Jingren Zhou, and Yong Jiang. Agentfrontier: Expanding the capability frontier of llm agents with zpd-guided data synthesis. *arXiv preprint arXiv:2510.24695*, 2025.

[30] Yanan Chen, Ali Pesaranghader, Tanmana Sadhu, and Dong Hoon Yi. Can we rely on llm agents to draft long-horizon plans? let's take travelplanner as an example. *arXiv preprint arXiv:2408.06318*, 2024.

[31] Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. Agent-flan: Designing data and methods of effective agent tuning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9354–9366, 2024.

[32] Zhaorun Chen, Zhuokai Zhao, Kai Zhang, Bo Liu, Qi Qi, Yifan Wu, Tarun Kalluri, Sara Cao, Yuanhao Xiong, Haibo Tong, Huaxiu Yao, Hengduo Li, Jiacheng Zhu, Xian Li, Dawn Song, Bo Li, Jason Weston, and Dat Huynh. Scaling Agent Learning via Experience Synthesis, November 2025. URL http://arxiv.org/abs/2511.03773. arXiv:2511.03773 [cs].

[33] Zhenfang Chen, Delin Chen, Rui Sun, Wenjun Liu, and Chuang Gan. Scaling autonomous agents via automatic reward modeling and planning. *arXiv preprint arXiv:2502.12130*, 2025.

[34] Zhi Chen, Qiguang Chen, Libo Qin, Qipeng Guo, Haijun Lv, Yicheng Zou, Hang Yan, Kai Chen, and Dahua Lin. What are the essential factors in crafting effective long context multi-hop instruction datasets? insights and best practices. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27129–27151, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1316. URL https://aclanthology.org/2025.acl-long.1316/.

[35] Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=6z4YKr0GK6.

[36] Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. Instruction pre-training: Language models are supervised multitask learners. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2550, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.148. URL https://aclanthology.org/2024.emnlp-main.148/.

[37] Augusto B Corrêa, Yoav Gelberg, Luckeciano C Melo, Ilia Shumailov, André G Pereira, and Yarin Gal. Iterative deployment improves planning skills in llms. *arXiv preprint arXiv:2512.24940*, 2025.

[38] Jieming Cui, Tengyu Liu, Meng Ziyu, Yu Jiale, Ran Song, Wei Zhang, Yixin Zhu, and Siyuan Huang. Grove: A generalized reward for learning open-vocabulary physical skill. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[39] Jeff Da, Clinton Wang, Xiang Deng, Yuntao Ma, Nikhil Barhate, and Sean Hendryx. Agent-rlvr: Training software engineering agents via guidance and environment rewards. *arXiv preprint arXiv:2506.11425*, 2025.

[40] Nicola Dainese, Matteo Merler, Minttu Alakuijala, and Pekka Marttinen. Generating code world models with large language models guided by monte carlo tree search. *Advances in Neural Information Processing Systems*, 37:60429–60474, 2024.

[41] Yuchen Deng, Shichen Fan, Naibo Wang, Xinkui Zhao, and See-Kiong Ng. AgentPro: Enhancing LLM agents with automated process supervision. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9981–10006, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.506. URL https://aclanthology.org/2025.emnlp-main.506/.

[42] Darshan Deshpande, Varun Gangal, Hersh Mehta, Anand Kannappan, Rebecca Qian, and Peng Wang. Memtrack: Evaluating long-term memory and state tracking in multi-platform dynamic agent environments. *arXiv preprint arXiv:2510.01353*, 2025.

[43] Zixiang Di, Jinyi Han, Shuo Zhang, Ying Liao, Zhi Li, Xiaofeng Ji, Yongqi Wang, and Zheming Yang. Not all negative samples are equal: Llms learn better from plausible reasoning, 2026. URL https://arxiv.org/abs/2602.03516.

[44] Guanting Dong, Junting Lu, Junjie Huang, Wanjun Zhong, Longxiang Liu, Shijue Huang, Zhenyu Li, Yang Zhao, Xiaoshuai Song, Xiaoxi Li, Jiajie Jin, Yutao Zhu, Hanbin Wang, Fangyu Lei, Qinyu Luo, Mingyang Chen, Zehui Chen, Jiazhan Feng, Ji-Rong Wen, and Zhicheng Dou. Agent-world: Scaling real-world environment synthesis for evolving general agent intelligence, 2026. URL https://arxiv.org/abs/2604.18292.

[45] Yao Dou, Michel Galley, Baolin Peng, Chris Kedzie, Weixin Cai, Alan Ritter, Chris Quirk, Wei Xu, and Jianfeng Gao. SimulatorArena: Are user simulators reliable proxies for multi-turn evaluation of AI assistants? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. URL https://aclanthology.org/2025.emnlp-main.1786/.

[46] Zi-Yi Dou, Cheng-Fu Yang, Xueqing Wu, Kai-Wei Chang, and Nanyun Peng. Re-rest: Reflection-reinforced self-training for language agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15394–15411, 2024.

[47] Hanwen Du, Bo Peng, and Xia Ning. Sapient: Mastering multi-turn conversational recommendation with strategic planning and monte carlo tree search. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2629–2648, 2025.

[48] Weihua Du, Hailei Gong, Zhan Ling, Kang Liu, Lingfeng Shen, Xuesong Yao, Yufei Xu, Dingyuan Shi, Yiming Yang, and Jiecao Chen. Generalizable End-to-End Tool-Use RL with Synthetic CodeGym, 2025. URL https://arxiv.org/abs/2509.17325. Version Number: 2.

[49] Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang, and Kam-Fai Wong. Perltqa: A personal long-term memory dataset for memory classification, retrieval, and fusion in question answering. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 152–164, 2024.

[50] Maxence Faldor, Jenny Zhang, Antoine Cully, and Jeff Clune. OMNI-EPIC: Open-endedness via models of human notions of interestingness with environments programmed in code. In *International Conference on Learning Representations*, volume 2025, pages 85260–85385, 2025.

[51] Shengda Fan, Xin Cong, Yuepeng Fu, Zhong Zhang, Shuyan Zhang, Yuanwei Liu, Yesai Wu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Workflowllm: Enhancing workflow orchestration capability of large language models. In *International Conference on Learning Representations*, volume 2025, pages 24498–24525, 2025.

[52] Shengda Fan, Xin Cong, Zhong Zhang, Yuepeng Fu, Yesai Wu, Hao Wang, Xinyu Zhang, Enrui Hu, and Yankai Lin. Generalizing experience for language agents with hierarchical metaflows. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=QsQGMijLhL.

[53] Runnan Fang, Shihao Cai, Baixuan Li, Jialong Wu, Guangyu Li, Wenbiao Yin, Xinyu Wang, Xiaobin Wang, Liangcai Su, Zhen Zhang, et al. Towards general agentic intelligence via environment scaling. *arXiv preprint arXiv:2509.13311*, 2025.

[54] Runnan Fang, Xiaobin Wang, Yuan Liang, Shuofei Qiao, Jialong Wu, Zekun Xi, Ningyu Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. SynWorld: Virtual Scenario Synthesis for Agentic Action Knowledge Refinement, June 2025. URL http://arxiv.org/abs/2504.03561. arXiv:2504.03561 [cs].

[55] Tianqing Fang, Hongming Zhang, Zhisong Zhang, Kaixin Ma, Wenhao Yu, Haitao Mi, and Dong Yu. WebEvolver: Enhancing Web Agent Self-Improvement with Coevolving World Model, August 2025. URL http://arxiv.org/abs/2504.21024. arXiv:2504.21024 [cs].

[56] Wei Fang, Yang Zhang, Kaizhi Qian, James R. Glass, and Yada Zhu. PLAY2PROMPT: Zero-shot tool instruction optimization for LLM agents via tool play. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26274–26290, 2025. doi: 10.18653/v1/2025.findings-acl.1347. URL https://aclanthology.org/2025.findings-acl.1347/.

[57] Xueyang Feng, Jingsen Zhang, Jiakai Tang, Wei Li, Guohao Cai, Xu Chen, Quanyu Dai, Yue Zhu, and Zhenhua Dong. Expectation confirmation preference optimization for multi-turn conversational recommendation agent. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5896–5914, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.307. URL https://aclanthology.org/2025.findings-acl.307/.

[58] Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae, and Honglak Lee. Autoguide: Automated generation and selection of context-aware guidelines for large language model agents. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=mRIQz8Zd6O.

[59] Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. Aligning LLM agents by learning latent preference from user edits. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=DlYNGpCuwa.

[60] Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous RL, 2025.

[61] Jiaxuan Gao, Jiaao Chen, Chuyi He, Wei-Chen Wang, Shusheng Xu, Hanrui Wang, Di Jin, and Yi Wu. From self-evolving synthetic data to verifiable-reward rl: Post-training multi-turn interactive tool-using agents. *arXiv preprint arXiv:2601.22607*, 2026.

[62] Shen Gao, Zhengliang Shi, Minghang Zhu, Bowen Fang, Xin Xin, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. Confucius: Iterative tool learning from introspection feedback by easy-to-difficult curriculum. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):18030–18038, Mar. 2024. doi: 10.1609/aaai.v38i16.29759. URL https://ojs.aaai.org/index.php/AAAI/article/view/29759.

[63] Weibo Gao, Qi Liu, Linan Yue, Fangzhou Yao, Rui Lv, Zheng Zhang, Hao Wang, and Zhenya Huang. Agent4edu: Generating learner response data by generative agents for intelligent education systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23923–23932, 2025.

[64] Xin Gao, Qizhi Pei, Zinan Tang, Yu Li, Honglin Lin, Jiang Wu, Lijun Wu, and Conghui He. A strategic coordination framework of small LMs matches large LMs in data synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025.

[65] Yubin Ge, Salvatore Romeo, Jason Cai, Raphael Shu, Yassine Benajiba, Monica Sunkara, and Yi Zhang. Tremu: Towards neuro-symbolic temporal reasoning for llm-agents with memory in multi-session dialogues. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18974–18988, 2025.

