

Endogenous Observations in Agent Self-Evolution: A Survey

Zhixin Zhang^{1,2,3,*}, Xinke Jiang^{1,2,3,*}, Jiaran Gao^{1,2,3,*}, Zhibang Yang^{1,2,3},
Yuheng Huang⁴, Junfeng Zhao^{2,3,†}, Xu Chu^{2,3,5,†}, Yasha Wang^{1,6,†}

¹National Engineering Research Center of Software Engineering,
Peking University, Beijing, China

²School of Computer Science, Peking University, Beijing, China

³Key Laboratory of High Confidence Software Technologies,
Ministry of Education, Beijing, China

⁴GRG Banking Equipment Co., Ltd., Guangzhou, China

⁵Center on Frontiers of Computing Studies, Peking University, Beijing, China

⁶Peking University Information Technology Institute (Tianjin Binhai), Tianjin, China

*Equal contribution †Corresponding authors

{zhixinzhang25,xinkejiang}@stu.pku.edu.cn; {chu_xu,zhaojf,wangyasha}@pku.edu.cn

Abstract

As training regimes centered on human-curated data encounter rising costs and an increasingly binding scaling ceiling, a core question in agent learning shifts from enlarging exogenous datasets to enabling agents to **self-evolve** under limited exogenous supervision. Prior surveys on self-evolution and self-improvement typically organize the literature by evolutionary stages or system architectures. While effective for summarizing implementation pipelines, these perspectives do not offer a unified explanation of what an agent actually observes during self-improvement, nor why such signals can serve as sufficient and reliable supervision. To address this gap, we introduce **endogenous observations**, defined as learning signals that arise from an agent’s own generation, evaluation, and interaction processes rather than exogenous labels. We systematize existing approaches into three categories—model-internal, peer-induced, and scenario-grounded observations—according to how these signals are produced, and we summarize the characteristic properties and typical application settings of each category. Finally, we discuss key challenges in reliability, scalability, and long-horizon stability, and outline open directions toward robust and sustainable self-evolving agents.

1 Introduction

LLM-based agents have advanced rapidly through scaling along three axes: model size, compute, and training data (Yao et al., 2023; Robol and Giorgini,

2026; Chen et al., 2025c). Yet the data axis is increasingly becoming the fragile one. Scaling-law and compute-optimal training studies establish that both data quantity and quality critically shape generalization (Kaplan et al., 2020; Hoffmann et al., 2022), while projections of available public human-generated text suggest that sustained reliance on high-quality exogenous data is approaching a hard ceiling (Villalobos et al., 2024). This constraint re-frames the central question of agent self-evolution: the challenge is no longer merely to collect more labels, but to design agents capable of converting their own actions, interactions, and failures into reusable experience.

Existing surveys offer complementary perspectives on this transition. Data-centric surveys address synthetic data generation, annotation, and curation (Tan et al., 2024; Long et al., 2024; Alismail and Lanquillon, 2025), while self-evolution surveys organize methods by training stage, system component, or agent architecture (Tao et al., 2024; He et al., 2025; Gao et al., 2026). Memory and multi-agent surveys further illuminate important subsystems (Zhang et al., 2025d; Wu et al., 2025; Chen et al., 2025c). These perspectives are valuable, yet they typically enter the picture only after a crucial event has already occurred: a reward has been assigned, a critique has been written, a trajectory has been stored, or a verifier has returned a score.

We argue that self-evolution requires a more foundational taxonomy anchored by a single question: *what does the agent observe when it im-*

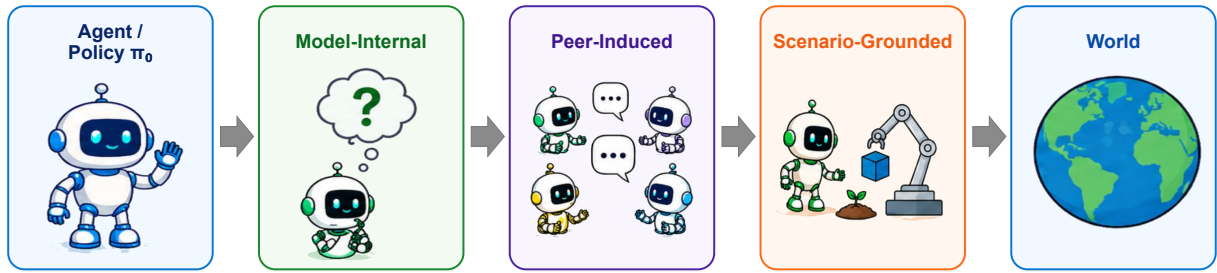


Figure 1: Overview of the three endogenous observation channels, moving from self to social to world: model-internal, peer-induced, and scenario-grounded observations.

proves itself? This is not merely an implementation detail. A confidence score, a self-critique, a peer’s counterexample, a compiler error, and a retrieved document differ fundamentally in provenance, cost, reliability, and susceptibility to gaming, and they support distinct update mechanisms. The same reinforcement-learning or preference-learning objective can be either well grounded or dangerously circular, depending on whether its signal is semantically grounded, socially challenged, or merely self-confirming.

What the abstraction adds. Self-feedback, multi-agent learning, and environment feedback name established mechanisms; our contribution is an orthogonal unit of analysis. We begin one step earlier than the optimizer and identify the observation event that supplies its evidence, then compare that event by provenance, grounding, acquisition cost, conversion interface, and failure risk. This reveals distinctions hidden by architecture-centric views: the same GRPO or preference objective may be driven by an internal confidence proxy, challenged by an independent peer, or grounded by execution. It also makes hybrid systems auditable without forcing each complete system into one mutually exclusive class.

Observation-Centric View. We define *endogenous observations* as learning-relevant signals elicited by an agent’s own actions and reused within its self-evolution loop. The key claim is not that all information must originate inside the model, but that self-evolution changes which signals become visible to the learning loop. Some observations arise from the model’s own distribution, generated language, internal representations, and self-constructed objectives. Others come from peer models that critique, challenge, rank, or adjudicate the agent’s outputs. Still others originate from external substrates such as tools, tests, retrieval systems, simulators, users, and memories. The

trace remains endogenous when it is elicited by the agent’s action and reused by the loop, even if the substrate that makes the trace visible is external. A detailed definition and boundary discussion are provided in Appendix A.1.

Based on observation provenance and formation mechanism, we organize existing approaches into three channels, as illustrated in Figure 1; the full leaf-level taxonomy appears in Figure 5. Figures 2–4 further mark the channel-specific boundary for each class: each diagram corresponds to a distinct source of observation and a distinct class of validity risks.

- **Model-Internal Observations:** Observations that arise from a model’s own computation and generation, including consistency cues, uncertainty, self-reflection, self-generated tasks, pseudo-labels, and latent evaluative signals (Liang et al., 2024).
- **Peer-Induced Observations:** Observations that emerge through interaction with other models or agent populations, such as critiques, counterexamples, rankings, role-specific judgments, and group decisions (Gao et al., 2026).
- **Scenario-Grounded Observations:** Observations elicited through interaction with tools, validators, retrieval systems, simulators, workflows, or memory systems, including execution outcomes, constraint satisfaction, retrieved evidence, and interaction trajectories (Long et al., 2024).

Contributions. Our core contribution is an observation-centric account of agent self-evolution. First, we define endogenous observations by action contingency, loop reuse, and non-static supervision, separating endogenous elicitation from external grounding. Second, we organize the literature into model-internal, peer-induced, and scenario-grounded channels, with the full taxonomy tree

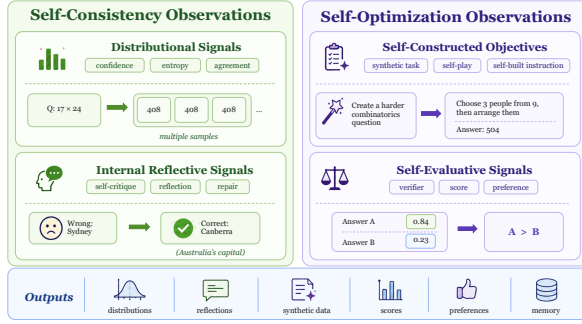


Figure 2: Model-internal observation.

provided in Figure 5. Third, we connect representative methods in each channel to the conversion rules that turn observations into updates, including rejection sampling, SFT, preference learning, reward modeling, RLVR, trajectory distillation, and memory update. Finally, we synthesize design and reporting insights about observation validity, novelty, stability, Goodhart pressure, evaluator independence, lifecycle governance, and stopping rules. **Survey scope and coding.** We searched the ACL Anthology, arXiv, OpenReview, and Crossref-indexed venues through August 28, 2026, combining queries for agents or language models with self-improvement, self-feedback, reflection, self-play, tool learning, execution feedback, trajectories, and memory, followed by citation tracing. After deduplication and full-text screening, we retained 96 primary methods and 35 contextual sources. We code the reusable observation rather than the whole system: self-produced signals are model-internal, signals formed by another model are peer-induced, and traces returned by tools or environments are scenario-grounded. Appendix B gives the queries, inclusion criteria, boundary cases, and integrity checks.

2 Model-Internal Observations

Model-internal observations are *endogenously triggered* signals that arise from a model’s own inference processes, without relying on exogenous agents, stronger teacher models, or environment execution. As shown in Figure 2, this channel turns internal distributions, reflections, self-constructed objectives, and self-evaluative signals into reusable learning pressure. Although such observations do not provide hard semantic verification, they are inexpensive to obtain, highly scalable, and repeatedly extractable from training or decoding dynamics. Recent verifier-free reinforcement learning

and unsupervised reasoning methods show that distribution-derived rewards (Zhao et al., 2026b; Yu et al., 2025a; Zuo et al., 2025) and entropy-based objectives (Zhang et al., 2025a; Agarwal et al., 2025b) can support learning under limited labeled supervision. Inspired by internal consistency and self-feedback (Liang et al., 2024), we divide this channel into ① *Self-Consistency Observations* and ② *Self-Optimization Observations*.

2.1 Self-Consistency Observations

Self-consistency observations capture agreement, uncertainty, and reliability cues embedded in the model’s own outputs and decoding dynamics. Such cues may appear as response-level agreement across multiple samples, decoding-level stability in logits, or reflective traces that expose uncertainty and local inconsistency. They do not certify semantic correctness, but they can indicate where the model is unstable, under-calibrated, or internally inconsistent. This makes them useful as intrinsic rewards, selection criteria, or test-time adaptation signals (Zhao et al., 2026b; Yu et al., 2025a; Zuo et al., 2025). We further distinguish *Distributional Signals*, which derive scalar pressure from probability distributions, and *Internal Reflective Signals*, which use language-form critiques or correction traces.

