SocraticAgent: An Autonomous Agent for Unlocking Latent Knowledge in LLMs

Yang Yan
Zhejiang University & Westlake University
Hangzhou, China
yan.yang@zju.edu.cn

Yu Lu
Westlake University
Hangzhou, China
luyu@westlake.edu.cn

Renjun Xu
Zhejiang University
Hangzhou, China
rux@zju.edu.cn

Zhenzhong Lan
Westlake University
Hangzhou, China
lanzhenzhong@westlake.edu.cn

ABSTRACT

Reasoning failures in Large Language Models (LLMs) used by autonomous agents are often attributed to knowledge deficits, leading to a reliance on solutions like Retrieval-Augmented Generation (RAG) or parametric fine-tuning. This paper empirically demonstrates that this assumption is often flawed. We identify a quantifiable "knowledge recall gap": while modern LLMs possess 90-97% of the necessary facts for a task, they spontaneously apply only 57-64% of this knowledge during reasoning. This reveals a significant performance gap rooted in a failure of recall, not a fundamental absence of knowledge. To address this, we introduce SocraticAgent, a zero-shot autonomous agent that emulates Socratic inquiry by guiding an LLM to first deconstruct a problem and comprehensively detail the internal knowledge required for its solution. Through a deterministic two-action cycle of (1) knowledge deconstruction and (2) grounded reasoning, it procedurally closes this recall gap without any model updates. Across a diverse suite of LLMs, SocraticAgent significantly improves reasoning accuracy, outperforming standard prompting and noisy external retrieval. Critically, our agentic, process-driven approach achieves performance competitive with expensive, data-dependent fine-tuning methods, but does so at inference time without any parametric changes. Our work demonstrates that a deliberative agentic process can serve as a powerful substitute for parametric memory adaptation. This paves the way for adaptable, capable autonomous reasoning systems, positioning agent-driven deliberation as a key mechanism for unlocking latent knowledge within LLMs. Code and prompts are available at https://github.com/kuri-leo/BigFive-LLM-Predictor.

KEYWORDS

Autonomous Agents; Agent Architectures; Large Language Models; Knowledge Representation; Automated Reasoning

ACM Reference Format:
Yang Yan, Yu Lu, Renjun Xu, and Zhenzhong Lan. 2026. SocraticAgent: An Autonomous Agent for Unlocking Latent Knowledge in LLMs. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/XGHC3223

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/XGHC3223

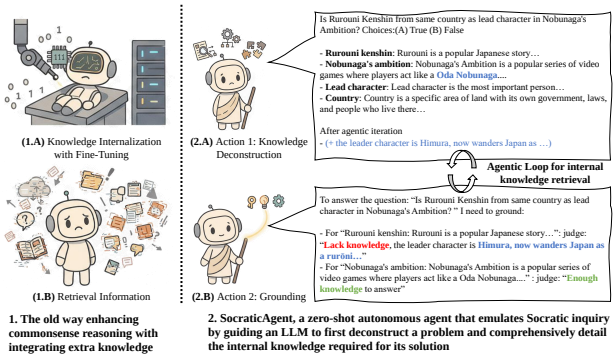

Figure 1: SocraticAgent's procedural approach versus conventional knowledge enhancement. *(1) Conventional methods* attempt to fix knowledge gaps through *(1.A)* resource-intensive fine-tuning or *(1.B)* potentially noisy external retrieval. *(2) SocraticAgent*, in contrast, orchestrates a zero-shot internal recall loop. First, during *(2.A) Knowledge Deconstruction*, it iteratively surfaces the latent facts (blue text) required for the solution. Second, during *(2.B) Grounded Reasoning*, the agent performs a critical self-evaluation: it judges its recalled knowledge, identifying where it is insufficient (red text) and augmenting its context by recalling more specific facts it knows (e.g., adding the blue text about 'Himura'). It proceeds only when it confirms its knowledge is sufficient (green text), ensuring the final answer is constructed from a complete, self-recalled, and verified context.