[66] Ran Gong, Qiuyuan Huang, Xiaojian Ma, Yusuke Noda, Zane Durante, Zilong Zheng, Demetri Terzopoulos, Li Fei-Fei, Jianfeng Gao, and Hoi Vo. MindAgent: Emergent gaming interaction. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3154–3183, 2024. doi: 10.18653/v1/2024.findings-naacl.200. URL https://aclanthology.org/2024.findings-naacl.200/.

[67] Boyu Gou, Zanming Huang, Yuting Ning, Yu Gu, Michael Lin, Weijian Qi, Andrei Kopanev, Botao Yu, Bernal Jimenez Gutierrez, Yiheng Shu, Chan Hee Song, Jiaman Wu, Shijie Chen, Hanane Nour Moussa, TIANSHU ZHANG, Jian Xie, Yifei Li, Tianci Xue, Zeyi Liao, Kai Zhang, Boyuan Zheng, Zhaowei Cai, Viktor Rozgic, Morteza Ziyadi, Huan Sun, and Yu Su. Mind2Web 2: Evaluating agentic search with agent-as-a-judge. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=AUaW6DS9si.

[68] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023.

[69] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[70] Jiacheng Guo, Ling Yang, Peter Chen, Qixin Xiao, Yinjie Wang, Xinzhe Juan, Jiahao Qiu, Ke Shen, and Mengdi Wang. Genenv: Difficulty-aligned co-evolution between llm agents and environment simulators, 2025. URL https://arxiv.org/abs/2512.19682.

[71] Lianghong Guo, Yanlin Wang, Caihua Li, Wei Tao, Pengyu Yang, Jiachi Chen, Haoyu Song, Duyu Tang, and Zibin Zheng. SWE-Factory: Your Automated Factory for Issue Resolution Training Data and Evaluation Benchmarks, 2025. URL https://arxiv.org/abs/2506.10954. Version Number: 3.

[72] Siyuan Guo et al. Fathom-DeepResearch: Unlocking long horizon information retrieval and synthesis for SLMs. *arXiv preprint arXiv:2509.24107*, 2025.

[73] Zhicheng Guo, Sijie Cheng, Hao Wang, Shihao Liang, Yujia Qin, Peng Li, Zhiyuan Liu, Maosong Sun, and Yang Liu. StableToolBench: Towards stable large-scale benchmarking on tool learning of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. URL https://aclanthology.org/2024.findings-acl.664/.

[74] Priyanshu Gupta, Shashank Kirtania, Ananya Singha, Sumit Gulwani, Arjun Radhakrishna, Gustavo Soares, and Sherry Shi. MetaReflection: Learning instructions for language agents using past reflections. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8369–8385, 2024. doi: 10.18653/v1/2024.emnlp-main.477. URL https://aclanthology.org/2024.emnlp-main.477/.

[75] Bingguang Hao, Zengzhuang Xu, Yuntao Wen, Xinyi Xu, Yang Liu, Tong Zhao, Maolin Wang, Long Chen, Dong Wang, Yicheng Chen, Cunyin Peng, Xiangyu Zhao, Chenyi Zhuang, and Ji Zhang. From failure to mastery: Generating hard samples for tool-use agents, 2026. URL https://arxiv.org/abs/2601.01498.

[76] RISHI HAZRA, Alkis Sygkounas, Andreas Persson, Amy Loutfi, and Pedro Zuidberg Dos Martires. REvolve: Reward evolution with large language models using human feedback. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cJPUpL8mOw.

[77] Bowei He, Yankai Chen, Xiaokun Zhang, Linghe Kong, Philip S Yu, Xue Liu, and Chen Ma. Pedagogically-inspired data synthesis for language model knowledge distillation. In *The Fourteenth International Conference on Learning Representations*, 2026.

[78] Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, and Weinan E. PaSa: An LLM agent for comprehensive academic paper search. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11663–11679, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.572. URL https://aclanthology.org/2025.acl-long.572/.

[79] Yifei He, Pranit Chawla, Yaser Souri, Subhojit Som, and Xia Song. Webstar: Scalable data synthesis for computer use agents with step-level filtering, 2025. URL https://arxiv.org/abs/2512.10962.

[80] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[81] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *International Conference on Computational Linguistics (COLING)*, 2020.

[82] Simin Hong, Jun Sun, and Hongyang Chen. Third-person appraisal agent: Simulating human emotional reasoning in text with large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23684–23701, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1288. URL https://aclanthology.org/2025.findings-emnlp.1288/.

[83] Michael Y. Hu, Benjamin Van Durme, Jacob Andreas, and Harsh Jhamtani. Sample-efficient online learning in lm agents via hindsight trajectory rewriting, 2026. URL https://arxiv.org/abs/2510.10304.

[84] Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in llm agents via incremental multi-turn interactions. *arXiv preprint arXiv:2507.05257*, 2025.

[85] Zhaolin Hu, Yixiao Zhou, Zhongan Wang, Xin Li, Weimin Yang, Hehe Fan, and Yi Yang. OSDA agent: Leveraging large language models for de novo design of organic structure directing agents. In *The Thirteenth International Conference on Learning Representations*, 2025.

[86] Ziyou Hu, Zhengliang Shi, Minghang Zhu, Haitao Li, Teng Sun, Pengjie Ren, Suzan Verberne, and Zhaochun Ren. Openreward: Learning to reward long-form agentic tasks via reinforcement learning. *arXiv preprint arXiv:2510.24636*, 2025.

[87] Yuncheng Hua, Lizhen Qu, and Reza Haf. Assistive large language model agents for socially-aware negotiation dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8047–8074, 2024.

[88] Jinbang Huang, Zhiyuan Li, Yuanzhao Hu, Zhanguang Zhang, Mark Coates, Xingyue Quan, and Yingxue Zhang. Self-CriTeach: LLM self-teaching and self-critiquing for improving robotic planning via automated domain generation. In *ICLR 2026 Workshop on RSI*, 2026.

[89] Kun Huang, Weikai Xu, Yuxuan Liu, Quandong Wang, Pengzhi Gao, Wei Liu, Jian Luan, Bin Wang, and Bo An. Mobileipl: Enhancing mobile agents thinking process via iterative preference learning. *arXiv preprint arXiv:2505.12299*, 2025.

[90] Wei-Chieh Huang, Weizhi Zhang, Yueqing Liang, Yuanchen Bei, Yankai Chen, Tao Feng, Xinyu Pan, Zhen Tan, Yu Wang, Tianxin Wei, Shanglin Wu, Ruiyao Xu, Liangwei Yang, Rui Yang, Wooseong Yang, Chin-Yuan Yeh, Hanrong Zhang, Haozhen Zhang, Siqi Zhu, Henry Peng Zou, et al. Rethinking memory mechanisms of foundation agents in the second half: A survey. *arXiv preprint arXiv:2602.06052*, 2026.

[91] Xiang Huang, Jiayu Shen, Shanshan Huang, Sitao Cheng, Xiaxia Wang, and Yuzhong Qu. Targa: Targeted synthetic data generation for practical reasoning over structured data. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2704–2726, 2025.

[92] Xingyue Huang, Rishabh, Gregor Franke, Ziyi Yang, Jiamu Bai, Weijie Bai, Jinhe Bi, Zifeng Ding, Yiqun Duan, Chengyu Fan, Wendong Fan, Xin Gao, Ruohao Guo, Yuan He, Yicheng He, Xianglong Hu, Neil Johnson, Bowen Li, Fangru Lin, Siyu Lin, Tong Liu, Yunpu Ma, HAO SHEN, Hao Sun, Beibei Wang, Fangyijie Wang, Hao Wang, Haoran Wang, Yang Wang, Yifeng Wang, Zhaowei Wang, Ziyang Wang, Yifan Wu, Zikai Xiao, Chengxing Xie, Fan Yang, Junxiao Yang, Qianshuo Ye, Ziyu Ye, Guangtao Zeng, Yuwen Ebony Zhang, Zeyu Zhang, Zihao Zhu, Bernard Ghanem, Philip Torr, and Guohao Li. Loong: Synthesize long chain-of-thoughts at scale through verifiers. In *NeurIPS 2025 Workshop on Bridging Language, Agent, and World Models for Reasoning and Planning*, 2025. URL https://openreview.net/forum?id=UjOmDs3NrR.

[93] Yuchen Huang, Sijia Li, Minghao Liu, Wei Liu, Shijue Huang, Zhiyuan Fan, Hou Pong Chan, and Yi R Fung. Environment scaling for interactive agentic experience collection: A survey. *arXiv preprint arXiv:2511.09586*, 2025.

[94] Yupeng Huo, Yaxi Lu, Zhong Zhang, Haotian Chen, and Yankai Lin. Atommem : Learnable dynamic agentic memory with atomic memory operation. *arXiv preprint arXiv:2601.08323*, 2026.

[95] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, 2025.

[96] Naman Jain, Jaskirat Singh, Manish Shetty, Tianjun Zhang, Liang Zheng, Koushik Sen, and Ion Stoica. R2E-Gym: Procedural environment generation and hybrid verifiers for scaling open-weights SWE agents. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=7evvwwdo3z.

[97] Kyochul Jang, Donghyeon Lee, Kyusik Kim, Dongseok Heo, Taewhoo Lee, Woojeong Kim, and Bongwon Suh. DICE-BENCH: Evaluating the tool-use capabilities of large language models in multi-round, multi-party dialogues. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. URL https://aclanthology.org/2025.findings-acl.1375/.

[98] Seongbo Jang, Minjin Jeon, Jaehoon Lee, Seonghyeon Lee, Dongha Lee, and Hwanjo Yu. From what to respond to when to respond: Timely response generation for open-domain dialogue agents, 2025.

[99] 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 *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=cDYqckEt6d.

[100] Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for llm agent reinforcement learning. *arXiv preprint arXiv:2509.21240*, 2025.

[101] Xinyi Jiang, Tianyi Hu, Yuheng Qin, Guoming Wang, Zhou Huan, Kehan Chen, Gang Huang, Rongxing Lu, and Siliang Tang. ChatMap: Mining human thought processes for customer service chatbots via multi-agent collaboration. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11927–11947, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.617. URL https://aclanthology.org/2025.findings-acl.617/.