2.1.1 Distributional Signals

Distributional signals are self-generated rewards derived from output probabilities, agreement, or uncertainty. INTUITOR (Zhao et al., 2026b), RLPR (Yu et al., 2025a), and RLSC (Li et al., 2025b) exploit confidence-related statistics such as KL divergence or self-assessed certainty for reinforcement learning with minimal or unlabeled supervision. Entropy-based variants form a second branch of distributional observation. EMPO (Zhang et al., 2025a) minimizes predictive entropy to encourage decisive reasoning on unlabeled queries, while Agarwal et al. (Agarwal et al., 2025b) study entropy minimization across finetuning, RL, and inference-time adjustment. Test-time scaling provides another source of distributional signal: TTRL (Zuo et al., 2025) converts majority voting and Maj@N statistics into reward estimates, and PEAR (Huang et al., 2025) uses phase-aware entropy shaping to reduce redundant reasoning while maintaining performance. Internal-state routing and exploration methods further learn temperature policies, estimate stepwise failure risk, or

shape advantages from uncertainty (Zhou et al., 2026b; Ye et al., 2026c; Xie et al., 2026a). Related work studies internal learning-state variables through metacognitive learning (Liu and van der Schaar, 2025), SEAL (Zweiger et al., 2025), SE-LAUR (Zhang et al., 2026a), and entropy-guided privileged self-distillation (Wu et al., 2026). These methods are scalable, but their signals remain proxies that require calibration against semantic correctness.

2.1.2 Internal Reflective Signals

Internal reflective signals externalize uncertainty and inconsistency as critiques, suggestions, or repair traces. Reflexion (Shinn et al., 2023) stores natural-language reflections as reusable feedback, and Self-Refine (Madaan et al., 2023) iterates draft generation, feedback, and rewriting without additional supervision. Compared with scalar confidence, such traces expose more structured information: the model may identify a missing assumption, flag a weak reasoning step, or propose a repair that can later be distilled into training data.

More recent approaches make reflective signals more granular or trainable. SSR (Shi et al., 2025) localizes unreliable reasoning steps through decomposed sub-questions and targeted refinement, producing interpretable correction chains. SCoRe (Kumar et al., 2025) trains self-correction through multi-turn online RL on self-generated correction traces, reducing mismatch between correction behavior observed at training and test time. LoongReflect instead treats reflection and backtracking as memory-control actions and distills globally informed reflection into a long-horizon search policy (Zhang et al., 2026h). Reflective signals provide richer diagnostics than scalar confidence, but can saturate when critiques become fluent rationalizations rather than verified corrections.

2.2 Self-Optimization Observations

Self-optimization observations use endogenous signals to construct objectives, curricula, or evaluators for sustained updates. Compared with self-consistency signals, they place stronger structure on the learning loop: the model does not only select among samples, but also creates tasks, pseudo-labels, reward surrogates, or verifier data that shape later parameter updates. We divide them into *Self-Constructed Objectives*, which expand training coverage, and *Self-Evaluative Signals*, which build evaluation mechanisms for endogenous learning.

2.2.1 Self-Constructed Objectives

Self-constructed objectives treat model generations as learning targets. Self-Instruct (Wang et al., 2023), Unnatural Instructions (Honovich et al., 2023), and instruction backtranslation (Li et al., 2024b) bootstrap instruction–response data with limited human annotation. Their key observation is that a model can generate candidate tasks, answers, or demonstrations whose filtered subset becomes new supervision, thereby converting generation into a data-expansion mechanism.

Because naive self-generation can create low-value loops, later methods introduce filtering, role alternation, or regularization. Selective Reflection Tuning (Li et al., 2024a) recycles only selected reflective traces, self-play fine-tuning (Chen et al., 2024) alternates teacher–student roles to create an endogenous curriculum, and Rstar-Math (Guan et al., 2025) shows how self-evolved reasoning data can support mathematical problem solving. SQL-Forge combines syntax constraints, reverse translation, and iterative domain exploration to filter and diversify self-generated Text-to-SQL data (Guo et al., 2025). Recent optimization-oriented methods such as CREAM (Wang et al., 2025b), SS-APO (Chu et al., 2025), and RLP (Hatamizadeh et al., 2026) further introduce consistency regularization, game-theoretic objectives, or information-driven reinforcement signals. These methods scale data creation, but rely on filtering to avoid drift.

2.2.2 Self-Evaluative Signals

Self-evaluative signals construct verifiers or reward surrogates from model-generated data. V-Star (Hosseini et al., 2024) trains verifiers on self-generated correct and incorrect solutions, and Generalist Reward Models (Li et al., 2025c) suggest that reward-relevant structure may already be latent in pretrained LLMs. This line moves beyond using the model as a generator: it asks whether the model can also expose evaluator-like structure that makes endogenous learning more selective.

Recent alignment-oriented methods strengthen this evaluative interface. Critic-RM (Yu et al., 2025b) augments scalar rewards with critiques, DICE (Chen et al., 2025a) bootstraps preference data from implicit DPO rewards, and Process-RM (Zhang et al., 2025b) promotes process-level scores into reusable supervision. Rubric-guided process rewards supervise stepwise routing decisions (Ye et al., 2026b), while persistent-consistency self-distillation weights token-level su-

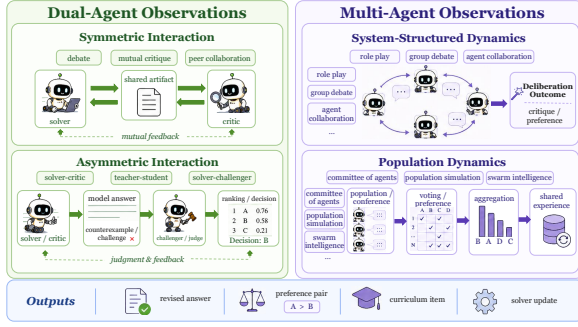


Figure 3: Peer-induced observation.

pervision by the stability of teacher-favoring signals (Lv et al., 2026). These signals make endogenous learning more structured, but policy–evaluator co-evolution must be controlled to prevent biased self-confirmation.

2.3 Observation-to-Update Interfaces

Model-internal observations matter only after they are converted into learning pressure. Agreement can serve as a rejection-sampling filter, an intrinsic reward, a preference label between traces, or a trigger for test-time adaptation. A generated critique can be stored as memory, distilled into a repaired answer, or converted into a process-level reward. A model-written task can become SFT data, a self-play curriculum, or a challenge set for later verification. Because this channel is weakly grounded, these conversions need safeguards such as held-out calibration, novelty filtering, or separation between generation and evaluation. Otherwise, self-feedback may improve surface consistency while reducing epistemic coverage, a risk consistent with broader concerns about recursive training on generated data (Shumailov et al., 2024a,b).

3 Peer-Induced Observations

Peer-induced observations arise when a model treats another model or multi-model system as an interaction partner and derives signals through feedback, questioning, adjudication, or population dynamics. As shown in Figure 3, this channel converts critiques, counterexamples, rankings, and group decisions into reusable observations. Such signals are endogenous to inter-model interaction rather than fixed labels, and can expose errors while reducing manual annotation (Chen et al., 2025b; Liu et al., 2025d; Qiu et al., 2025b; Chan et al., 2024). Unlike model-internal self-feedback, peer-induced observation can reveal blind spots that are

not visible from one model’s own samples; unlike scenario-grounded feedback, however, it does not automatically provide hard semantic verification. We organize this channel into ①Dual-Agent Observations and ②Multi-Agent Observations (Motwani et al., 2025; Yuan et al., 2025; Park et al., 2025).

3.1 Dual-Agent Observations

Dual-agent observations arise when a model elicits supervision by interacting with one peer under a structured two-player protocol. The peer’s critique, counterexample, score, or adversarial prompt becomes the signal for refinement, and the resulting update is endogenous because it is conditioned on the focal agent’s own artifact. Such protocols are more controllable than larger collectives and make role-specific credit assignment easier. They often yield sharper error discovery than single-model self-refinement, while avoiding the coordination complexity of many-agent systems. Their weakness is that two agents may still share model-family biases, converge to premature agreement, or oscillate without an external adjudicator.

3.1.1 Symmetric Interaction

Symmetric interaction gives both agents comparable roles, as in debate, peer-to-peer challenge, cooperative co-evolution, or self-play. DebateCoder (Chen et al., 2025b) illustrates test-case-driven debate for code generation, where agents expose implementation errors by proposing objections and tests. Sequential cooperative co-evolution (Ma et al., 2024), Self-Reflect+Debate (Ki et al., 2025), and SeRL (Fang et al., 2025) instantiate related mutual-revision or self-play patterns. In these settings, the observation is not a single answer but the relation produced by exchange: disagreement, counterargument, revised artifact, or agreed solution.

3.1.2 Asymmetric Interaction

Asymmetric interaction assigns different roles, such as solver–critic, teacher–student, or solver–challenger. This separation yields clearer signal interfaces and often improves credit assignment: the solver produces an artifact, the critic or challenger exposes a weakness, and the learner can attach the resulting signal to a specific revision. Examples include Debater/Judger decomposition for harmful content detection (Liu et al., 2025d), teacher–student construction of challenging QA instances (Qiu et al., 2025b), solver–challenger

refinement for retrieval or reasoning (Hu et al., 2025; Huang et al., 2026), and Absolute Zero’s curriculum construction with little or no human task data (Zhao et al., 2025).

Asymmetry is useful for constructing difficult cases or evaluative feedback without a full environment, but evaluator or challenger bias may dominate the loop if calibration and anti-collapse constraints are weak. Related inference-time systems use a learned refiner or externally supplied thought to alter a solver’s subsequent reasoning (Zhou et al., 2026a; Liu et al., 2025b); these are useful boundary cases because the observation changes behavior without necessarily producing a persistent learner update.

3.2 Multi-Agent Observations

Multi-agent observations emerge from systems of three or more agents through role specialization, structured collaboration, group deliberation, or population-level selection. They can broaden coverage through redundancy and division of labor, but the observation pathway becomes a system property rather than a single interaction edge. Compared with dual-agent exchange, multi-agent settings can generate richer signals such as cross-checking, emergent specialization, and group disagreement, while also amplifying coordination overhead, collusion risk, and reward attribution difficulty.

3.2.1 System-Structured Dynamics

System-structured dynamics organize agents into explicit workflows or hierarchies, such as role-based teams, planner–executor architectures, or generator–verifier–refiner pipelines. Automatic role differentiation (Li et al., 2025a), sequential multi-agent training (Motwani et al., 2025), and centralized hierarchical planning (Zhang et al., 2026e) show how role outputs can become structured observations for downstream agents. In such systems, plans, critiques, verifications, and intermediate artifacts form a computation graph of observations that can localize errors and distribute expertise.