1 INTRODUCTION

The integration of Large Language Models (LLMs) into autonomous agents has unlocked unprecedented capabilities in complex decision-making. A ubiquitous response to knowledge-intensive task failures is Retrieval-Augmented Generation (RAG) [11], an agent action that injects external facts to fill knowledge gaps. The success of RAG in mitigating issues like hallucination and outdated knowledge has cemented its role in the field [4, 6, 21]. However, its ubiquity is built on a potentially flawed assumption: that reasoning failures

stem from knowledge deficits. This paper argues that the focus on *external* knowledge deficits has obscured a more fundamental, counterintuitive problem: a failure of *internal* knowledge recall.

We empirically diagnose a systemic gap between an LLM's latent knowledge possession and its practical application. Through a systematic analysis of commonsense reasoning benchmarks, we uncover a striking paradox. A *Knowledge Possession Check* reveals that modern LLMs, from 3B to 72B parameters, can articulate the isolated facts needed to solve a problem with high accuracy, often reaching 90-97% for capable models. The knowledge is demonstrably present within the agent's core model. Yet, a subsequent *Knowledge Coverage Analysis* shows that during a standard Chain-of-Thought (CoT) [29] process, these same models spontaneously utilize only 57% to 64% of this internal knowledge. This reveals a stark, roughly 30-percentage-point "knowledge recall gap," providing quantitative proof that a critical bottleneck is often a failure of recall, not a fundamental absence of knowledge.

This diagnosis reframes the problem from knowledge injection to knowledge access, motivating our core research questions:

RQ1 Causality: Is the performance gap in complex reasoning primarily caused by a failure of internal knowledge recall, rather than an absence of knowledge within the model?

RQ2 Procedural Solution: Can a zero-shot autonomous agent, operating purely at inference time, systematically close this recall gap by governing the LLM's reasoning process?

To investigate this, we pivot from costly and often brittle parametric solutions, such as fine-tuning [9] or direct knowledge editing [2, 18, 33], which can risk unintended side effects [16, 22]. Instead, we propose a flexible, agent-based approach. We introduce **SocraticAgent**, a novel autonomous agent whose core strategy is to govern the LLM's internal cognitive process, as shown in Figure 1. Inspired by Socratic dialogue, the agent employs a deterministic policy over a discrete set of two cognitive actions. First, a *knowledge deconstruction* action prompts the LLM to introspect and externalize relevant facts from its latent memory. Second, a *grounded reasoning* action compels the LLM to construct its final answer by strictly adhering to this self-recalled knowledge. This agentic scaffolding distinguishes our work from other multi-step reasoning frameworks [28, 30, 31, 34] by focusing purely on manipulating the agent's internal cognitive state.

Our comprehensive evaluation demonstrates that this agentic process effectively closes the recall gap. SocraticAgent consistently and significantly improves reasoning performance across a diverse suite of LLMs. More importantly, our approach provides a highly efficient and general alternative to dominant paradigms. The agent's zero-shot process achieves performance competitive with state-of-the-art fine-tuning methods [9] without any parametric updates or retraining costs. Most strikingly, our results reveal a profound insight into the power of agentic process: for a smaller model like `Qwen2.5-1.5B`, the structured deliberation imposed by SocraticAgent achieves a mean accuracy of 76.55%, perfectly matching the 76.55% gained from an oracle-RAG system that provides a flawless, externally-curated answer key. For an agent with brittle reasoning, a superior internal process can be as valuable as perfect external knowledge. This finding is particularly salient as RAG is not a panacea, facing challenges from noisy retrieval [32] and context utilization issues [12, 13, 15], underscoring the value of first unlocking an agent's internal potential.

In summary, this paper makes the following contributions:

(1) We provide the first empirical quantification of the 'knowledge recall gap' in LLMs, revealing the significant disparity between latent knowledge and its application in reasoning.
(2) We introduce SocraticAgent, a zero-shot agent that improves reasoning by governing the LLM's internal loop of knowledge deconstruction and grounded synthesis.
(3) Comprehensive experiments demonstrate that this resource-efficient process achieves performance competitive with expensive fine-tuning and, for some models, matches oracle-level external retrieval.