[102] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024.

[103] Song Jin, Shuqi Li, Shukun Zhang, and Rui Yan. Finrpt: Dataset, evaluation system and llm-based multi-agent framework for equity research report generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 507–515, 2026.

[104] Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang, Xinya Du, and Dong Yu. DSBench: How far are data science agents from becoming data science experts? In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=DSsSPr0RZJ.

[105] Da Ju, Song Jiang, Andrew Cohen, Aaron Foss, Sasha Mitts, Arman Zharmagambetov, Brandon Amos, Xian Li, Justine T Kao, Maryam Fazel-Zarandi, et al. To the globe (ttg): Towards language-driven guaranteed travel planning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 240–249, 2024.

[106] Yeonsung Jung, Trilok Padhi, Sina Shaham, Dipika Khullar, Joonhyun Jeong, Ninareh Mehrabi, and Eunho Yang. Co-evolving agents: Learning from failures as hard negatives. *arXiv preprint arXiv:2511.22254*, 2025.

[107] Zaid Khan, Elias Stengel-Eskin, Jaemin Cho, and Mohit Bansal. Dataenvgym: Data generation agents in teacher environments with student feedback. In *International Conference on Learning Representations*, volume 2025, pages 53480–53507, 2025.

[108] Namyoung Kim, Kai Tzu-iunn Ong, Yeonjun Hwang, Minseok Kang, Iiseo Jihn, Gayoung Kim, Minju Kim, and Jinyoung Yeo. PRINCIPLES: Synthetic strategy memory for proactive dialogue agents. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 21329–21368, 2025. doi: 10.18653/v1/2025.findings-emnlp.1164. URL https://aclanthology.org/2025.findings-emnlp.1164/.

[109] Satyapriya Krishna, Kalpesh Krichene, Anirudh Ramamurthy, Rajarshi Maheshwary, Gabriel Stanovsky, and Eunsol Choi. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. *arXiv preprint arXiv:2409.12941*, 2024.

[110] Canasai Kruengkrai and Koichiro Yoshino. Teaching text agents to learn sequential decision making from failure. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31619–31635, 2025.

[111] Ninad Kulkarni, Xian Carrie Wu, Siddharth Varia, and Dmitriy Bespalov. Agent vs. agent: Automated data generation and red-teaming for custom agentic workflows. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2025.

[112] Tharindu Kumarage, Ninareh Mehrabi, Anil Ramakrishna, Xinyan Zhao, Richard Zemel, Kai-Wei Chang, Aram Galstyan, Rahul Gupta, and Charith Peris. Towards safety reasoning in llms: Ai-agentic deliberation for policy-embedded cot data creation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22694–22715, 2025.

[113] Guangchen (Eric) Lan, Huseyin A. Inan, Sahar Abdelnabi, Janardhan Kulkarni, Lukas Wutschitz, Reza Shokri, Christopher Brinton, and Robert Sim. Contextual integrity in llms via reasoning and reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 104355–104391. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/968591afcfe4ba889eccffb6a4ca2d45-Paper-Conference.pdf.

[114] Tian Lan, Wenwei Zhang, Chengqi Lyu, Shuaibin Li, Chen Xu, Heyan Huang, Dahua Lin, Xian-Ling Mao, and Kai Chen. Training language models to critique with multi-agent feedback. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1474–1501, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.78. URL https://aclanthology.org/2025.findings-emnlp.78/.

[115] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

[116] Hwiwon Lee, Ziqi Zhang, Hanxiao Lu, and Lingming Zhang. Sec-bench: Automated benchmarking of llm agents on real-world software security tasks. *arXiv preprint arXiv:2506.11791*, 2025.

[117] Jihyun Lee, Yejin Min, San Kim, Yejin Jeon, Sung Jun Yang, Hyounghun Kim, and Gary Lee. Panictocalm: A proactive counseling agent for panic attacks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12853–12885, 2025.

[118] Young-Suk Lee, Md Arafat Sultan, Yousef El-Kurdi, Tahira Naseem Muñoz, Radu Florian, Salim Roukos, and Ramón Fernandez Astudillo. Ensemble-instruct: Instruction tuning data generation with a heterogeneous mixture of LMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

[119] Yujeong Lee, Sangwoo Shin, Wei-Jin Park, and Honguk Woo. LLM-based offline learning for embodied agents via consistency-guided reward ensemble. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3006–3029, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.170. URL https://aclanthology.org/2024.findings-emnlp.170/.

[120] Yongqi Leng, Yikun Lei, Xikai Liu, Meizhi Zhong, Bojian Xiong, Yurong Zhang, Yan Gao, Yiwu, Yao Hu, and Deyi Xiong. DecEx-RAG: Boosting agentic retrieval-augmented generation with decision and execution optimization via process supervision. In Saloni Potdar, Lina Rojas-Barahona, and Sebastien Montella, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1412–1425, Suzhou (China), November 2025. Association for Computational Linguistics. ISBN 979-8-89176-333-3. doi: 10.18653/v1/2025.emnlp-industry.99. URL https://aclanthology.org/2025.emnlp-industry.99/.

[121] Junlong Li, Wenshuo Zhao, Jian Zhao, Weihao Zeng, Haoze Wu, Xiaochen Wang, Rui Ge, Yuxuan Cao, Yuzhen Huang, Wei Liu, Junteng Liu, Zhaochen Su, Yiyang Guo, Fan Zhou, Lueyang Zhang, Juan Michelini, Xingyao Wang, Xiang Yue, Shuyan Zhou, Graham Neubig, and Junxian He. The Tool Decathlon: Benchmarking language agents for diverse, realistic, and long-horizon task execution. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.25726.

[122] Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Yida Zhao, Liwen Zhang, Litu Ou, Dingchu Zhang, Xixi Wu, Jialong Wu, et al. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. *arXiv preprint arXiv:2509.13305*, 2025.

[123] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. WebSailor: Navigating super-human reasoning for web agent, 2025.

[124] Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-Bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. URL https://aclanthology.org/2023.emnlp-main.187/.

[125] Muzhi Li, Jinhu Qi, Yihong Wu, Minghao Zhao, Liheng Ma, Yifan Li, Xinyu Wang, Yingxue Zhang, Ho fung Leung, and Irwin King. From evidence to trajectory: Abductive reasoning path synthesis for retrieval-augmented generation agents development. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025. URL https://openreview.net/forum?id=wt5EF9bAqb.

[126] Pengxiang Li, Zhi Gao, Bofei Zhang, Yapeng Mi, Xiaojian Ma, Chenrui Shi, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Iterative tool usage exploration for multimodal agents via step-wise preference tuning. *arXiv preprint arXiv:2504.21561*, 2025.

[127] Wenjun Li, Changyu Chen, and Pradeep Varakantham. Unlocking the planning capabilities of large language models with maximum diversity fine-tuning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3318–3340, 2025.

[128] Xinze Li, Sen Mei, Zhenghao Liu, Yukun Yan, Shuo Wang, Shi Yu, Zheni Zeng, Hao Chen, Ge Yu, Zhiyuan Liu, Maosong Sun, and Chenyan Xiong. RAG-DDR: Optimizing retrieval-augmented generation using differentiable data rewards, 2024.

[129] Xinzhe Li. A review of prominent paradigms for LLM-based agents: Tool use, planning (including RAG), and feedback learning. In *Proceedings of the 31st international conference on computational linguistics*, pages 9760–9779, 2025.

[130] Xirui Li, Ming Li, Derry Xu, Wei-Lin Chiang, Ion Stoica, Cho-Jui Hsieh, and Tianyi Zhou. Clawenvkit: Automatic environment generation for claw-like agents. *arXiv preprint arXiv:2604.18543*, 2026.

[131] Yifei Li, Hanane Nour Moussa, Ziru Chen, Shijie Chen, Botao Yu, Mingyi Xue, Benjamin Burns, Tzu-Yao Chiu, Vishal Dey, Zitong Lu, Chen Wei, Qianheng Zhang, Tianyu Zhang, Song Gao, Xuhui Huang, Xia Ning, Nesreen K. Ahmed, Ali Payani, and Huan Sun. AutoSDT: Scaling data-driven discovery tasks toward open co-scientists. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

[132] Yu Li, Shenyu Zhang, Rui Wu, Xiutian Huang, Yongrui Chen, Wenhao Xu, Guilin Qi, and Dehai Min. Mateval: A multi-agent discussion framework for advancing open-ended text evaluation. In Makoto Onizuka, Jae-Gil Lee, Yongxin Tong, Chuan Xiao, Yoshiharu Ishikawa, Sihem Amer-Yahia, H. V. Jagadish, and Kejing Lu, editors, *Database Systems for Advanced Applications*, pages 415–426, Singapore, 2024. Springer Nature Singapore. ISBN 978-981-97-5575-2.

[133] Yu Li, Guangfeng Cai, Shengtian Yang, Han Luo, Shuo Han, Xu He, Dong Li, and Lei Feng. Phgpo: Pheromone-guided policy optimization for long-horizon tool planning. *arXiv preprint arXiv:2602.13691*, 2026.

[134] Yuetai Li, Huseyin A. Inan, Xiang Yue, Wei-Ning Chen, Lukas Wutschitz, Janardhan Kulkarni, Radha Poovendran, Robert Sim, and Saravan Rajmohan. Simulating Environments with Reasoning Models for Agent Training, November 2025. URL http://arxiv.org/abs/2511.01824. arXiv:2511.01824 [cs].

[135] Zhigen Li, Jianxiang Peng, Yanmeng Wang, Yong Cao, Tianhao Shen, Minghui Zhang, Linxi Su, Shang Wu, Yihang Wu, Yuqian Wang, et al. Chatsop: An sop-guided mcts planning framework for controllable llm dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17637–17659, 2025.