This structure improves reliability through specialization, but it is sensitive to weak links: a single poor role can corrupt the downstream workflow. Multi-agent protocol distillation compresses proprietary agents’ search, repair, and exploration traces into supervision for an open-source policy (Liu et al., 2026b). Recent work extends this line to-

ward robustness and persistent skill discovery (tse Huang et al., 2025; Weng et al., 2026; Peng et al., 2026; Alzubi et al., 2026). These methods suggest that multi-agent observations are most useful when the protocol records not only the final group output, but also which role produced which intermediate signal and how later agents used it.

3.2.2 Population Dynamics

Population dynamics aggregate signals from group debate, evolutionary selection, or multi-agent post-training. ChatEval (Chan et al., 2024) uses debate-style evaluator teams, EvoAgent (Yuan et al., 2025) searches over agent configurations, and MAPORL (Park et al., 2025) rewards both answer quality and collaboration. Shared-Reflector (Bo et al., 2024) further highlights the need for counterfactual credit assignment in shared components.

Instead of a fixed workflow, population methods treat aggregate behavior as the observation: agreement patterns, minority objections, selection outcomes, and group-level rewards become signals for improving agents or the collective. Population methods improve coverage only when diversity is preserved and correlated agreement is controlled. Their governance challenge is therefore not only accuracy, but also maintaining diversity while preventing collusion, mode collapse, and runaway compute.

3.3 Interactional Evidence and Credit Assignment

Peer-induced observation changes the unit of evidence from a single statement to an interactional relation: critique, counterexample, rank, disagreement, or revision. A useful protocol should separate the artifact being judged, the challenge or critique, the decision rule, and the revision that follows. Otherwise, a group reward may reinforce persuasive style or majority agreement rather than error discovery. Diversity is therefore a control variable: homogeneous agents may converge quickly but share the same misconception, while heterogeneous agents may expose more errors at higher coordination cost. Recent safety-oriented evidence on self-evolving agent populations further suggests that group norms can drift, so peer-induced learning should be evaluated for both task improvement and norm preservation (Wang et al., 2026a).

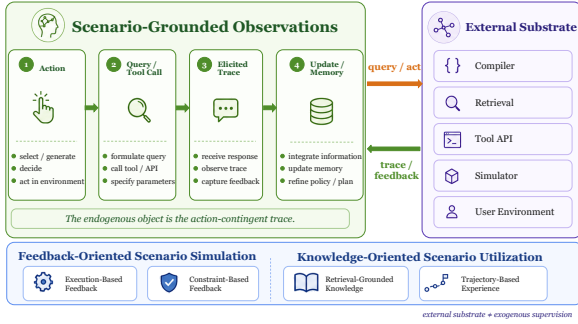


Figure 4: Scenario-grounded observation.

4 Scenario-Grounded Observations

Scenario-grounded observations are *endogenously triggered* signals that arise when a model actively interacts with executable environments, validators, retrieval systems, tools, or structured workflows (Dou et al., 2024; Wang et al., 2025a). As shown in Figure 4, this channel uses execution, constraints, retrieval, and trajectories as action-contingent evidence. Although the substrate is external, the observed trace is endogenous when it is elicited by the model’s own action and reused by the learning loop. This distinction is important because external grounding does not make the signal exogenous: the compiler, corpus, simulator, or memory system only becomes informative after the agent chooses a program, query, rollout, or interaction that produces a reusable trace. We divide this channel into ❶ *Feedback-Oriented Scenario Simulation* and ❷ *Knowledge-Oriented Scenario Utilization*.

4.1 Feedback-Oriented Scenario Simulation

Feedback-oriented scenario simulation converts model actions into adjudicable signals through execution environments or validators. The key advantage is that the feedback is relatively hard: a program compiles or fails, a SQL query returns a result, a schema is satisfied or violated, and a simulator transition can be inspected. This makes the channel attractive for reinforcement learning, filtering, and repair, but the resulting supervision is only as reliable as the coverage and fidelity of the checks (Dou et al., 2024; Lee et al., 2025). We distinguish execution-based feedback, which evaluates semantic consequences through running artifacts, from constraint-based feedback, which checks structural or protocol compliance.

4.1.1 Execution-Based Feedback

Execution-based feedback derives supervision from compilation, unit tests, SQL results, run-time exceptions, or simulator transitions. Step-Coder converts compiler and test outcomes into reinforcement-learning signals through curriculum decomposition and fine-grained optimization (Dou et al., 2024); RoboGen (Wang et al., 2024b) uses simulator outcomes in a self-guided propose–generate–learn loop; AutoIF (Dong et al., 2024) constructs instruction-following data from execution feedback; and HeroSQL (Qiu et al., 2025a) refines SQL generation through outcomes of self-generated queries. Table-reasoning systems likewise reuse evidence-guided intermediate table states, rollback signals, or gold SQL execution trajectories to prune large tables (Ye et al., 2026a; Guo et al., 2026). Dr. Zero (Yue et al., 2026) extends the same pattern to tool and search behavior. The strength of this signal is hard verification; the weakness is sparse rewards, execution cost, and incomplete environment coverage.

4.1.2 Constraint-Based Feedback

Constraint-based feedback uses grammar, schema, format, or protocol checks to provide dense validation without full semantic execution. DOMINO (Beurer-Kellner et al., 2024) enforces constraints during decoding, while Good-Enough (Lee et al., 2025) repairs outputs against real-world schemas. Mechanisms (Agarwal et al., 2025a) studies syntactic evaluation signals, DeepSeek-R1 (Guo et al., 2024) uses format-oriented rewards, and StackPlanner (Zhang et al., 2026e) applies format feedback to argument generation for sub-agents and tool calls. Such constraints improve structural compliance, but need complementary evidence to ensure semantic success.

4.2 Knowledge-Oriented Scenario Utilization

Knowledge-oriented scenario utilization uses retrieved evidence, tool outputs, trajectories, or memory to increase self-information gain. Unlike execution checks, these observations expose information not known *a priori*, making evidence quality, tool-use calibration, and lifecycle governance central concerns (Qian et al., 2025; Kim et al., 2024; Wang et al., 2024a, 2025a). This category is broader than feedback simulation: the agent learns not only from pass/fail outcomes, but also from what it chooses to retrieve, store, revisit, or forget.

4.2.1 Retrieval-Grounded Knowledge

Retrieval-grounded knowledge treats documents, exemplars, API outputs, or tool results as observations for answering, planning, and tool selection. RaDA supports web-agent planning with retrieved exemplars (Kim et al., 2024), SMART calibrates when tools are necessary (Qian et al., 2025), and ReAct (Yao et al., 2023) and TC-RAG (Jiang et al., 2025) integrate retrieval with reasoning. DFAMS uses attributed internal information flow to align retrieval across heterogeneous knowledge sources (Yang et al., 2026a), while ScaffoldAgent evaluates outline edits from retrieval gain, structural coherence, and trial-generation quality (Yang et al., 2026b). ToolBeHonest (Zhang et al., 2024), ET-Agent (Chen et al., 2026), Tool-R0 (Acikgoz et al., 2026), Agent0 (Xia et al., 2025), and reward-free world-knowledge exploration (Zhang et al., 2026d) make query selection, tool choice, and evidence reuse part of the self-evolution loop. The main risk is that stale or irrelevant evidence becomes persistent training pressure.

4.2.2 Trajectory-Based Experience

Trajectory-based experience distills multi-step interaction traces—dialogue turns, tool calls, state transitions, failures, repairs, or successful rollouts—into learnable data or retrievable memory (Wu et al., 2025; Zhang et al., 2025d; Hu et al., 2026). STE (Wang et al., 2024a) and STeCa (Wang et al., 2025a) use trial-and-error, memory, and trajectory calibration for tool learning, while Stack-Planner (Zhang et al., 2026e), EDGE (Xie et al., 2026b), Mem0 (Chhikara et al., 2025), and Live-Evo (Zhang et al., 2026g) emphasize experience retrieval, distillation, and online memory evolution. The same pattern appears in software, recommendation, EDA, and harness-engineering agents (Zhang, 2026; Wang et al., 2026b; Yu and Ren, 2026; Robol and Giorgini, 2026; Lin et al., 2026), and in recursive systems such as Godel Agent (Yin et al., 2025), Darwin Godel Machine (Zhang et al., 2026b), Polaris (Kakade et al., 2026), Hyperagents (Zhang et al., 2026c), and Autogenesis (Zhang et al., 2026f). These observations are rich but require provenance, update, forgetting, and rollback policies.

4.3 Lifecycle and Grounding Governance

Scenario-grounded observations require a clear boundary between external substrate and endogenous trace. A compiler, retrieval corpus, simulator,

or user environment may be external, but the observation remains endogenous when the agent chooses the action that elicits the trace and the loop reuses it. This distinction matters because the policy cannot improve the corpus or test harness directly, but it can improve when to query, what evidence to trust, which trace to remember, and how to avoid repeating failed actions.

This channel also creates durable state. A memory written today can shape tomorrow’s query; a retrieved document can become stale; and a tool API can change while the learned policy still assumes the old interface. Scenario-grounded self-evolution should therefore report tool and corpus versions, timestamps for retrieved evidence, eviction rules, conflict resolution, and rollback mechanisms. Because every test, schema, retrieval result, or simulator state is a proxy for task intent, repeated optimization can induce Goodhart pressure. Recent work on unfaithful self-evolving agents and lifelong-agent forgetting reinforces the need to track verifier coverage, retrieval freshness, memory versioning, and preservation of old capabilities (Zhao et al., 2026a; Yu et al., 2026).

5 Cross-Channel Comparison

Table 1 summarizes the central trade-off. Model-internal observations are cheap and scalable but can confuse confidence with correctness. Peer-induced evidence can expose blind spots, but only when roles, judges, and memories are sufficiently independent. Scenario-grounded traces provide stronger evidence, yet visible tests can under-cover intent and retrieved or stored evidence can become stale. A practical loop can escalate from internal signals to peer challenge and then grounded verification; conversely, grounded failures can be compressed into reflections or peer challenges. Systems should therefore report which channel produced each accepted observation, how it became learning pressure, and whether fixed-budget channel-drop ablations preserve the gain. Appendix C.1 provides method-level cases and a quality framework.

6 Future Work

This perspective yields testable questions rather than only broad themes.