2 RELATED WORK

Our research intersects three key areas: knowledge augmentation for LLMs, internal knowledge unlocking, and agent reasoning.

2.1 Knowledge Sources: External vs. Internal

The challenge of grounding an autonomous agent's reasoning in factual knowledge is a central theme in recent AI research. The dominant paradigm for this is RAG [6, 11], which dynamically augments an agent's context with information retrieved from an external, non-parametric knowledge source. This approach has become the standard for equipping agents to interact with the external world, evolving into complex, modular frameworks [6]. However, the effectiveness of RAG is not absolute. Its performance hinges on a brittle retrieval step that can introduce noise or irrelevant information, a significant problem that has spawned fields like advanced context ranking [32]. More fundamentally, an agent's reasoning is not guaranteed even with perfect retrieval. It can suffer from a "tug-of-war" effect where its internal parametric beliefs override the provided external context [4, 21], or be hindered by architectural limitations like "lost in the middle" where it fails to utilize information within a long context window [15]. These persistent issues have fueled a broader debate on the best way to integrate knowledge, with studies showing there is "no silver bullet" in the choice between RAG and long-context models [12, 13]. This motivates a complementary line of inquiry. Instead of focusing on the complexities of the external world, we can focus on the opportunities of the agent's internal world. Recent work in mechanistic interpretability confirms that while factual knowledge is encoded within a model's parameters, it is often not recalled correctly during reasoning [26, 27]. SocraticAgent is designed in direct response to this, sidestepping the issues of external retrieval to provide a procedural solution for internal recall. It acts as a necessary counterpart to RAG, providing a structured mechanism for an agent to first query its own internal, parametric world before deciding to consult the external, non-parametric one.

2.2 Unlocking Internal Knowledge: Agentic Process vs. Parametric Adaptation

While our work is among the first to empirically quantify the recall gap, other research has implicitly targeted this problem through

two competing philosophies: parametric adaptation and process-driven intervention. The parametric paradigm, the incumbent and resource-intensive approach, aims to improve recall by directly modifying a model's weights. This can be done through fine-tuning, which seeks to bake a desired cognitive process into the model by training it with special tokens to separate memory recall from reasoning [9]. A more surgical version is knowledge editing, where methods apply precise updates to model weights to alter factual associations [2, 18, 19, 33]. This direct manipulation, however, is brittle and risks model integrity. Research shows that such edits can cause logical inconsistencies and catastrophic forgetting of related, unedited facts, revealing the deeply entangled nature of knowledge in LLMs [16, 22].

In stark contrast, the process-driven paradigm champions a fundamentally different philosophy. It leaves the model's weights untouched, instead orchestrating a more effective reasoning process at inference time through an agentic control loop. Recent work such as SSRL [5] addresses knowledge utilization through reinforcement learning, enabling models to iteratively refine their internal knowledge access. This philosophy began with techniques like CoT prompting [29] and has evolved into formal agentic architectures [28] and methods that structure implicit knowledge into explicit graphs [8]. SocraticAgent is situated firmly in this latter tradition. It contributes a novel agentic process specifically designed for internal knowledge recall, providing a flexible, general, and non-destructive alternative to the powerful but brittle parametric methods. It achieves this by shifting the locus of control from static model weights to a dynamic, inference-time agentic policy.

2.3 Agent-Driven Reasoning Architectures

The concept of using a structured process to improve LLM reasoning began with CoT prompting [29], which established that making reasoning an explicit process can unlock complex capabilities. This foundational idea was quickly enhanced to be more accessible, using simple phrases to trigger reasoning [10], and more robust, by exploring multiple reasoning paths to select the most consistent answer [25]. This evolution led to a diverse landscape of autonomous agent architectures, which we can categorize by the locus of their primary actions.

One major branch of research focuses on agents whose actions manipulate an **external environment**. Frameworks like ReAct [31] establish an iterative loop of reasoning and acting, where actions involve using external tools to gather information or affect a world state. This line of work has produced advanced systems that empower agents to create their own tools on the fly [1], further deepening the agent's interaction with its external surroundings.