[136] Qiqiang Lin, Muning Wu, Zhibin Zhao, Miao Li, Yanchao Wang, Shanshan Cai, Xiaolei Yuan, Tao Wang, Ziming Fu, Yizhi Yuan, Bang Han, Jian Gao, Wen Zhao, Haoda Yin, Xiaoyun Zhu, Qingwei Wang, Dacheng Ding, Pengyu Liu, Yun Ye, Bing Wang, and Jianye Zheng. Hammer: Robust function-calling for on-device language models via function masking, 2024.

[137] Zongyu Lin, Yao Tang, Xingcheng Yao, Da Yin, Ziniu Hu, Yizhou Sun, and Kai-Wei Chang. QLASS: Boosting language agent inference via q-guided stepwise search. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=f6lio2CZIM.

[138] Grace Liu, Yuxiao Qu, Jeff Schneider, Aarti Singh, and Aviral Kumar. CaRT: Teaching LLM agents to know when they know enough. *arXiv preprint arXiv:2510.08517*, 2025.

[139] Jiateng Liu, Zhenhailong Wang, Xiaojiang Huang, Yingjie Li, Xing Fan, Xiang Li, Chenlei Guo, Ruhi Sarikaya, and Heng Ji. Analyzing and internalizing complex policy documents for LLM agents, 2025.

[140] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[141] Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, Jiayuan Song, Zhengmao Zhu, Wenhu Chen, Pengyu Zhao, and Junxian He. WebExplorer: Explore and evolve for training long-horizon web agents, 2025. URL https://arxiv.org/abs/2509.06501. arXiv preprint arXiv:2509.06501.

[142] Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, et al. Toolace: Winning the points of llm function calling. *arXiv preprint arXiv:2409.00920*, 2024.

[143] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xingjian Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *The Twelfth International Conference on Learning Representations*, 2024.

[144] Yitao Liu, Chenglei Si, Karthik R. Narasimhan, and Shunyu Yao. Contextual experience replay for self-improvement of language agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14179–14198, 2025. doi: 10.18653/v1/2025.acl-long.694. URL https://aclanthology.org/2025.acl-long.694/.

[145] Yue Liu et al. COAST: Enhancing the code debugging ability of LLMs through communicative agent based data synthesis. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics*, 2025.

[146] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[147] Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, 2024.

[148] Haocheng Lu, Minjun Zhu, and Henry Yu. Hard negative sample-augmented DPO post-training for small language models. *arXiv preprint arXiv:2512.19728*, 2025.

[149] Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025. URL https://aclanthology.org/2025.findings-naacl.65/.

[150] Yuxuan Lu, Ziyi Wang, Yingzhou Lu, Yisi Sang, Jiri Gesi, Xianfeng Tang, Yimeng Zhang, Zhenwei Dai, Hui Liu, Hanqing Lu, Chen Luo, Qi He, Benoit Dumoulin, Jing Huang, and Dakuo Wang. Firefly: Illuminating large-scale verified tool-call data generation from real apis, 2026. URL https://arxiv.org/abs/2605.17558.

[151] Zimu Lu, Aojun Zhou, Ke Wang, Houxing Ren, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. Mathcoder2: Better math reasoning from continued pretraining on model-translated mathematical code. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=1Iuw1jcIrf.

[152] Haoran Luo, Haihong E, Yikai Guo, Qika Lin, Xiaobao Wu, Xinyu Mu, Wenhao Liu, Meina Song, Yifan Zhu, and Luu Anh Tuan. KBQA-o1: Agentic knowledge base question answering with Monte Carlo Tree Search, 2025. URL https://arxiv.org/abs/2501.18922. arXiv preprint arXiv:2501.18922.

[153] Haotian Luo, Huaisong Zhang, Xuelin Zhang, Haoyu Wang, Zeyu Qin, Wenjie Lu, Guozheng Ma, Haiying He, Yingsha Xie, Qiyang Zhou, Zixuan Hu, Hongze Mi, Yibo Wang, Naiqiang Tan, Hong Chen, Yi R. Fung, Chun Yuan, and Li Shen. UltraHorizon: Benchmarking agent capabilities in ultra long-horizon scenarios. *arXiv preprint arXiv:2509.21766*, 2025.

[154] Man Luo, David Cobbley, Xin Su, Shachar Rosenman, Vasudev Lal, Shao-Yen Tseng, and Phillip Howard. Dpo learning with llms-judge signal for computer use agents, 2025. URL https://arxiv.org/abs/2506.03095.

[155] Ziyang Luo, Zhiqi Shen, Wenzhuo Yang, Zirui Zhao, Prathyusha Jwalapuram, Amrita Saha, Doyen Sahoo, Silvio Savarese, Caiming Xiong, and Junnan Li. MCP-Universe: Benchmarking large language models with real-world model context protocol servers, 2025. URL https://arxiv.org/abs/2508.14704.

[156] Yuanjie Lyu, Chengyu Wang, Jun Huang, and Tong Xu. From correction to mastery: Reinforced distillation of large language model agents. *arXiv preprint arXiv:2509.14257*, 2025.

[157] Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn llm agents. *Advances in neural information processing systems*, 37:74325–74362, 2024.

[158] Zhiming Ma, Peidong Wang, Minhua Huang, Jinpeng Wang, Kai Wu, Xiangzhao Lv, Yachun Pang, Yin Yang, Wenjie Tang, and Yuchen Kang. Teleantifraud-28k: An audio-text slow-thinking dataset for telecom fraud detection. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 5853–5862, 2025.

[159] Navid Madani and Rohini K Srihari. Esc-judge: A framework for comparing emotional support conversational agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16059–16076, 2025.

[160] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, 2024.

[161] Ethan Mendes and Alan Ritter. Language models can self-improve at state-value estimation for better search. *arXiv preprint arXiv:2503.02878*, 2025.

[162] Dehai Min, Nan Hu, Rihui Jin, Nuo Lin, Jiaoyan Chen, Yongrui Chen, Yu Li, Guilin Qi, Yun Li, Nijun Li, and Qianren Wang. Exploring the impact of table-to-text methods on augmenting LLM-based question answering with domain hybrid data. In Yi Yang, Aida Davani, Avi Sil, and Anoop Kumar, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 464–482, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-industry.41. URL https://aclanthology.org/2024.naacl-industry.41/.

[163] Dehai Min, Zhiyang Xu, Guilin Qi, Lifu Huang, and Chenyu You. Unihgkr: unified instruction-aware heterogeneous knowledge retrievers. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4577–4594, 2025.

[164] Dehai Min, Kailin Zhang, Tongtong Wu, and Lu Cheng. Quco-rag: Quantifying uncertainty from the pre-training corpus for dynamic retrieval-augmented generation. *arXiv preprint arXiv:2512.19134*, 2025.

[165] Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Codas, Yadong Lu, Wei-Ge Chen, Olga Vrousgos, Corby Rosset, Fillipe Silva, Hamed Khanpour, Yazan Lara, and Ahmed Awadallah. AgentInstruct: Toward generative teaching with agentic flows, 2024.

[166] Guozhao Mo, Wenliang Zhong, Jiawei Chen, Qianhao Yuan, Xuanang Chen, Yaojie Lu, Hongyu Lin, Ben He, Xianpei Han, and Le Sun. LiveMCPBench: Can agents navigate an ocean of MCP tools?, 2025. URL https://arxiv.org/abs/2508.01780.

[167] Niels Mündler, Mark Niklas Mueller, Jingxuan He, and Martin Vechev. SWT-Bench: Testing and validating real-world bug-fixes with code agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[168] Shikhar Murty, Christopher D. Manning, Peter Shaw, Mandar Joshi, and Kenton Lee. BAGEL: Bootstrapping agents by guiding exploration with language. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 36894–36910, 2024. URL https://proceedings.mlr.press/v235/murty24a.html.

[169] Shikhar Murty, Hao Zhu, Dzmitry Bahdanau, and Christopher D. Manning. NNetNav: Unsupervised learning of browser agents through environment interaction in the wild, 2024. URL https://arxiv.org/abs/2410.02907.

[170] Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Mikhail Plekhanov, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, Dieuwke Hupkes, Ricardo Silveira Cabral, Tatiana Shavrina, Jakob Nicolaus Foerster, Yoram Bachrach, William Yang Wang, and Roberta Raileanu. MLGym: A new framework and benchmark for advancing AI research agents. In *Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=ryTr83DxRq.

[171] Ansong Ni, Ruta Desai, Yang Li, Xinjie Lei, Dong Wang, Jiemin Zhang, Jane Yu, Ramya Raghavendra, Gargi Ghosh, Shang-Wen Li, et al. Collaborative reasoner: Self-improving social agents with synthetic conversations. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[172] Yansong Ning and Hao Liu. UrbanKGent: A unified large language model agent framework for urban knowledge graph construction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Nycj81Z692.

[173] Yansong Ning, Rui Liu, Jun Wang, Kai Chen, Wei Li, Jun Fang, Kan Zheng, Naiqiang Tan, and Hao Liu. Deeptravel: An end-to-end agentic reinforcement learning framework for autonomous travel planning agents. *arXiv preprint arXiv:2509.21842*, 2025.

[174] Kai Tzu-iunn Ong, Namyoung Kim, Minju Gwak, Hyungjoo Chae, Taeyoon Kwon, Yohan Jo, Seung-won Hwang, Dongha Lee, and Jinyoung Yeo. Towards lifelong dialogue agents via timeline-based memory management. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8631–8661, 2025. doi: 10.18653/v1/2025.naacl-long.435. URL https://aclanthology.org/2025.naacl-long.435/.

[175] Odhran O'Donoghue, Aleksandar Shtedritski, John Ginger, Ralph Abboud, Ali Ghareeb, and Samuel Rodriques. Bioplanner: automatic evaluation of llms on protocol planning in biology. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2676–2694, 2023.

[176] Trilok Padhi, Pinxian Lu, Abdulkadir Erol, Tanmay Sutar, Gauri Sharma, Mina Sonmez, Munmun De Choudhury, and Ugur Kursuncu. Echoes of human malice in agents: Benchmarking LLMs for multi-turn online harassment attacks, 2025. URL https://arxiv.org/abs/2510.14207.