Selection and robustness. Under a fixed observation budget, which acceptance policy maximizes held-out improvement? Channel-removal and novelty-filtering ablations should separate gen-

Channel	Typical observation	Grounding / cost	Common reuse	Principal risk
Model-internal	Confidence, uncertainty, self-critique, generated tasks, latent scores	Weak / low	Filtering, SFT, intrinsic reward, test-time revision	Self-confirmation, redundancy, evaluator drift
Peer-induced	Critiques, counterexamples, ranks, debate, group decisions	Medium / medium-high	Revision, preference learning, group reward	Correlated bias, judge dependence, weak credit assignment
Scenario-grounded	Tests, constraints, retrieval, tool traces, trajectories, memory	Stronger / medium-high	RLVR, repair, trajectory distillation, memory update	Proxy gaming, stale evidence, incomplete coverage

Table 1: Cross-channel comparison. Grounding strength and cost are relative tendencies rather than universal rankings; hybrid systems should code each reusable observation separately.

uine capability gaps from sampling variation and self-confirming confidence (Razin et al., 2026; Zhang et al., 2025c). Counterfactual tests can replace a useful signal with a plausible but incorrect critique, correlated peer vote, stale retrieval result, or incomplete verifier outcome and measure whether the loop rejects it. Multi-cycle evaluations should track error amplification and distributional collapse (Shumailov et al., 2024a,b).

Composition and calibration. When should a system escalate from inexpensive internal signals to peer challenge or scenario-grounded verification? Experiments should vary the amount and timing of human labels, hidden tests, and external audits under a fixed total budget, measuring calibration, drift, and the marginal value of each channel (Luo et al., 2025). Conflicting observations also require explicit resolution and channel-level credit assignment.

Collective and long-horizon limits. Peer studies should vary model family, role, memory sharing, and judge independence to determine when diversity reveals blind spots rather than producing correlated consensus (Gao et al., 2026). Repeated-cycle studies should hold compute fixed and measure saturation, forgetting, memory growth, evidence aging, and rollback effectiveness. These tests can determine whether endogenous information is eventually exhausted or shifts as agents and environments co-evolve (Tao et al., 2024).

7 Conclusion

This survey presents an observation-centric account of agent self-evolution. Endogenous observations are action-contingent signals elicited by an agent’s own generation, evaluation, or interaction and reused by the learning loop. Organizing the literature into model-internal, peer-induced, and scenario-grounded channels clarifies what information an agent can obtain, how that information becomes learning pressure, and why each channel requires different safeguards. The goal is not a

closed self-confirming loop, but a self-evolution process whose observations remain informative, diverse, grounded, and auditable.

Limitations

Our taxonomy classifies observation events by provenance rather than assigning complete systems to mutually exclusive categories; hybrid agents can span all three channels. The literature is rapidly evolving and includes recent preprints, which we use to identify mechanisms and risks rather than durable superiority. Cross-paper quantitative comparison is limited by differences in models, tasks, budgets, and protocols, so we compare numbers only within compatible settings. Finally, source papers often omit acceptance rates, evaluator overlap, verifier coverage, and memory lifecycle details, limiting causal attribution of improvements to a specific observation channel. Collection and boundary decisions are documented in Appendix B.

Ethical Considerations

This survey analyzes publicly available scientific literature and does not involve new data collection, human participants, or the training or deployment of agent systems. Nevertheless, self-evolving agents may amplify bias, privacy leakage, security vulnerabilities, reward hacking, and other harmful behavior when self-generated signals are treated as reliable supervision. Our taxonomy is descriptive rather than an endorsement of unrestricted autonomous self-improvement. We therefore emphasize provenance tracking, independent evaluation, human oversight, access controls, rollback mechanisms, and lifecycle governance, especially for systems that interact with users, tools, persistent memory, or changing environments. Researchers and practitioners remain responsible for evaluating risks in the specific domains, populations, data, and deployment settings in which these methods are used.

Acknowledgments

This work is supported by the Natural Science Foundation of China (Grant No.62506010). Generative AI tools assisted with language polishing, figure preparation, and checking camera-ready formatting. The authors reviewed and verified the resulting text, citations, diagrams, technical claims, taxonomy decisions, and final manuscript, and remain fully responsible for the content.

References

- Emre Can Acikgoz, Cheng Qian, Jonas Hübötter, Heng Ji, Dilek Hakkani-Tür, and Gokhan Tur. 2026. [Tool-r0: Self-evolving llm agents for tool-learning from zero data](#). *Preprint*, arXiv:2602.21320.
- Ananth Agarwal, Jasper Jian, Christopher D Manning, and Shikhar Murty. 2025a. [Mechanisms vs. outcomes: Probing for syntax fails to explain performance on targeted syntactic evaluations](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33737–33757. Association for Computational Linguistics.
- Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. 2025b. [The unreasonable effectiveness of entropy minimization in LLM reasoning](#). In *Advances in Neural Information Processing Systems 38*, NeurIPS 2025. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Ahmad Alismail and Carsten Lanquillon. 2025. [A Survey of LLM-Based Methods for Synthetic Data Generation and the Rise of Agentic Workflows](#), pages 119–135. Springer Nature Switzerland.
- Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Weiyuan Chen, and Tu Vu. 2026. [Evoskill: Automated skill discovery for multi-agent systems](#). *Preprint*, arXiv:2603.02766.
- Shengnan An, Xunliang Cai, Xuezhi Cao, Xiaoyu Li, Yehao Lin, Junlin Liu, Xinxuan Lv, Dan Ma, Xu-anlin Wang, Ziwen Wang, and Shuang Zhou. 2025. [AMO-bench: Large language models still struggle in high school math competitions](#). *arXiv preprint arXiv:2510.26768*.
- Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2024. [Guiding LLMs the right way: Fast, non-invasive constrained generation](#). In *Proceedings of the 41st International Conference on Machine Learning*, pages 3658–3673. PMLR.
- Xiaohe Bo, Zeyu Zhang, Quanyu Dai, Xueyang Feng, Lei Wang, Rui Li, Xu Chen, and Ji-Rong Wen. 2024. [Reflective multi-agent collaboration based on large language models](#). In *Advances in Neural Information Processing Systems 37*, NeurIPS 2024, pages 138595–138631. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. [Chateval: Towards better LLM-based evaluators through multi-agent debate](#). In *ICLR*.
- Yupeng Chang, Yi Chang, and Yuan Wu. 2026. [BA-LoRA: Bias-alleviating low-rank adaptation to mitigate catastrophic inheritance in large language models](#). In *The Fourteenth International Conference on Learning Representations*.
- Yupeng Chang, Chenlu Guo, Yi Chang, and Yuan Wu. 2025. [LoRA-MGPO: Mitigating double descent in low-rank adaptation via momentum-guided perturbation optimization](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 648–659.
- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. [A survey on evaluation of large language models](#). *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.
- Changyu Chen, Zichen Liu, Chao Du, Tianyu Pang, Qian Liu, Arunesh Sinha, Pradeep Varakantham, and Min Lin. 2025a. [Bootstrapping language models with DPO implicit rewards](#). In *ICLR*.
- Jizheng Chen, Kounianhua Du, Xinyi Dai, Weiming Zhang, Xihuai Wang, Yasheng Wang, Ruiming Tang, Weinan Zhang, and Yong Yu. 2025b. [Debatecoder: Towards collective intelligence of llms via test case driven llm debate for code generation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12055–12065. Association for Computational Linguistics.
- Shuaihang Chen, Yuanxing Liu, Wei Han, Weinan Zhang, and Ting Liu. 2025c. [A survey on llm-based multi-agent system: Recent advances and new frontiers in application](#). *Preprint*, arXiv:2412.17481.
- Yifei Chen, Guanting Dong, and Zhicheng Dou. 2026. [Et-agent: Incentivizing effective tool-integrated reasoning agent via behavior calibration](#). *Preprint*, arXiv:2601.06860.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. [Self-play fine-tuning converts weak language models to strong language models](#). In *Proceedings of the 41st International Conference on Machine Learning*, pages 6621–6642. PMLR.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. [Mem0: Building production-ready ai agents with scalable long-term memory](#). In *ECAI 2025*. IOS Press.
- Xu Chu, Zhixin Zhang, Tianyu Jia, and Yujie Jin. 2025. [Stackelberg self-annotation: A robust approach to data-efficient LLM alignment](#). In *Advances in Neural*

- Information Processing Systems* 38, NeurIPS 2025. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Guanting Dong, Keming Lu, Chengpeng Li, Tingyu Xia, Bowen Yu, Chang Zhou, and Jingren Zhou. 2024. [Self-play with execution feedback: Improving instruction-following capabilities of large language models](#). *Preprint*, arXiv:2406.13542.
- Shihan Dou, Yan Liu, Haoxiang Jia, Enyu Zhou, Limao Xiong, Junjie Shan, Caishuang Huang, Xiao Wang, Xiaoran Fan, Zhiheng Xi, Yuhao Zhou, Tao Ji, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. [Stepcoder: Improving code generation with reinforcement learning from compiler feedback](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4571–4585. Association for Computational Linguistics.
- Wenkai Fang, Shunyu Liu, Yang Zhou, Kongcheng Zhang, Tongya Zheng, Kaixuan Chen, Mingli Song, and Dacheng Tao. 2025. [Serl: Self-play reinforcement learning for large language models with limited data](#). In *Advances in Neural Information Processing Systems* 38, NeurIPS 2025. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, Hongru Wang, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xi-ang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, and 8 others. 2026. [A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence](#). *Preprint*, arXiv:2507.21046.
- Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. [rstar-math: Small LLMs can master math reasoning with self-evolved deep thinking](#). In *Proceedings of the 42nd International Conference on Machine Learning*, pages 20640–20661. PMLR.
- Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xi-angliang Zhang. 2024. [Large language model based multi-agents: A survey of progress and challenges](#). *Preprint*, arXiv:2402.01680.
- Yu Guo, Dong Jin, Shenghao Ye, Shuangwu Chen, Jian Yang, and Xiaobin Tan. 2025. [SQLForge: Synthesizing reliable and diverse data to enhance text-to-SQL reasoning in LLMs](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8441–8452. Association for Computational Linguistics.
- Yu Guo, Shenghao Ye, Shuangwu Chen, Zijian Wen, Tao Zhang, Bai Qirui, Dong Jin, Yunpeng Hou, Huasen He, Jianyang, and Xiaobin Tan. 2026. [Re-thinking table pruning in TableQA: From sequential revisions to gold trajectory-supervised parallel search](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12960–12976. Association for Computational Linguistics.
- Ali Hatamizadeh, Syeda Nahida Akter, Shrimai Prabhumoye, Jan Kautz, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. 2026. [Rlp: Reinforcement as a pretraining objective](#). *Preprint*, arXiv:2510.01265.
- Tao He, Hao Li, Jingchang Chen, Runxuan Liu, Yixin Cao, Lizi Liao, Zihao Zheng, Zheng Chu, Jiafeng Liang, Ming Liu, and Bing Qin. 2025. [Breaking the reasoning barrier a survey on llm complex reasoning through the lens of self-evolution](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7377–7417. Association for Computational Linguistics.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, and 3 others. 2022. [Training compute-optimal large language models](#). In *Advances in Neural Information Processing Systems* 35, NeurIPS 2022, pages 30016–30030. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. [Unnatural instructions: Tuning language models with \(almost\) no human labor](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428. Association for Computational Linguistics.
- Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. [V-STar: Training verifiers for self-taught reasoners](#). In *CoLM*.
- Wentao Hu, Wengyu Zhang, Yiyang Jiang, Chen Jason Zhang, Xiaoyong Wei, and Li Qing. 2025. [Removal of hallucination on hallucination: Debate-augmented rag](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15839–15853. Association for Computational Linguistics.
- Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2026. [Memory in the age of ai agents](#). *Preprint*, arXiv:2512.13564.
- Chen Huang, Wei Lu, and Wenxuan Zhang. 2025. [Pear: Phase entropy aware reward for efficient reasoning](#). *Preprint*, arXiv:2510.08026.
- Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang,