A parallel thread has focused on agents whose actions explore an **internal search space** for complex deliberation and planning. Tree of Thoughts [30] moves beyond linear reasoning chains by having the agent navigate a tree of potential thought paths. This is extended by frameworks like Language Agent Tree Search [35], which integrates planning and search more formally. Similarly, self-improving architectures use actions to drive multi-trial refinement and reflection [17, 23]. In these systems, the agent's goal is to find an optimal path through a vast space of possible reasoning trajectories.

SocraticAgent introduces a third, distinct category, fitting within the broader classification of agentic frameworks discussed in recent surveys [4, 6, 34]. Resonating with recent calls for "reasonable parrots" that engage in argumentative design [20], its actions do not operate on an external environment or a complex search space. Instead, the agent's actions exclusively manipulate its own **internal cognitive state**, specifically the 'recalled knowledge context'. It employs a simple, deterministic two-action policy, not for search, but for cognitive governance. The agent's contribution is therefore foundational. Instead of orchestrating external tool-use or complex planning, SocraticAgent pioneers the use of a simple agentic policy to govern a purely internal cognitive loop: the structured introspection and subsequent application of the model's own latent knowledge.

3 SOCRATICAGENT: AN AUTONOMOUS AGENT FOR INTERNAL RECALL

To address the empirically identified knowledge recall gap, we move beyond simple prompting techniques and propose **SocraticAgent**, an autonomous agent that procedurally structures an LLM's internal reasoning process. The agent operates at inference time, requiring no parametric updates or specialized training data. Its design is grounded in the hypothesis that a deliberate, two-stage cognitive cycle can causally mitigate the recall failures observed in standard, single-pass generation. This section formalizes the agent's architecture and its operational workflow.

3.1 The Agent Model

We formally define SocraticAgent as an autonomous agent $\mathcal{A} = \langle S, A, \pi, P, G \rangle$ operating within the environment of a large language model. This framing allows us to precisely articulate its components and behavior in a manner consistent with established agent theory.

- **Perception (P):** The agent perceives a single input from the user, a question q.
- **State (S):** The agent maintains a simple internal cognitive state, $s \in S$, which is a tuple $s = (q, \mathcal{K})$. Here, q is the initial question, and $\mathcal{K}$ is the *recalled knowledge context*, a set of facts and concepts that is initially empty.
- **Actions (A):** The agent possesses a discrete set of two cognitive actions, $A = \{a_1, a_2\}$:
  - $a_1 = \texttt{Deconstruct}(q)$: An introspection action prompting the LLM to list knowledge points relevant to q. The result updates the agent's state by populating the recalled knowledge $\mathcal{K}$.
  - $a_2 = \texttt{GroundReasoning}(q, \mathcal{K})$: A synthesis action where the agent prompts the LLM to generate the final answer to q, with the explicit constraint that its reasoning must be strictly grounded in the facts contained in the now-populated knowledge context $\mathcal{K}$.
- **Policy (π):** SocraticAgent employs a fixed, deterministic policy, $\pi : S \to A$. Upon perceiving a question q (and thus initializing its state with an empty $\mathcal{K}$), the policy is a simple, non-learning sequence: execute a_1, update state, then execute a_2. This deterministic process ensures reliability and analyzability.
- **Goal (G):** The agent's goal is to produce a final response that is both correct and demonstrably well-reasoned, maximizing the utilization of the LLM's available internal knowledge.