[177] Vishal Pallagani, Bharath Muppasani, Keerthiram Murugesan, Francesca Rossi, Lior Horesh, Biplav Srivastava, Francesco Fabiano, and Andrea Loreggia. Plansformer: Generating symbolic plans using transformers. *arXiv preprint arXiv:2212.08681*, 2022.

[178] Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. Training software engineering agents and verifiers with SWE-Gym. *arXiv preprint arXiv:2412.21139*, 2024.

[179] Shrey Pandit, Xuan-Phi Nguyen, Yifei Ming, Austin Xu, Jiayu Wang, Caiming Xiong, and Shafiq Joty. Synthesizing agentic data for web agents with progressive difficulty enhancement mechanisms, 2025. URL https://arxiv.org/abs/2510.13913.

[180] Deep Anil Patel, Iain Melvin, Christopher Malon, and Martin Renqiang Min. Discussllm: Teaching large language models when to speak. *arXiv preprint arXiv:2508.18167*, 2025.

[181] Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive APIs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e4c61f578ff07830f5c37378dd3ecb0d-Abstract-Conference.html.

[182] Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 48371–48392, 2025.

[183] Tim Pearce, Tabish Rashid, David Bignell, Raluca Georgescu, Sam Devlin, and Katja Hofmann. Scaling laws for pre-training agents and world models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=HHwGfLOKxq.

[184] Hoang Pham, Thuy-Duong Nguyen, and Khac-Hoai Nam Bui. Agent-UniRAG: A trainable open-source LLM agent framework for unified retrieval-augmented generation systems, 2025.

[185] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[186] Gabriel Poesia, David Broman, Nick Haber, and Noah D. Goodman. Learning formal mathematics from intrinsic motivation. In *Advances in Neural Information Processing Systems*, 2024.

[187] Thomas Pouplin, Kasia Kobalczyk, Hao Sun, and Mihaela van der Schaar. The synergy of LLMs & RL unlocks offline learning of generalizable language-conditioned policies with low-fidelity data. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=5hyfZ2jYfI.

[188] Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Manoj Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, et al. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay. *Advances in Neural Information Processing Systems*, 38, 2026.

[189] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

[190] Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, Caiming Xiong, and Huan Wang. UserBench: An interactive gym environment for user-centric agents, 2025. URL https://arxiv.org/abs/2507.22034.

[191] Rushi Qiang, Yuchen Zhuang, Anikait Singh, Percy Liang, Chao Zhang, Sherry Yang, and Bo Dai. MLE-Smith: Scaling MLE tasks with automated multi-agent pipeline, 2025.

[192] Shuofei Qiao, Yanqiu Zhao, Zhisong Qiu, Xiaobin Wang, Jintian Zhang, Zhao Bin, Ningyu Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Scaling generalist data-analytic agents, 2025.

[193] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *ICLR*, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

[194] Huachuan Qiu and Zhenzhong Lan. Psydial: A large-scale long-term conversational dataset for mental health support. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21624–21655, 2025.

[195] Pengcheng Qiu, Chaoyi Wu, Junwei Liu, Qiaoyu Zheng, Yusheng Liao, Haowen Wang, Yun Yue, Qianrui Fan, Shuai Zhen, Jian Wang, et al. Evolving diagnostic agents in a virtual clinical environment. *arXiv preprint arXiv:2510.24654*, 2025.

[196] Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. Tool learning with large language models: A survey. *Frontiers of Computer Science*, 19(8):198343, 2025.

[197] Yincen Qu, Huan Xiao, Feng Li, Gregory Li, Hui Zhou, Xiangying Dai, and Xiaoru Dai. Tripscore: Benchmarking and rewarding real-world travel planning with fine-grained evaluation. *arXiv preprint arXiv:2510.09011*, 2025.

[198] Melissa Kazemi Rad, Alberto Purpura, Himanshu Kumar, Emily Chen, and Mohammad Shahed Sorower. GRAID: Synthetic data generation with geometric constraints and multi-agentic reflection for harmful content detection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 30047–30065, 2025.

[199] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[200] Salman Rahman, Liwei Jiang, James Shiffer, Genglin Liu, Sheriff Issaka, Md Rizwan Parvez, Hamid Palangi, Kai-Wei Chang, Yejin Choi, and Saadia Gabriel. X-teaming: Multi-turn jailbreaks and defenses with adaptive multi-agents. *arXiv preprint arXiv:2504.13203*, 2025.

[201] Yiting Ran, Xintao Wang, Rui Xu, Xinfeng Yuan, Jiaqing Liang, Yanghua Xiao, and Deqing Yang. Capturing minds, not just words: Enhancing role-playing language models with personality-indicative data. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14566–14576, 2024.

[202] Manik Rana, Calissa Man, Anotida Expected Msiiwa, Jeffrey Paine, Kevin Zhu, Sunishchal Dev, Vasu Sharma, and Ahan M R. AgentChangeBench: A multi-dimensional evaluation framework for goal-shift robustness in conversational AI, 2025. URL https://arxiv.org/abs/2510.18170. NeurIPS 2025 Workshop on Multi-Turn Interactions.

[203] Haris Riaz, Sourav Sanjukta Bhabesh, Vinayak Arannil, Miguel Ballesteros, and Graham Horwood. MetaSynth: Meta-prompting-driven agentic scaffolds for diverse synthetic data generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18770–18803, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.962. URL https://aclanthology.org/2025.findings-acl.962/.

[204] Willem Röpke, Samuel Coward, Andrei Lupu, Thomas Foster, Tim Rocktäschel, and Jakob Nicolaus Foerster. Déjàq: Open-ended evolution of diverse, learnable and verifiable problems, 2026. URL https://openreview.net/forum?id=ONfuqbOys3.

[205] Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of LM agents with an LM-emulated sandbox. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=GEcwtMk1uA. Spotlight.

[206] Swarnadeep Saha, Archiki Prasad, Justin Chih-Yao Chen, Peter Hase, Elias Stengel-Eskin, and Mohit Bansal. System-1. x: Learning to balance fast and slow planning with language models. *arXiv preprint arXiv:2407.14414*, 2024.

[207] Vinay Samuel, Henry Peng Zou, Yue Zhou, Shreyas Chaudhari, Ashwin Kalyan, Tanmay Rajpurohit, Ameet Deshpande, Karthik Narasimhan, and Vishvak Murahari. PersonaGym: Evaluating persona agents and LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025. URL https://aclanthology.org/2025.findings-emnlp.368/.

[208] Vishnu Sarukkai, Zhiqiang Xie, and Kayvon Fatahalian. Self-generated in-context examples improve LLM agents for sequential decision-making tasks. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML 2025*, 2025. URL https://openreview.net/forum?id=YurjMGGTTj.

[209] Zhiyu Shen, Ziming Wu, Fuming Lai, Shaobing Lian, and Yanghui Rao. Membuilder: Reinforcing llms for long-term memory construction via attributed dense rewards. *arXiv preprint arXiv:2601.05488*, 2026.

[210] Dingfeng Shi, Jingyi Cao, Qianben Chen, Weichen Sun, Weizhen Li, Hongxuan Lu, Fangchen Dong, Tianrui Qin, King Zhu, Minghao Liu, Yuchen Eleanor Jiang, Jian Yang, Ge Zhang, Jiaheng Liu, Changwang Zhang, Jun Wang, and Wangchunshu Zhou. Taskcraft: Automated generation of agentic tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=UJFCyrYM1V.

[211] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

[212] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *Nature*, 631(8021):592–598, 2024.

[213] Maojia Song, Renhang Liu, Xinyu Wang, Yong Jiang, Pengjun Xie, Fei Huang, Jingren Zhou, Dorien Herremans, and Soujanya Poria. Demystifying deep search: A holistic evaluation with hint-free multi-hop questions and factorised metrics, 2025. URL https://arxiv.org/abs/2510.05137. arXiv preprint arXiv:2510.05137.

[214] Xiaoshuai Song, Haofei Chang, Guanting Dong, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Envscaler: Scaling tool-interactive environments for llm agent via programmatic synthesis. *arXiv preprint arXiv:2601.05808*, 2026.

[215] Yifan Song, Weimin Xiong, Xiutian Zhao, Dawei Zhu, Wenhao Wu, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. Agentbank: Towards generalized llm agents via fine-tuning on 50000+ interaction trajectories. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2124–2141, 2024.

[216] Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. Trial and error: Exploration-based trajectory optimization of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7584–7600. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.409. URL https://aclanthology.org/2024.acl-long.409/.

[217] Atharv Sonwane, Isadora White, Hyunji Lee, Matheus Pereira, Lucas Caccia, Minseon Kim, Zhengyan Shi, Chinmay Singh, Alessandro Sordoni, Marc-Alexandre Côté, and Xingdi Yuan. BugPilot: Complex bug generation for efficient learning of SWE skills, 2025.

[218] Sakhinana Sagar Srinivas, Shivam Gupta, and Venkataramana Runkana. AutoChemSchematic AI: Agentic physics-aware automation for chemical manufacturing scale-up, 2025.

[219] Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research, 2025. URL https://arxiv.org/abs/2504.01848.

[220] Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. LACIE: Listener-aware finetuning for calibration in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=RnvgYd9RAh.

[221] Hongjin SU, Ruoxi Sun, Jinsung Yoon, Pengcheng Yin, Tao Yu, and Sercan O Arik. Learn-by-interact: A data-centric framework for self-adaptive agents in realistic environments. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=3UKOzGWCVY.

[222] Liangcai Su, Zhen Zhang, Guangyu Li, Zhuo Chen, Chenxi Wang, Maojia Song, Xinyu Wang, Kuan Li, Jialong Wu, Xuanzhong Chen, et al. Scaling agents via continual pre-training. *arXiv preprint arXiv:2509.13310*, 2025.

[223] Michael Sullivan, Mareike Hartmann, and Alexander Koller. Procedural Environment Generation for Tool-Use Agents, 2025. URL https://arxiv.org/abs/2506.11045. Version Number: 2.