- Haitao Mi, and Dong Yu. 2026. [R-zero: Self-evolving reasoning llm from zero data](#). *Preprint*, arXiv:2508.05004.
- Xinke Jiang, Yue Fang, Rihong Qiu, Haoyu Zhang, Yongxin Xu, Hao Chen, Wentao Zhang, Ruizhe Zhang, Yuchen Fang, Xinyu Ma, Xu Chu, Junfeng Zhao, and Yasha Wang. 2025. [Tc-rag: Turing-complete rag’s case study on medical llm systems](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11400–11426. Association for Computational Linguistics.
- Aditya Kakade, Vivek Srivastava, and Shirish Karande. 2026. [Polaris: A gödel agent framework for small language models through experience-abstracted policy repair](#). *Preprint*, arXiv:2603.23129.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#). *Preprint*, arXiv:2001.08361.
- Dayeon Ki, Rachel Rudinger, Tianyi Zhou, and Marine Carpuat. 2025. [Multiple llm agents debate for equitable cultural alignment](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24841–24877. Association for Computational Linguistics.
- Minsoo Kim, Victor Bursztny, Eunye Koh, Shunan Guo, and Seung-won Hwang. 2024. [Rada: Retrieval-augmented web agent planning with llms](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13511–13525. Association for Computational Linguistics.
- Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2025. [Training language models to self-correct via reinforcement learning](#). In *ICLR*.
- Ivan Lee, Loris D Antoni, and Taylor Berg-Kirkpatrick. 2025. [Good-enough structured generation: A case study on json schema](#). In *Deep Learning for Code in the Agentic Era (DL4C) @ NeurIPS*.
- Haoran Li, Ziyi Su, Yun Xue, Zhiliang Tian, Yiping Song, and Minlie Huang. 2025a. [Advancing collaborative debates with role differentiation through multi-agent reinforcement learning](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22655–22666. Association for Computational Linguistics.
- Ming Li, Lichang Chen, Jiuhai Chen, Shwai He, Jiuxiang Gu, and Tianyi Zhou. 2024a. [Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16189–16211. Association for Computational Linguistics.
- Pengyi Li, Matvey Skripkin, Alexander Zubrey, Andrey Kuznetsov, and Ivan Oseledets. 2025b. [Confidence is all you need: Few-shot rl fine-tuning of language models](#). *Preprint*, arXiv:2506.06395.
- Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2024b. [Self-alignment with instruction back-translation](#). In *ICLR*.
- Yi-Chen Li, Tian Xu, Yang Yu, Xuqin Zhang, Xiong-Hui Chen, Zhongxiang Ling, Ningjing Chao, Lei Yuan, and Zhi-Hua Zhou. 2025c. [Generalist reward models: Found inside large language models](#). *Preprint*, arXiv:2506.23235.
- Xun Liang, Shichao Song, Zifan Zheng, Hanyu Wang, Qingchen Yu, Xunkai Li, Rong-Hua Li, Yi Wang, Zhonghao Wang, Feiyu Xiong, and Zhiyu Li. 2024. [Internal consistency and self-feedback in large language models: A survey](#). *Preprint*, arXiv:2407.14507.
- Jiahang Lin, Shichun Liu, Chengjun Pan, Lizhi Lin, Shihan Dou, Zhiheng Xi, Xuanjing Huang, Hang Yan, Zhenhua Han, Tao Gui, and Yu-Gang Jiang. 2026. [Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses](#). *Preprint*, arXiv:2604.25850.
- Junlin Liu, Shengnan An, Shuang Zhou, Dan Ma, Shixiong Luo, Ying Xie, Yuan Zhang, Wenling Yuan, Yifan Zhou, Xiaoyu Li, Ziwen Wang, Xuezhi Cao, and Xunliang Cai. 2026a. [General365: Benchmarking general reasoning in large language models across diverse and challenging tasks](#). *arXiv preprint arXiv:2604.11778*.
- Junlin Liu, Jiangwang Chen, Zixin Song, Shuaiyu Zhou, Chunji Lv, Hank Wu, Kailin Jiang, Jinyang Wu, Bohan Yu, and Chenxi Zhou. 2026b. [From proprietary to open-source: Bridging the distribution gap via multi-agent protocol distillation in agentic search](#). *arXiv preprint arXiv:2607.24280*.
- Tennison Liu and Mihaela van der Schaar. 2025. [Truly self-improving agents require intrinsic metacognitive learning](#). *Preprint*, arXiv:2506.05109.
- Yule Liu, Heyi Zhang, Jinyi Zheng, Zhen Sun, Zifan Peng, Jiaheng Wei, Tianshuo Cong, Yilong Yang, and Xinlei He. 2025a. [Auditing data membership in reinforcement learning with verifiable rewards](#). *arXiv preprint arXiv:2511.14045*.
- Yule Liu, Jingyi Zheng, Zhen Sun, Zifan Peng, Wenhan Dong, Zeyang Sha, Shiwen Cui, Weiqiang Wang, and Xinlei He. 2025b. [Thought manipulation: External thought can be efficient for large reasoning models](#). *arXiv preprint arXiv:2504.13626*.
- Yule Liu, Zhiyuan Zhong, Yifan Liao, Zhen Sun, Jingyi Zheng, Jiaheng Wei, Qingyuan Gong, Fenghua Tong, Yang Chen, Yang Zhang, and Xinlei He. 2025c. [On](#)

- the generalization and adaptation ability of machine-generated text detectors in academic writing. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 5674–5685.
- Ziyan Liu, Chunxiao Fan, Haoran Lou, Yuexin Wu, and Kaiwei Deng. 2025d. [Mind: A multi-agent framework for zero-shot harmful meme detection](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 923–947. Association for Computational Linguistics.
- Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. [On llms-driven synthetic data generation, curation, and evaluation: A survey](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082. Association for Computational Linguistics.
- Junyu Luo, Bohan Wu, Xiao Luo, Zhiping Xiao, Yiqiao Jin, Rong-Cheng Tu, Nan Yin, Yifan Wang, Jingyang Yuan, Wei Ju, and Ming Zhang. 2025. [A survey on efficient large language model training: From data-centric perspectives](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30904–30920. Association for Computational Linguistics.
- Chunji Lv, Yangguang Wei, Junlin Liu, Yang Gao, Ming Liu, Xinming Wang, Jinyang Wu, Guoren Wang, and Changsheng Li. 2026. [PCSD: Persistent consistency for self-distillation in agentic reinforcement learning](#). *arXiv preprint arXiv:2608.01837*.
- Hao Ma, Tianyi Hu, Zhiqiang Pu, Boyin Liu, Xiaolin Ai, Yanyan Liang, and Min Chen. 2024. [Coevolving with the other you: Fine-tuning llm with sequential cooperative multi-agent reinforcement learning](#). In *Advances in Neural Information Processing Systems 37*, NeurIPS 2024, pages 15497–15525. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#). *Preprint*, arXiv:2303.17651.
- Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Rafael Rafailov, Ivan Laptev, Philip H. S. Torr, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. 2025. [MALT: Improving reasoning with multi-agent LLM training](#). In *CoLM*.
- Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman E. Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. 2025. [Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30215–30248. Association for Computational Linguistics.
- Yulin Peng, Xinxin Zhu, Chenxing Wei, Nianbo Zeng, Leilei Wang, Ying Tiffany He, and F. Richard Yu. 2026. [Sage: Multi-agent self-evolution for llm reasoning](#). *Preprint*, arXiv:2603.15255.
- Cheng Qian, Emre Can Acikgoz, Hongru Wang, Xiusi Chen, Avirup Sil, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. [Smart: Self-aware agent for tool overuse mitigation](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4604–4621. Association for Computational Linguistics.
- Rihong Qiu, Zhibang Yang, Xinke Jiang, Weibin Liao, Xin Gao, Xu Chu, Junfeng Zhao, and Yasha Wang. 2025a. [Bridging global intent with local details: A hierarchical representation approach for semantic validation in text-to-sql](#). *Preprint*, arXiv:2512.22744.
- Yimiao Qiu, Yang Deng, Quanming Yao, Zhimeng Zhang, Zhiang Dong, Chang Yao, and Jingyuan Chen. 2025b. [Think both ways: Teacher-student bidirectional reasoning enhances mcq generation and distractor quality](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8240–8253. Association for Computational Linguistics.
- Noam Razin, Yong Lin, Jiarui Yao, and Sanjeev Arora. 2026. [Why is your language model a poor implicit reward model?](#) *Preprint*, arXiv:2507.07981.
- Marco Robol and Paolo Giorgini. 2026. [Self-evolving software agents](#). *Preprint*, arXiv:2604.27264.
- Haizhou Shi, Ye Liu, Bo Pang, Zeyu Leo Liu, Hao Wang, Silvio Savarese, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2025. [Ssr: Socratic self-refine for large language model reasoning](#). *Preprint*, arXiv:2511.10621.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: language agents with verbal reinforcement learning](#). In *Advances in Neural Information Processing Systems 36*, NeurIPS 2023, pages 8634–8652. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2024a. [The curse of recursion: Training on generated data makes models forget](#). *Preprint*, arXiv:2305.17493.
- Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024b. [Ai models collapse when trained on recursively generated data](#). *Nature*, 631(8022):755–759.
- Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansoor Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. [Large language models for data annotation and synthesis: A survey](#). In *Proceedings of the*

- 2024 *Conference on Empirical Methods in Natural Language Processing*, pages 930–957. Association for Computational Linguistics.
- Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. 2024. [A survey on self-evolution of large language models](#). *Preprint*, arXiv:2404.14387.
- Jen tse Huang, Jiaxu Zhou, Tailin Jin, Xuhui Zhou, Zixi Chen, Wenxuan Wang, Youliang Yuan, Michael Lyu, and Maarten Sap. 2025. [On the resilience of LLM-based multi-agent collaboration with faulty agents](#). In *Proceedings of the 42nd International Conference on Machine Learning*, pages 26202–26226. PMLR.
- Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. [Will we run out of data? limits of llm scaling based on human-generated data](#). *Preprint*, arXiv:2211.04325.
- Boshi Wang, Hao Fang, Jason Eisner, Benjamin Van Durme, and Yu Su. 2024a. [Llms in the imagination: Tool learning through simulated trial and error](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10583–10604. Association for Computational Linguistics.
- Chenxu Wang, Chaozhuo Li, Songyang Liu, Zejian Chen, Jinyu Hou, Ji Qi, Rui Li, Litian Zhang, Qiwei Ye, Zheng Liu, Xu Chen, Xi Zhang, and Philip S. Yu. 2026a. [The devil behind moltbook: Anthropic safety is always vanishing in self-evolving ai societies](#). *Preprint*, arXiv:2602.09877.
- Hanlin Wang, Jian Wang, Chak Tou Leong, and Wenjie Li. 2025a. [Steca: Step-level trajectory calibration for llm agent learning](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11597–11614. Association for Computational Linguistics.
- Haochen Wang, Yi Wu, Daryl Chang, Li Wei, and Lukasz Heldt. 2026b. [Self-evolving recommendation system: End-to-end autonomous model optimization with llm agents](#). *Preprint*, arXiv:2602.10226.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [Self-instruct: Aligning language models with self-generated instructions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508. Association for Computational Linguistics.
- Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. 2024b. [RoboGen: Towards unleashing infinite data for automated robot learning via generative simulation](#). In *Proceedings of the 41st International Conference on Machine Learning*, pages 51936–51983. PMLR.
- Zhaoyang Wang, Weilei He, Zhiyuan Liang, Xuchao Zhang, Chetan Bansal, Ying Wei, Weitong Zhang, and Huaxiu Yao. 2025b. [CREAM: Consistency regularized self-rewarding language models](#). In *ICLR*.
- Zhaotian Weng, Antonis Antoniadis, Deepak Nathani, Zhen Zhang, Xiao Pu, and Xin Eric Wang. 2026. [Group-evolving agents: Open-ended self-improvement via experience sharing](#). *Preprint*, arXiv:2602.04837.
- Xingjian Wu, Junlin Liu, Xingchen Liu, Xuhang Zhu, Jianing Wang, Linsen Guo, Xiaoyu Li, Xuezhi Cao, and Xunliang Cai. 2026. [Contrastive reinforced policy optimization via privileged self-distillation](#). *arXiv preprint arXiv:2607.28026*.
- Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. 2025. [From human memory to ai memory: A survey on memory mechanisms in the era of llms](#). *Preprint*, arXiv:2504.15965.
- Peng Xia, Kaide Zeng, Jiaqi Liu, Can Qin, Fang Wu, Yiyang Zhou, Caiming Xiong, and Huaxiu Yao. 2025. [Agent0: Unleashing self-evolving agents from zero data via tool-integrated reasoning](#). *Preprint*, arXiv:2511.16043.
- Xi Xiao, Xingjian Li, Yunbei Zhang, Cheng Han, Tianming Liu, Tianyang Wang, Runmin Jiang, Jihun Hamm, Xiao Wang, and Min Xu. 2026a. [Layer-specific prompt fusion discovery via differentiable search in vision foundation models](#). *arXiv preprint arXiv:2606.26379*.
- Xi Xiao, Chen Liu, Chih-Ting Liao, Yunbei Zhang, Qizhen Lan, Yuxiang Wei, Lin Zhao, Janet Wang, Jianyang Gu, Muchao Ye, Tianyang Wang, and Hao Xu. 2026b. [Staying VIGILant: Mitigating visual laziness via counterfactual visual alignment in MLLMs](#). *arXiv preprint arXiv:2606.26387*.
- Xi Xiao, Chenrui Ma, Yunbei Zhang, Chen Liu, Zhuxuanzi Wang, Yanshu Li, Lin Zhao, Guosheng Hu, Tianyang Wang, and Hao Xu. 2026c. [Not all directions matter: Towards structured and task-aware low-rank model adaptation](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2132–2154.
- Xi Xiao, Yunbei Zhang, Xingjian Li, Tianyang Wang, Xiao Wang, Yuxiang Wei, Jihun Hamm, and Min Xu. 2025. [Visual instance-aware prompt tuning](#). In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 2880–2889.
- Xi Xiao, Yunbei Zhang, Lin Zhao, Yiyang Liu, Xiaoying Liao, Zheda Mai, Xingjian Li, Xiao Wang, Hao Xu, Jihun Hamm, Xue Lin, Min Xu, Qifan Wang, Tianyang Wang, and Cheng Han. 2026d. [Prompt-based adaptation in large-scale vision models: A survey](#). *Transactions on Machine Learning Research*.

- Can Xie, Ruotong Pan, Xiangyu Wu, Yunfei Zhang, Jiayi Fu, Tingting Gao, and Guorui Zhou. 2026a. [Unlocking exploration in RLVR: Uncertainty-aware advantage shaping for deeper reasoning](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 19057–19076.
- Can Xie, Yuyi Zhou, Wen Yang, Ziyi Zhang, Siyao Song, Yingzhuo Deng, Shuo Ren, and Jiajun Zhang. 2026b. [EDGE: Experience-distillation for guided exploration in agentic reinforcement learning](#). *Preprint*, arXiv:2608.21946.
- Zhibang Yang, Xinke Jiang, Rihong Qiu, Ruiqing Li, Yihang Zhang, Yue Fang, Yongxin Xu, Hongxin Ding, Xu Chu, Junfeng Zhao, and Yasha Wang. 2026a. [DFAMS: Dynamic-flow guided federated alignment based multi-prototype search](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24728–24752. Association for Computational Linguistics.
- Zhibang Yang, Xinke Jiang, Yuzhen Xiao, Ruizhe Zhang, Yue Fang, XinFei Wan, Zhengxing Song, Yuxuan Liu, Yuheng Huang, Xu Chu, Junfeng Zhao, and Yasha Wang. 2026b. [Scaffoldagent: Utility-guided dynamic outline optimization for open-ended deep research](#). *Preprint*, arXiv:2606.20122.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. [React: Synergizing reasoning and acting in language models](#). In *ICLR*.
- Shenghao Ye, Yu Guo, Dong Jin, Yuxiang Wang, Yikai Shen, Yunpeng Hou, Shuangwu Chen, Jianyang, and Xiaofeng Jiang. 2026a. [When TableQA meets noise: A dual denoising framework for complex questions and large-scale tables](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24022–24045. Association for Computational Linguistics.
- Shenghao Ye, Yu Guo, Zhengheng Li, Shuangwu Chen, and Jian Yang. 2026b. [Rubric-guided process reward for stepwise model routing](#). *Preprint*, arXiv:2605.29310.
- Shenghao Ye, Yuxiang Wang, Yu Guo, Dong Jin, Shuangwu Chen, and Jian Yang. 2026c. [Rethinking stepwise model routing: A cost-efficient table reasoning perspective](#). *Preprint*, arXiv:2605.29319.
- Xunjian Yin, Xinyi Wang, Liangming Pan, Li Lin, Xiaojun Wan, and William Yang Wang. 2025. [Gödel agent: A self-referential agent framework for recursive self-improvement](#). *Preprint*, arXiv:2410.04444.
- Cunxi Yu and Haoxing Ren. 2026. [Autonomous evolution of eda tools: Multi-agent self-evolved abc](#). *Preprint*, arXiv:2604.15082.
- Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. 2025a. [Rlpr: Extrapolating rlvr to general domains without verifiers](#). *Preprint*, arXiv:2506.18254.
- Ye Yu, Xiaopeng Yuan, Haibo Jin, Heming Liu, Yaoning Yu, and Haohan Wang. 2026. [Do self-evolving agents forget? capability degradation and preservation in lifelong llm agent adaptation](#). *Preprint*, arXiv:2605.09315.
- Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuewei Wang, Suchin Gururangan, Chao Zhang, Melanie Kambadur, Dhruv Mahajan, and Rui Hou. 2025b. [Self-generated critiques boost reward modeling for language models](#). 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 11499–11514. Association for Computational Linguistics.
- Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Dongsheng Li, and Deqing Yang. 2025. [Evoagent: Towards automatic multi-agent generation via evolutionary algorithms](#). 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 6192–6217. Association for Computational Linguistics.
- Zhenrui Yue, Kartikeya Upasani, Xianjun Yang, Suyu Ge, Shaoliang Nie, Yuning Mao, Zhe Liu, and Dong Wang. 2026. [Dr. zero: Self-evolving search agents without training data](#). *Preprint*, arXiv:2601.07055.
- Dengjia Zhang, Xiaoou Liu, Lu Cheng, Yaqing Wang, Kenton Murray, and Hua Wei. 2026a. [Selaurl: Self evolving llm agent via uncertainty-aware rewards](#). *Preprint*, arXiv:2602.21158.
- Di Zhang. 2026. [Agentdevel: Reframing self-evolving llm agents as release engineering](#). *Preprint*, arXiv:2601.04620.
- Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. 2026b. [Darwin godel machine: Open-ended evolution of self-improving agents](#). *Preprint*, arXiv:2505.22954.
- Jenny Zhang, Bingchen Zhao, Wannan Yang, Jakob Foerster, Jeff Clune, Minqi Jiang, Sam Devlin, and Tatiana Shavrina. 2026c. [Hyperagents](#). *Preprint*, arXiv:2603.19461.
- Qifan Zhang, Dongyang Ma, Tianqing Fang, Jia Li, Jing Tang, Nuo Chen, Haitao Mi, and Yan Wang. 2026d. [Training llm agents for spontaneous, reward-free self-evolution via world knowledge exploration](#). *Preprint*, arXiv:2604.18131.
- Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. 2025a. [Right question is already half the answer: Fully unsupervised LLM reasoning incentivization](#). In *Advances in Neural Information Processing Systems 38*, NeurIPS 2025.

- Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Ruizhe Zhang, Xinke Jiang, Zhibang Yang, Zhixin Zhang, Jiaran Gao, Yuzhen Xiao, Tao Feng, Yue Fang, Yuxuan Liu, Ruiqing Li, Hongbin Lai, Huheng Huang, Xu Chu, Junfeng Zhao, and Yasha Wang. 2026e. [Stackplanner: A centralized hierarchical multi-agent system with task-experience memory management](#). *Preprint*, arXiv:2601.05890.
- Shimao Zhang, Xiao Liu, Xin Zhang, Junxiao Liu, Zheheng Luo, Shujian Huang, and Yeyun Gong. 2025b. [Process-based self-rewarding language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18097–18110. Association for Computational Linguistics.
- Wentao Zhang, Zhe Zhao, Haibin Wen, Yingcheng Wu, Cankun Guo, Ming Yin, and Bo An. 2026f. [Auto-genesis: A self-evolving agent protocol](#). *Preprint*, arXiv:2604.15034.
- Yanzhi Zhang, Zhaoxi Zhang, Haoxiang Guan, Yilin Cheng, Yitong Duan, Chen Wang, Yue Wang, Shuxin Zheng, and Jiyan He. 2025c. [No free lunch: Rethinking internal feedback for llm reasoning](#). *Preprint*, arXiv:2506.17219.
- Yaolun Zhang, Yiran Wu, Yijiong Yu, Qingyun Wu, and Huazheng Wang. 2026g. [Live-evo: Online evolution of agentic memory from continuous feedback](#). *Preprint*, arXiv:2602.02369.
- Yuxiang Zhang, Jing Chen, Junjie Wang, Yaxin Liu, Cheng Yang, Chufan Shi, Xinyu Zhu, Zihao Lin, Hanwen Wan, Yujiu Yang, Tetsuya Sakai, Tian Feng, and Hayato Yamana. 2024. [Toolbehonest: A multi-level hallucination diagnostic benchmark for tool-augmented large language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11388–11422. Association for Computational Linguistics.
- Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025d. [A survey on the memory mechanism of large language model-based agents](#). *ACM Transactions on Information Systems*, 43(6):1–47.
- Zhixin Zhang, Xinke Jiang, Zhibang Yang, Weixuan Xu, Guohong Qiu, Xu Chu, Junfeng Zhao, and Yasha Wang. 2026h. [Loongreflect: Boosting long-horizon reflection in search agents via global perspective distillation](#). *Preprint*, arXiv:2608.11967.
- Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. 2025. [Absolute zero: Reinforced self-play reasoning with zero data](#). In *Advances in Neural Information Processing Systems 38*, NeurIPS 2025. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Weixiang Zhao, Yingshuo Wang, Yichen Zhang, Yang Deng, Yanyan Zhao, Wanxiang Che, Bing Qin, and Ting Liu. 2026a. [Large language model agents are not always faithful self-evolvers](#). *Preprint*, arXiv:2601.22436.
- Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. 2026b. [Learning to reason without external rewards](#). *Preprint*, arXiv:2505.19590.
- Yixiao Zhou, Dongzhou Cheng, Zhiliang Wu, Yi Yang, Yu Cheng, and Hehe Fan. 2026a. [One refiner to unlock them all: Inference-time reasoning elicitation via reinforcement query refinement](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 38957–38978.
- Yixiao Zhou, Yang Li, Dongzhou Cheng, Hehe Fan, and Yu Cheng. 2026b. [Look inward to explore outward: Learning temperature policy from LLM internal states via hierarchical RL](#). *arXiv preprint arXiv:2602.13035*.
- Yixiao Zhou, Ziyu Zhao, Dongzhou Cheng, Zhiliang Wu, Jie Gui, Yi Yang, Fei Wu, Yu Cheng, and Hehe Fan. 2025. [Dropping experts, recombining neurons: Retraining-free pruning for sparse mixture-of-experts LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15169–15186.
- Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. 2025. [TTRL: Test-time reinforcement learning](#). In *Advances in Neural Information Processing Systems 38*, NeurIPS 2025. Neural Information Processing Systems Foundation, Inc. (NeurIPS).
- Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. 2025. [Self-adapting language models](#). In *Advances in Neural Information Processing Systems 38*, NeurIPS 2025. Neural Information Processing Systems Foundation, Inc. (NeurIPS).

A Definition and Taxonomy

A.1 Definition and Rules

Definition. Endogenous observations are learning-relevant signals that arise from an agent’s own generation, evaluation, and interaction processes rather than from labels fixed before the agent acts. This framing covers self-generated tasks and pseudo-labels (Wang et al., 2023; Li et al., 2024b), self-reflection and repair traces (Shinn et al., 2023; Madaan et al., 2023), peer feedback and group judgment (Chen et al., 2025b; Chan et al., 2024), and execution- or tool-grounded traces (Dou et al., 2024; Yao et al., 2023). An observation is *endogenous* for agent self-evolution when it is elicited by

the agent’s own action and then reused by the same learning loop. Let an agent with policy π_θ produce an action or artifact $a_t \in \mathcal{A}$ under context c_t . An observation $o_t \in \mathcal{O}$ is endogenous when

$$o_t = f(a_t, c_t, s_t), \quad u_t = g(o_t, a_t, c_t), \quad (1)$$

where f is an observation channel, s_t denotes the substrate that makes the observation possible, and g converts the observation into a reusable update signal such as a reward, preference pair, pseudo-label, memory entry, verifier trace, or curriculum item. The definition has three requirements: the signal is contingent on the agent’s action, the loop reuses the signal to change future behavior, and the signal is not a static label attached before the agent acts.

Boundary of the concept. The boundary starts from the event that creates the signal, not from where the substrate is located. A human-written label supplied before the agent acts is exogenous supervision. A model-generated critique that is accepted into a repair loop is endogenous only if the critique is conditioned on the agent’s artifact and affects a later action. A compiler, retrieval corpus, search API, simulator, or user environment can be external while still grounding an endogenous observation: the program, query, rollout, or interaction is chosen by the agent, and the resulting trace becomes usable only after the loop converts it into an update. The key distinction is therefore *elicitation*: external content does not become endogenous merely by being present, and external grounding does not make an action-contingent trace exogenous.

Observation channels. With this boundary fixed, the taxonomy can be read as three observation channels. First, *model-internal observations* arise solely from a model’s internal computation and generation, including consistency cues, reflective traces, pseudo-labels, self-constructed objectives, and self-evaluative signals. Second, *peer-induced observations* arise from interactions among multiple models or agents, including peer feedback, questioning, counterexample generation, adjudication, and population-level selection. Third, *scenario-grounded observations* arise when a model interacts with tools, executable environments, validators, retrieval systems, structured workflows, simulators, users, or memory substrates. These are analytical categories rather than an assumed development path. A single system may combine

all three, but the distinction matters because each channel has different costs, failure modes, and safeguards.

Observation contracts. Figure 5 is not meant to classify methods by optimizer name. The same optimizer can be well grounded or circular depending on what it observes. We therefore code methods by four questions: what agent action made the signal observable, what substrate grounded the signal, how the raw trace was converted into an update, and what failure mode threatens that conversion. This contract connects the definition to the rest of the survey. Section 2 studies model-internal signals that are cheap but vulnerable to self-confirmation. Section 3 studies peer-induced signals that can expose blind spots but require diversity and credit assignment. Section 4 studies scenario-grounded signals that are more grounded but vulnerable to proxy gaming and lifecycle drift.

A.2 Taxonomy

Figure 5 provides the full leaf-level taxonomy corresponding to the three observation channels summarized in the main text.

B Survey Design

B.1 Collection and Coding

Collection. We consolidated work indexed by the ACL Anthology, arXiv, OpenReview, and Crossref-indexed venues through August 28, 2026. Query families combined *agent* or *language model* with *self-evolution*, *self-improvement*, *self-feedback*, *reflection*, *self-play*, *multi-agent learning*, *tool learning*, *execution feedback*, *trajectory*, and *memory*; backward and forward citation tracing expanded the pool. We deduplicated by DOI, arXiv identifier, and normalized title, screened titles and abstracts, and then inspected full texts for the observation and update mechanism. The final bibliography contains 131 sources: 96 primary method papers represented in the taxonomy and 35 contextual sources used for definitions, risks, evaluation, and boundary decisions.

Screening. We include a method when an agent action elicits a signal and that signal is reused to alter a later action, memory state, data item, objective, or policy. We exclude static supervision, benchmark-only studies, and architecture or adaptation methods without such reuse. Coding operates on an individual observation rather than a whole system: a focal model’s uncertainty or cri-