[224] Chenkai Sun, Denghui Zhang, ChengXiang Zhai, and Heng Ji. Beyond reactive safety: Risk-aware llm alignment via long-horizon simulation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6422–6434, 2025.

[225] Fan-Yun Sun, SI Harini, Angela Yi, Yihan Zhou, Alex Zook, Jonathan Tremblay, Logan Cross, Jiajun Wu, and Nick Haber. Factorsim: Generative simulation via factorized representation. *Advances in Neural Information Processing Systems*, 37:87438–87472, 2024.

[226] Yu Sun, Xingyu Qian, Weiwen Xu, Hao Zhang, Chenghao Xiao, Long Li, Deli Zhao, Wenbing Huang, Tingyang Xu, Qifeng Bai, and Yu Rong. ReasonMed: A 370K multi-agent generated dataset for advancing medical reasoning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26446–26467, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1344. URL https://aclanthology.org/2025.emnlp-main.1344/.

[227] Ivan Sviridov, Amina Miftakhova, Tereshchenko Artemiy Vladimirovich, Galina Zubkova, Pavel Blinov, and Andrey Savchenko. 3mdbench: Medical multimodal multi-agent dialogue benchmark. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26625–26665, 2025.

[228] Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8416–8439, 2025.

[229] Chenming Tang, Hsiu-Yuan Huang, Weijie Liu, Junqiang Zheng, Saiyong Yang, and Yunfang Wu. Tool learning needs nothing more than a free 8b language model. *arXiv preprint arXiv:2604.17739*, 2026.

[230] Xiangru Tang, Chunyuan Deng, Hanminwang Hanminwang, Haoran Wang, Yilun Zhao, Wenqi Shi, Yi Fung, Wangchunshu Zhou, Jiannan Cao, Heng Ji, et al. Mimir: A customizable agent tuning platform for enhanced scientific applications. In *Proceedings of the 2024 Conference on empirical methods in natural language processing: system demonstrations*, pages 486–496, 2024.

[231] Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

[232] Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. SciCode: A research coding benchmark curated by scientists. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=SbmB5xjtiN.

[233] Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, Kuan Li, Liangcai Su, Litu Ou, Liwen Zhang, Pengjun Xie, Rui Ye, Wenbiao Yin, Xinmiao Yu, Xinyu Wang, Xixi Wu, Xuanzhong Chen, Yida Zhao, Zhen Zhang, Zhengwei Tao, Zhongwang Zhang, Zile Qiao, Chenxi Wang, Donglei Yu, Gang Fu, Haiyang Shen, Jiayin Yang, Jun Lin, Junkai Zhang, Kui Zeng, Li Yang, Hailong Yin, Maojia Song, Ming Yan, Minpeng Liao, Peng Xia, Qian Xiao, Rui Min, Ruixue Ding, Runnan Fang, Shaowei Chen, Shen Huang, Shihang Wang, Shihao Cai, Weizhou Shen, Xiaobin Wang, Xin Guan, Xinyu Geng, Yingcheng Shi, Yuning Wu, Zhuo Chen, Zijian Li, and Yong Jiang. Tongyi DeepResearch Technical Report, 2025. URL https://arxiv.org/abs/2510.24701. Version Number: 2.

[234] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics (TACL)*, 2022.

[235] Dunwei Tu, Hongyan Hao, Hansi Yang, Yihao Chen, Yi-Kai Zhang, Zhikang Xia, Yu Yang, Yueqing Sun, Xingchen Liu, Furao Shen, Qi Gu, Hui Su, and Xunliang Cai. Scaleenv: Scaling environment synthesis from scratch for generalist interactive tool-use agent training, 2026. URL https://arxiv.org/abs/2602.06820.

[236] Quan Tu, Shilong Fan, Zihang Tian, Tianhao Shen, Shuo Shang, Xin Gao, and Rui Yan. CharacterEval: A chinese benchmark for role-playing conversational agent evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024. URL https://aclanthology.org/2024.acl-long.638/.

[237] Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36:38975–38987, 2023.

[238] David Venuto, Mohammad Sami Nur Islam, Martin Klissarov, Doina Precup, Sherry Yang, and Ankit Anand. Code as reward: Empowering reinforcement learning with VLMs. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=6P88DMUDvH.

[239] Pulkit Verma, Ngoc La, Anthony Favier, Swaroop Mishra, and Julie A Shah. Teaching llms to plan: Logical chain-of-thought instruction tuning for symbolic planning. *arXiv preprint arXiv:2509.13351*, 2025.

[240] Harsh Vishwakarma, Ankush Agarwal, Ojas Patil, Chaitanya Devaguptapu, and Mahesh Chandran. Can llms help you at work? a sandbox for evaluating llm agents in enterprise environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9178–9212, 2025.

[241] Qian Wan, Wangzi Shi, Jintian Feng, Shengyingjie Liu, Luona Wei, Zhicheng Dai, and Jianwen Sun. Empowering math problem generation and reasoning for large language model via synthetic data based continual learning framework. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23972–23991, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1223. URL https://aclanthology.org/2025.emnlp-main.1223/.

[242] Tianxi Wan, Jiaming Luo, Siyuan Chen, Kunyao Lan, Jianhua Chen, Haiyang Geng, and Mengyue Wu. From medical records to diagnostic dialogues: A clinical-grounded approach and dataset for psychiatric comorbidity. *arXiv preprint arXiv:2510.25232*, 2025.

[243] Boshi Wang, Hao Fang, Jason Eisner, Benjamin Van Durme, and Yu Su. LLMs in the imaginarium: Tool learning through simulated trial and error. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10583–10604, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.570. URL https://aclanthology.org/2024.acl-long.570/.

[244] Hanlin Wang, Jian Wang, Chak Tou Leong, and Wenjie Li. STeCa: Step-level trajectory calibration for LLM agent learning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11597–11614, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.604. URL https://aclanthology.org/2025.findings-acl.604/.

[245] Haoran Wang, Zhenyu Hou, Yao Wei, Jie Tang, and Yuxiao Dong. Swe-dev: Building software engineering agents with training and inference scaling. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3742–3761, 2025.

[246] Huisheng Wang, Zhuoshi Pan, Hangjing Zhang, Mingxiao Liu, Hanqing Gao, and H Vicky Zhao. Investalign: Overcoming data scarcity in aligning large language models with investor decision-making processes under herd behavior. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10021–10052, 2025.

[247] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[248] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, 2024.

[249] Qi Wang, Hongzhi Zhang, Jia Fu, Kai Fu, Yahui Liu, Tinghai Zhang, Chenxi Sun, Gangwei Jiang, Jingyi Tang, Xingguang Ji, Yang Yue, Jingyuan Zhang, Fuzheng Zhang, Kun Gai, and Guorui Zhou. Klear-AgentForge: Forging Agentic Intelligence through Posttraining Scaling, 2025. URL https://arxiv.org/abs/2511.05951. Version Number: 1.

[250] Renxi Wang, Xudong Han, Lei Ji, Shu Wang, Timothy Baldwin, and Haonan Li. Toolgen: Unified tool retrieval and calling via generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XLMAMmowdY.

[251] Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=jJ9BoXAfFa.

[252] Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Yuan, Jiayi Pan, Yueqi Zhang, Neil Hessel, Erik Schluntz, et al. OpenHands: An open platform for AI software developers as generalist agents. In *International Conference on Learning Representations (ICLR)*, 2025.

[253] Xinyu Wang, Mingze Li, Peng Lu, Xiao-Wen Chang, Lifeng Shang, Jinping Li, Fei Mi, Prasanna Parthasarathi, and Yufei Cui. Infmem: Learning system-2 memory control for long-context agent. *arXiv preprint arXiv:2602.02704*, 2026.

[254] Yinjie Wang, Ling Yang, Ye Tian, Ke Shen, and Mengdi Wang. Co-evolving llm coder and unit tester via reinforcement learning. *arXiv preprint arXiv:2506.03136*, 2025.

[255] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.

[256] Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-{\alpha}: Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*, 2025.

[257] Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Stephen W. Huang, Jie Fu, and Junran Peng. RoleLLM: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. URL https://aclanthology.org/2024.findings-acl.878/.

[258] Zhaoyang Wang, Canwen Xu, Boyi Liu, Yite Wang, Siwei Han, Zhewei Yao, Huaxiu Yao, and Yuxiong He. Agent world model: Infinity synthetic environments for agentic reinforcement learning. *arXiv preprint arXiv:2602.10090*, 2026.

[259] Zhenting Wang, Qi Chang, Hemani Patel, Shashank Biju, Cheng-En Wu, Quan Liu, Aolin Ding, Alireza Rezazadeh, Ankit Shah, Yujia Bao, and Eugene Siow. MCP-Bench: Benchmarking tool-using LLM agents with complex real-world tasks via MCP servers. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fe8mzHwMxN.

[260] Zihan Wang, Siyao Liu, Yang Sun, Hongyan Li, and Kai Shen. CodeContests+: High-quality test case generation for competitive programming, 2025. URL https://arxiv.org/abs/2506.05817. arXiv preprint arXiv:2506.05817.

[261] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[262] Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

[263] Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. BrowseComp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

[264] Yifan Wei, Li Du, Xiaoyan Yu, Yang Feng, and Angsheng Li. Towards compositional generalization of llms via skill taxonomy guided data synthesis, 2026. URL https://arxiv.org/abs/2601.03676.

[265] Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I. Wang. SWE-RL: Advancing LLM reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025.

[266] Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. Cycleresearcher: Improving automated research via automated review. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bjcsVLoHYs.

[267] Hjalmar Wijk, Tao Roa Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Joshua M. Clymer, Jai Dhyani, Elena Ericheva, Katharyn Garcia, Brian Goodrich, Nikola Jurkovic, Megan Kinniment, Aron Lajko, Seraphina Nix, Lucas Jun Koba Sato, William Saunders, Maksym Taran, Ben West, and Elizabeth Barnes. RE-Bench: Evaluating frontier AI r&d capabilities of language model agents against human experts. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pages 66772–66832, 2025. URL https://proceedings.mlr.press/v267/wijk25a.html.

[268] Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*, 2024.

[269] Dingbang Wu, Rui Hao, Haiyang Wang, Shuzhe Wu, Han Xiao, Zhenghong Li, Bojiang Zhou, Zheng Ju, Zichen Liu, Lue Fan, et al. Mobilegym: A verifiable and highly parallel simulation platform for mobile gui agent research. *arXiv preprint arXiv:2605.26114*, 2026.

[270] Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. WebWalker: Benchmarking LLMs in web traversal. *arXiv preprint arXiv:2501.07572*, 2025.

[271] Mengsong Wu, YaFei Wang, Yidong Ming, Yuqi An, Yuwei Wan, Wenliang Chen, Binbin Lin, Yuqiang Li, Tong Xie, and Dongzhan Zhou. Chematagent: Enhancing llms for chemistry and materials science through tree-search based tool learning, 2025. URL https://arxiv.org/abs/2506.07551.

[272] Xixi Wu, Qianguo Sun, Ruiyang Zhang, Chao Song, Junlong Wu, Yiyan Qi, and Hong Cheng. Demystifying reinforcement learning for long-horizon tool-using agents: A comprehensive recipe. *arXiv preprint arXiv:2603.21972*, 2026.

[273] Yiran Wu, Mauricio Velazco, Andrew Zhao, Manuel Raúl Meléndez Luján, Srisuma Movva, Yogesh K Roy, Quang Nguyen, Roberto Rodriguez, Qingyun Wu, Michael Albada, et al. Excytin-bench: Evaluating llm agents on cyber threat investigation. *arXiv preprint arXiv:2507.14201*, 2025.

[274] Yue Wu, Yewen Fan, Paul Pu Liang, Amos Azaria, Yuanzhi Li, and Tom M Mitchell. Read and reap the rewards: Learning to play atari with the help of instruction manuals. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 1009–1023. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/034d7bfeace2a9a258648b16fc626298-Paper-Conference.pdf.

[275] Zhaofen Wu, Hanrong Zhang, Fulin Lin, Wujiang Xu, Xinran Xu, Yankai Chen, Henry Peng Zou, Shaowen Chen, Weizhi Zhang, Xue Liu, Philip S. Yu, and Hongwei Wang. Gam: Hierarchical graph-based agentic memory for llm agents, 2026. URL https://arxiv.org/abs/2604.12285.

[276] Zijian Wu, Xiangyan Liu, Xinyuan Zhang, Lingjun Chen, Fanqing Meng, Lingxiao Du, Yiran Zhao, Fanshi Zhang, Yaoqi Ye, Jiawei Wang, Zirui Wang, Jinjie Ni, Yufan Yang, Arvin Xu, and Michael Qizhe Shieh. MCPMark: A benchmark for stress-testing realistic and comprehensive MCP use. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=uobROwBsJm.

[277] Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. Agentless: Demystifying LLM-based software engineering agents. *arXiv preprint arXiv:2407.01489*, 2024.

[278] Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents. In *International Conference on Machine Learning*, pages 54590–54613. PMLR, 2024.

[279] Weikai Xie, Li Zhang, Shihe Wang, Rongjie Yi, and Mengwei Xu. DroidCall: A dataset for LLM-powered android intent invocation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9116–9134, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.484. URL https://aclanthology.org/2025.findings-emnlp.484/.

[280] Wei Xiong, Chengshuai Shi, Jiaming Shen, Aviv Rosenberg, Zhen Qin, Daniele Calandriello, Misha Khalman, Rishabh Joshi, Bilal Piot, Mohammad Saleh, et al. Building math agents with multi-turn iterative preference learning. *arXiv preprint arXiv:2409.02392*, 2024.

[281] Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. Watch every step! LLM agent learning via iterative step-level process refinement. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1556–1572, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.93. URL https://aclanthology.org/2024.emnlp-main.93/.

[282] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=CfXh93NDgH.

[283] Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Zhiruo Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Melroy Maben, Raj Mehta, Wayne Chi, Lawrence Keunho Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. TheAgentCompany: Benchmarking LLM agents on consequential real world tasks. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=LZnKNApvhG.

[284] Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See Kiong Ng, and Jiashi Feng. MAgIC: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. URL https://aclanthology.org/2024.emnlp-main.416/.

[285] Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=FiM0M8gcct.

[286] Xiangzhe Xu, Guangyu Shen, Zian Su, Siyuan Cheng, Hanxi Guo, Lu Yan, Xuan Chen, Jiasheng Jiang, Xiaolong Jin, Chengpeng Wang, et al. Astra: Autonomous spatial-temporal red-teaming for ai software assistants. *arXiv preprint arXiv:2508.03936*, 2025.

[287] Yiheng Xu, Dunjie Lu, Zhennan Shen, Junli Wang, Zekun Wang, Yuchen Mao, Caiming Xiong, and Tao Yu. Agenttrek: Agent trajectory synthesis via guiding replay with web tutorials. In *International Conference on Learning Representations*, volume 2025, pages 79822–79843, 2025.

[288] Zhangchen Xu, Adriana Meza Soria, Shawn Tan, Anurag Roy, Ashish Sunil Agrawal, Radha Poovendran, and Rameswar Panda. Toucan: Synthesizing 1.5 m tool-agentic data from real-world mcp environments. *arXiv preprint arXiv:2510.01179*, 2025.

[289] Zhihao Xu, Rumei Li, Jiahuan Li, Rongxiang Weng, Jingang Wang, Xunliang Cai, and Xiting Wang. Unlocking implicit experience: Synthesizing tool-use trajectories from text. *arXiv preprint arXiv:2601.10355*, 2026.

[290] Xiangyuan Xue, Yifan Zhou, Guibin Zhang, Zaibin Zhang, Yijiang Li, Chen Zhang, Zhenfei Yin, Philip Torr, Wanli Ouyang, and Lei Bai. Comas: Co-evolving multi-agent systems via interaction rewards. *arXiv preprint arXiv:2510.08529*, 2025.

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

[292] Bohao Yang, Dong Liu, Chenghao Xiao, Kun Zhao, Chen Tang, Chao Li, Lin Yuan, Guang Yang, and Chenghua Lin. Crafting customisable characters with llms: A persona-driven role-playing agent framework. *arXiv preprint arXiv:2406.17962*, 2024.

[293] Chen Yang, Ran Le, Yun Xing, Zhenwei An, Zongchao Chen, Wayne Xin Zhao, Yang Song, and Tao Zhang. Toolmind technical report: A large-scale, reasoning-enhanced tool-use dataset. *arXiv preprint arXiv:2511.15718*, 2025.

[294] John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[295] John Yang, Kilian Lieret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. SWE-smith: Scaling data for software engineering agents. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025.

[296] John Yang, Kilian Lieret, Carlos Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. Swe-smith: Scaling data for software engineering agents. *Advances in Neural Information Processing Systems*, 38, 2026.

[297] Ruihan Yang, Fanghua Ye, Jian Li, Siyu Yuan, Yikai Zhang, Zhaopeng Tu, Xiaolong Li, and Deqing Yang. The lighthouse of language: Enhancing llm agents via critique-guided improvement. *arXiv preprint arXiv:2503.16024*, 2025.

[298] Yuhao Yang, Tianyu Fan, and Chao Huang. Cli-anything: Towards agent-native computer use, 2026. URL https://arxiv.org/abs/2606.03854.

[299] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

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

[301] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025.

[302] Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. Survey on evaluation of LLM-based agents. *arXiv preprint arXiv:2503.16416*, 2025.

[303] Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. Lumos: Learning agents with unified data, modular design, and open-source llms. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2023. URL https://arxiv.org/abs/2311.05657.

[304] Fan Yin, Zifeng Wang, I-Hung Hsu, Jun Yan, Ke Jiang, Yanfei Chen, Jindong Gu, Long Le, Kai-Wei Chang, Chen-Yu Lee, Hamid Palangi, and Tomas Pfister. Magnet: Multi-turn tool-use data synthesis and distillation via graph translation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32600–32616, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1566. URL https://aclanthology.org/2025.acl-long.1566/.

[305] Xuyan Yin, Xinran Yang, Zihao Li, Lixin Zou, and Chenliang Li. Charactercraft: Bridging the literature-reality dialogue gap for practical role-playing agents. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6074–6106, 2025.

[306] Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. AssistantBench: Can web agents solve realistic and time-consuming tasks? *arXiv preprint arXiv:2407.15711*, 2024.

[307] Boxi Yu, Yuxuan Zhu, Pinjia He, and Daniel Kang. UTBoost: Rigorous evaluation of coding agents on SWE-Bench. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3762–3774, 2025. URL https://aclanthology.org/2025.acl-long.189/.

[308] Peijie Yu, Yifan Yang, Jinjian Li, Zelong Zhang, Haorui Wang, Xiao Feng, and Feng Zhang. Multi-mission tool bench: Assessing the robustness of LLM based agents through related and dynamic missions, 2025. URL https://arxiv.org/abs/2504.02623.

[309] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

[310] Yurun Yuan and Tengyang Xie. Reinforce LLM reasoning through multi-agent reflection. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=6k3oFS3Lbl.

[311] Yanwei Yue, Boci Peng, Xuanbo Fan, Jiaxin Guo, Qiankun Li, and Yan Zhang. Mem-t: Densifying rewards for long-horizon memory agents. *arXiv preprint arXiv:2601.23014*, 2026.

[312] Abhay Zala, Jaemin Cho, Han Lin, Jaehong Yoon, and Mohit Bansal. Envgen: Generating and adapting environments via llms for training embodied agents. *arXiv preprint arXiv:2403.12014*, 2024.

[313] Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3053–3077, 2024.

[314] Weihao Zeng, Can Xu, Yingxiu Zhao, Jian-Guang Lou, and Weizhu Chen. Automatic instruction evolving for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

[315] Xingshan Zeng, Weiwen Liu, Xu Huang, Zezhong Wang, Lingzhi Wang, Liangyou Li, Yasheng Wang, Lifeng Shang, Xin Jiang, Ruiming Tang, and Qun Liu. ToolACE-R: Tool learning with adaptive self-refinement. *arXiv preprint arXiv:2504.01400*, 2025.