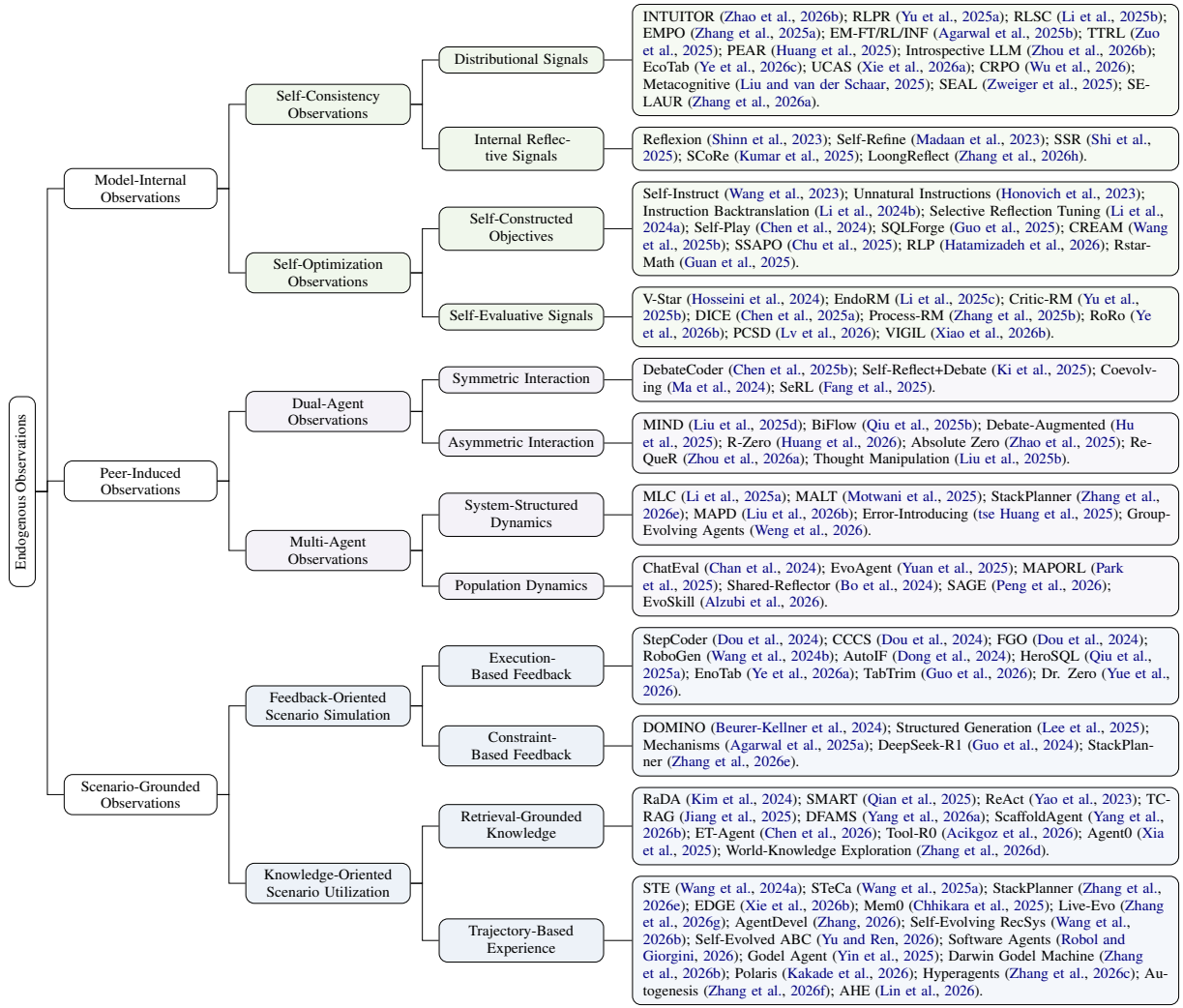


Figure 5: A systematic taxonomy tree of endogenous observations. The tree follows the order from self to social to world: model-internal, peer-induced, and scenario-grounded observations.

technique is model-internal; another model’s critique, test, or rank is peer-induced; and a compiler result, retrieved document, or simulator trace is scenario-grounded. A hybrid system may therefore receive multiple codes. For example, a peer-proposed test is peer-induced, whereas the resulting failed assertion is scenario-grounded. We record the eliciting action, grounding substrate, reused object, update interface, and principal invalidation risk for each primary work.

Reference integrity. For the camera-ready corpus, titles and bibliographic identities were checked against an authoritative landing page or registry (ACL Anthology, DOI/Crossref, arXiv, OpenReview, or publisher page). Recent preprints are used to identify emerging mechanisms rather than establish durable superiority; quantitative comparisons are made only when tasks, models, metrics, and budgets are compatible.

B.2 Positioning

Existing surveys on agent self-evolution are typically organized around *evolutionary processes or system structures* (Alismail and Lanquillon, 2025; Tan et al., 2024; Luo et al., 2025), for example by decomposing self-evolution into stages (Tao et al., 2024), or by analyzing which elements evolve and through which algorithms or system mechanisms (Gao et al., 2026; Alismail and Lanquillon, 2025; Tan et al., 2024; Luo et al., 2025). Our survey instead begins from the observational basis of self-evolution. The distinction has three aspects:

- Unit of analysis.** We classify the individual observation reused by a learning loop rather than the complete system, optimizer, or training stage.
- Shared comparison dimensions.** We compare heterogeneous signals by provenance, ground-

ing, acquisition cost, conversion interface, and invalidation risk.

3. **Design implications.** Treating self-evolution as expansion of an agent’s observable world exposes questions of information quality, long-term stability, and cross-channel composition that component taxonomies leave implicit.

B.3 Boundary Cases

Work that changes model capacity or evaluates outputs without feeding an action-contingent observation back into the learning loop is contextual rather than part of the taxonomy. This includes general evaluation surveys and reasoning benchmarks (Chang et al., 2024; An et al., 2025; Liu et al., 2026a), data-membership and machine-generated-text auditing (Liu et al., 2025a,c), retraining-free expert pruning (Zhou et al., 2025), and low-rank or prompt adaptation (Chang et al., 2026, 2025; Xiao et al., 2026a,c,d, 2025). Counterfactual visual alignment (Xiao et al., 2026b) constructs a reusable internal signal and is included as an adjacent multimodal instance of model-internal observation. These decisions prevent topical proximity from being mistaken for an endogenous observation loop.

C Analysis and Guidance

C.1 Cases and Quality

Observation quality. Quality is not one-dimensional. *Validity* asks whether a signal is correct or faithfully grounded; *novelty* asks whether it adds information beyond existing traces; *diversity* measures coverage across failures, peers, and trajectories; and *informativeness* estimates the signal’s expected downstream learning value. A valid but redundant observation wastes budget, homogeneous peer feedback can amplify correlated errors, and stale retrieval can turn once-valid evidence into persistent bias. Conversely, a failed trajectory may be highly informative when retained as a hard negative. Evaluations should therefore report acceptance rates and fixed observation budgets, then ablate quality filters and individual channels rather than attributing gains to extra sampling alone.

C.2 Channel Trade-offs

Model-internal conversion. Agreement may become a rejection filter, intrinsic reward, preference signal, or test-time adaptation trigger; critiques may

become memory, repaired demonstrations, or process rewards; and generated tasks may become SFT data or curricula. Because this channel is weakly grounded, calibration anchors, diversity filtering, and generator–evaluator separation are important safeguards. Recursive training can narrow distributions and erase tail knowledge (Shumailov et al., 2024a,b).

Peer-induced credit. Peer evidence is relational: a critique, counterexample, rank, disagreement, or revision. A protocol should record the artifact, challenge, decision rule, and resulting revision; otherwise, group rewards may reinforce persuasive style rather than error discovery. Model family, role prompt, memory, and incentive diversity should be treated as experimental variables, especially when group norms can drift (Wang et al., 2026a).

Scenario-grounded governance. Tools, corpora, tests, simulators, and memories can change after joining the learning loop. Reports should track versions, retrieval timestamps, memory eviction, conflict resolution, and rollback. Strong grounding also creates strong proxy pressure: visible checks may be optimized without matching task intent (Zhao et al., 2026a; Yu et al., 2026).

Composition. Model-internal observations are cheap and scalable but weakly grounded; peer-induced observations expose blind spots but require diversity and credit assignment; scenario-grounded observations provide stronger evidence but introduce proxy gaming and lifecycle drift. Practical systems can escalate across these channels and should report which channel produced each accepted observation.

C.3 Coding Protocol

For each method, we record the eliciting action, substrate, object promoted into learning or memory, and invalidation risk. This separates observation provenance from optimizer choice, since the same optimizer can use confidence, peer critique, execution feedback, or retrieved evidence (Liang et al., 2024; Chen et al., 2025b; Dou et al., 2024; Yao et al., 2023).

C.4 Worked Example

Consider a code-generation agent facing edge cases. Disagreement among its own sampled programs is model-internal and may trigger repair, rejection sampling, or intrinsic reward (Liang et al., 2024;

Case	Observation source	Grounding / cost	Update interface	Task and evidence	Principal risk
INTUITOR (Zhao et al., 2026b)	Model confidence and KL statistics.	Weak / low	Intrinsic RL reward	Mathematical reasoning; source-reported label-free gains.	Confidence may not track correctness.
Self-Refine (Madaan et al., 2023)	Self-written language critique.	Weak / low-medium	Iterative test-time revision	Seven generation and reasoning tasks; source-reported improvements.	Generator and critic share blind spots.
DebateCoder (Chen et al., 2025b)	Peer objections and generated tests.	Medium / high	Code revision and selection	Code generation benchmarks; execution-scored outcomes.	Correlated agents and extra calls.
ChatEval (Chan et al., 2024)	Multi-agent debate and judgments.	Medium / high	Aggregated evaluation	LLM evaluation benchmarks; agreement with reference judgments.	Consensus is not semantic correctness.
StepCoder (Dou et al., 2024)	Compiler and unit-test outcomes.	Strong / medium-high	Fine-grained RL reward	Code generation; executable pass rates.	Sparse rewards and incomplete tests.
STeCa (Wang et al., 2025a)	Failures, repairs, and trajectories.	Medium-strong / high	Trajectory calibration and reuse	Tool-agent benchmarks; source-reported task success.	Rollout, credit, and memory costs.

Table 2: Representative observation-to-update loops. Cross-task numerical ranking is avoided because experimental settings are not comparable.

Coding question	Model-internal	Peer-induced	Scenario-grounded
What creates it?	Sampling, self-critique, revision, task generation, or evaluator invocation.	Exposure to critics, challengers, judges, teachers, or populations.	Execution, tool calls, retrieval, simulation, memory writes, or user feedback.
What exposes it?	Distributions, generated language, latent evaluators, or model-written tasks.	Peer models, role prompts, debate protocols, and shared or isolated memories.	Tests, compilers, APIs, corpora, validators, simulators, users, or memory.
What is reused?	Filtered samples, repairs, pseudo-labels, process scores, or curricula.	Critiques, counterexamples, ranks, preferences, group rewards, or revisions.	Verifier results, tool traces, evidence, trajectories, rewards, or memory updates.
What can fail?	Confidence amplification, duplicates, evaluator drift, or diversity collapse.	Correlated bias, collusion, judge dependence, or weak credit assignment.	Reward gaming, stale evidence, incomplete coverage, API drift, or unsafe persistence.

Table 3: Operational coding protocol for endogenous observations.

Shinn et al., 2023; Madaan et al., 2023). A critic model’s adversarial input is peer-induced (Chen et al., 2025b); executing that input and observing a failed assertion is scenario-grounded (Dou et al., 2024; Yao et al., 2023). The failed assertion can then become a scalar reward, supervised repair, preference pair, or memory item. Recording this conversion prevents a grounded trace from being mistaken for a well-grounded objective.

C.5 Update Interfaces

The same observation can support different interfaces. Execution failure, for example, may become a reward, repair example, preference pair, or memory item; reports should therefore name both the observation channel and conversion interface.

C.6 Evaluation

Evaluation should test whether a loop discovers useful information, rejects misleading signals, and remains stable over repeated cycles. We recommend counterfactual observation tests, hidden verifier-coverage splits, peer-diversity ablations, memory and evidence aging tests, and channel-

Interface	Audit before update
Selection and filtering	Preserve hard and diverse cases; avoid retaining only high-confidence traces.
Supervised compression	State the acceptance rule and errors inherited from accepted traces.
Preference or reward optimization	Test whether signals track task intent beyond judge or verifier shortcuts.
State or memory update	Record provenance, versioning, eviction, conflict handling, and rollback.

Table 4: Audit checklist for converting observations into learning pressure.

attribution ablations. Reports should include candidate counts, rejection reasons, accepted observation types, hidden-test coverage, evidence provenance, memory growth and eviction, observation budgets, stopping rules, and repeated-cycle failures.