[316] Xingshan Zeng, Weiwen Liu, Lingzhi Wang, Liangyou Li, Fei Mi, Yasheng Wang, Lifeng Shang, Xin Jiang, and Qun Liu. ToolACE-MT: Non-autoregressive generation for agentic multi-turn interaction. *arXiv preprint arXiv:2508.12685*, 2025.

[317] Chao Zhang, Yuhao Wang, Derong Xu, Haoxin Zhang, Yuanjie Lyu, Yuhao Chen, Shuochen Liu, Tong Xu, Xiangyu Zhao, Yan Gao, Yao Hu, and Enhong Chen. TeaRAG: A token-efficient agentic retrieval-augmented generation framework, 2025.

[318] Danyang Zhang, Situo Zhang, Ziyue Yang, Zichen Zhu, Zihan Zhao, Ruisheng Cao, Lu Chen, and Kai Yu. Progrm: Build better gui agents with progress rewards. *arXiv preprint arXiv:2505.18121*, 2025.

[319] Ding-Chu Zhang, Yida Zhao, Jialong Wu, Liwen Zhang, Baixuan Li, Wenbiao Yin, Yong Jiang, Yu-Feng Li, Kewei Tu, Pengjun Xie, and Fei Huang. EvolveSearch: An iterative self-evolving search agent. In *EMNLP*, 2025. URL https://aclanthology.org/2025.emnlp-main.663/.

[320] Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. Agent security bench (ASB): Formalizing and benchmarking attacks and defenses in LLM-based agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=V4y0CpX4hK.

[321] Hanrong Zhang, Shicheng Fan, Henry Peng Zou, Yankai Chen, Zhenting Wang, Jiayu Zhou, Chengze Li, Wei-Chieh Huang, Yifei Yao, Kening Zheng, Xue Liu, Xiaoxiao Li, and Philip S. Yu. Coevoskills: Self-evolving agent skills via co-evolutionary verification, 2026. URL https://arxiv.org/abs/2604.01687.

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

[323] Jianguo Zhang, Tian Lan, Ming Zhu, Zuxin Liu, Thai Quoc Hoang, Shirley Kokane, Weiran Yao, Juntao Tan, Akshara Prabhakar, Haolin Chen, et al. xlam: A family of large action models to empower ai agent systems. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11583–11597, 2025.

[324] Jianxiang Zhang et al. DeepDive: Advancing deep search agents with knowledge graphs and multi-turn RL. *arXiv preprint arXiv:2509.10446*, 2025.

[325] Jiayi Zhang, Yiran Peng, Fanqi Kong, Cheng Yang, Yifan Wu, Zhaoyang Yu, Jinyu Xiang, Jianhao Ruan, Jinlin Wang, Maojia Song, HongZhang Liu, Xiangru Tang, Bang Liu, Chenglin Wu, and Yuyu Luo. AutoEnv: Automated Environments for Measuring Cross-Environment Agent Learning, 2025. URL https://arxiv.org/abs/2511.19304. Version Number: 2.

[326] Jipeng Zhang, Haolin Yang, Kehao Miao, Ruiyuan Zhang, Renjie Pi, Jiahui Gao, and Xiaofang Zhou. ExeSQL: Self-taught text-to-SQL models with execution-driven bootstrapping for SQL dialects. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24305–24326, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1320. URL https://aclanthology.org/2025.findings-emnlp.1320/.

[327] Kehao Zhang, Shangtong Gui, Sheng Yang, Wei Chen, and Yang Feng. Learning to remember: End-to-end training of memory agents for long-context reasoning. *arXiv preprint arXiv:2602.18493*, 2026.

[328] Shaokun Zhang, Yi Dong, Jieyu Zhang, Jan Kautz, Bryan Catanzaro, Andrew Tao, Qingyun Wu, Zhiding Yu, and Guilin Liu. Nemotron-research-tool-n1: Tool-using language models with reinforced reasoning. *arXiv preprint arXiv:2505.00024*, 2025.

[329] Shaolei Zhang, Ju Fan, Meihao Fan, Guoliang Li, and Xiaoyong Du. Deepanalyze: Agentic large language models for autonomous data science, 2025. URL https://arxiv.org/abs/2510.16872.

[330] Shaowei Zhang and Deyi Xiong. Debate4MATH: Multi-agent debate for fine-grained reasoning in math. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16810–16824, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.862. URL https://aclanthology.org/2025.findings-acl.862/.

[331] Xikai Zhang, Bo Wang, Likang Xiao, Yongzhi Li, Quan Chen, Wenjun Wu, and Liu Liu. Imagine: Integrating multi-agent system into one model for complex reasoning and planning. *arXiv preprint arXiv:2510.14406*, 2025.

[332] Zeyu Zhang, Quanyu Dai, Luyu Chen, Zeren Jiang, Rui Li, Jieming Zhu, Xu Chen, Yi Xie, Zhenhua Dong, and Ji-Rong Wen. Memsim: A bayesian simulator for evaluating memory of llm-based personal assistants. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2024.

[333] Weikang Zhao, Xili Wang, Chengdi Ma, Lingbin Kong, Zhaohua Yang, Mingxiang Tuo, Xiaowei Shi, Yitao Zhai, and Xunliang Cai. MUA-RL: Multi-turn user-interacting agent reinforcement learning for agentic tool use. *arXiv preprint arXiv:2508.18669*, 2025.

[334] Wenting Zhao, Nan Jiang, Eric Hirsch, John Yang, Arman Cohan, Kilian Q. Weinberger Yang, Ofir Press, Yoav Goldberg, Claire Cardie, and Alexander M. Rush. Commit0: Library generation from scratch. *arXiv preprint arXiv:2412.01769*, 2024.

[335] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

[336] Hang Zhou, Yehui Tang, Haochen Qin, Yujie Yang, Renren Jin, Deyi Xiong, Kai Han, and Yunhe Wang. Star-agents: Automatic data optimization with LLM agents for instruction tuning. *Advances in Neural Information Processing Systems*, 37:4575–4597, 2024.

[337] Jinfeng Zhou, Yongkang Huang, Bosi Wen, Guanqun Bi, Yuxuan Chen, Pei Ke, Zhuang Chen, Xiyao Xiao, Libiao Peng, Kuntian Tang, Rongsheng Zhang, Le Zhang, Tangjie Lv, Zhipeng Hu, Hongning Wang, and Minlie Huang. CharacterBench: Benchmarking character customization of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025. URL https://ojs.aaai.org/index.php/AAAI/article/view/34806.

[338] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024.

[339] Xiaofeng Zhou, Heyan Huang, and Lizi Liao. Debate, reflect, and distill: Multi-agent feedback with tree-structured preference optimization for efficient language model enhancement. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9122–9137, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.475. URL https://aclanthology.org/2025.findings-acl.475/.

[340] Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=mM7VurbA4r.

[341] Yuxuan Zhou et al. Open data synthesis for deep research. *arXiv preprint arXiv:2509.00375*, 2025.

[342] Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, and Jiaxuan You. MultiAgentBench: Evaluating the collaboration and competition of LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8580–8622, 2025. doi: 10.18653/v1/2025.acl-long.421. URL https://aclanthology.org/2025.acl-long.421/.

[343] Minghang Zhu, Zhengliang Shi, Zhiwei Xu, Shiguang Wu, Lingjie Wang, Pengjie Ren, Zhaochun Ren, and Zhumin Chen. Bridging the capability gap: Joint alignment tuning for harmonizing LLM-based multi-agent systems. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 21846–21861, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1192. URL https://aclanthology.org/2025.findings-emnlp.1192/.

[344] Yuchen Zhuang, Jingfeng Yang, Haoming Jiang, Xin Liu, Kewei Cheng, Sanket Lokegaonkar, Yifan Gao, Qing Ping, Tianyi Liu, Binxuan Huang, Zheng Li, Zhengyang Wang, Pei Chen, Ruijie Wang, Rongzhi Zhang, Nasser Zalmout, Priyanka Nigam, Bing Yin, and Chao Zhang. Hephaestus: Improving fundamental agent capabilities of large language models through continual pre-training. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6041–6068, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.308. URL https://aclanthology.org/2025.naacl-long.308/.

[345] Yuchen Zhuang, Jingfeng Yang, Haoming Jiang, Xin Liu, Kewei Cheng, Sanket Lokegaonkar, Yifan Gao, Qing Ping, Tianyi Liu, Binxuan Huang, et al. Hephaestus: Improving fundamental agent capabilities of large language models through continual pre-training. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6041–6068, 2025.

[346] Mingchen Zhuge, Changsheng Zhao, Dylan R. Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-judge: Evaluate agents with agents. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pages 80569–80611, 2025. URL https://proceedings.mlr.press/v267/zhuge25a.html.

[347] Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations (ICLR)*, 2025.

[348] Henry Peng Zou, Wei-Chieh Huang, Yaozu Wu, Jizhou Guo, Yankai Chen, Chunyu Miao, Hoang Nguyen, Yue Zhou, Weizhi Zhang, Liancheng Fang, Hanrong Zhang, Fangxin Wang, Pengfei Zhang, Huacan Wang, Langzhou He, Yangning Li, Dongyuan Li, Renhe Jiang, Xue Liu, and Philip S. Yu. LLM-based human-agent collaboration and interaction systems: A survey, 2026. URL https://arxiv.org/abs/2505.00753.

[349] Henry Peng Zou, Chunyu Miao, Wei-Chieh Huang, Yankai Chen, Yue Zhou, Hanrong Zhang, Yaozu Wu, Liancheng Fang, Zhengyao Gu, Zhen Zhang, Kening Zheng, Fangxin Wang, Yi Nian, Shanghao Li, Wenzhe Fan, Langzhou He, Weizhi Zhang, Xue Liu, and Philip S. Yu. When users change their mind: Evaluating interruptible agents in long-horizon web navigation, 2026. URL https://arxiv.org/abs/2604.00892.