Algorithm 1 The SocraticAgent Deliberation Cycle

Require: Question q
Ensure: Final answer A_{final}
$\mathcal{K} \leftarrow$ Deconstruct(q) ▷ Execute Action 1
$A_{final} \leftarrow$ GroundReasoning($q, \mathcal{K}$) ▷ Execute Action 2
return A_{final}
function Deconstruct(q)
  $C \leftarrow$ RefineConcepts(q) ▷ Iteratively find key concepts
  $\mathcal{E} \leftarrow \emptyset$ ▷ Initialize explanations map
  for each concept $c \in C$ **do**
    $\mathcal{E}[c] \leftarrow$ RefineExplanation(c, q) ▷ Iteratively refine
  end for
  return Aggregate($\mathcal{E}$) ▷ Build knowledge context $\mathcal{K}$
end function
function RefineConcepts(q)
  for $i \leftarrow 1 \ldots$ max_iters **do**
    if $i > 1$ **then**
      $C_{new} \leftarrow$ LLM.Revise($C, crit$)
    else
      $C_{new} \leftarrow$ LLM.Propose(q)
    end if
    $crit \leftarrow$ LLM.Critique(C_{new}, q)
    $C \leftarrow C_{new}$
    if $crit.decision$ = "ACCEPT" **then break**
    end if
  end for
  return C
end function
function RefineExplanation(c, q)
  for $i \leftarrow 1 \ldots$ max_iters **do**
    if $i > 1$ **then**
      $e_{new} \leftarrow$ LLM.Revise($e, crit$)
    else
      $e_{new} \leftarrow$ LLM.Explain(c)
    end if
    $crit \leftarrow$ LLM.Critique(e_{new}, c, q)
    $e \leftarrow e_{new}$
    if $crit.decision$ = "ACCEPT" **then break**
    end if
  end for
  return e
end function
function GroundReasoning($q, \mathcal{K}$)
  $prompt \leftarrow$ "Context:" $\oplus\, \mathcal{K} \oplus$ " Question:" $\oplus\, q$
  return LLM.GenerateAnswer($prompt$)
end function

In this model, the LLM serves as both the agent's primary tool for action execution and its immediate environment. The agent's actions are not physical manipulations but are instead carefully constructed prompts that interact with and shape the LLM's cognitive process. This positions SocraticAgent as an agent designed specifically to govern the internal reasoning dynamics of another system, a key challenge in building more robust and reliable AI.

3.2 Agentic Workflow: A Two-Action Deliberation Cycle

The agent's deterministic policy, π, translates into a simple yet powerful two-action workflow that executes sequentially upon perception of a question q. This cycle, detailed in Algorithm 1, is designed to directly counteract the recall failure mode identified in our initial analysis by transforming a single, probabilistic generation step into a structured, two-stage deliberation.

3.2.1 Action 1: `Deconstruct(q)`. Upon receiving a question q, the agent first executes the deconstruct action. The purpose of this action is to causally intervene in the recall process by forcing the LLM to convert its latent, distributed knowledge into an explicit, symbolic representation within the agent's internal state ($\mathcal{K}$).

Operationally, as detailed in Algorithm 1, this is implemented as an iterative process. The agent first executes `RefineConcepts` to iteratively identify atomic keywords required for the question, followed by `RefineExplanation` to generate standalone definitions for each. Both steps utilize `LLM.Critique` to return structured decisions (ACCEPT/REVISE), ensuring the quality of the recalled facts before populating the knowledge context $\mathcal{K}$. This act compels the LLM to surface the knowledge that our initial analysis showed it possesses but often fails to apply.

3.2.2 Action 2: `GroundReasoning(q, K)`. With the knowledge context $\mathcal{K}$ now populated, the agent executes the grounding action. The objective here is to mitigate the recall failure by ensuring the surfaced knowledge is actively used. It prevents the model from "forgetting" or deviating from critical facts, the primary failure mode identified in our *Knowledge Coverage Analysis*. The agent achieves this by synthesizing a compound prompt containing both the original question q and the full set of self-recalled knowledge points from $\mathcal{K}$. This prompt explicitly instructs the LLM to formulate its final, step-by-step answer by strictly adhering to the provided context. This constrains the LLM's vast reasoning space to a trajectory consistent with its own recalled knowledge, mitigating attentional drift and ensuring that the previously latent facts are integrated into the final reasoning chain. Together, these two actions form a complete, procedural intervention that directly remedies the diagnosed recall gap.

4 EXPERIMENTAL SETUP

To empirically validate our research questions, we designed a comprehensive, two-stage experimental protocol. The first stage diagnoses the knowledge recall gap (RQ1) through a detailed knowledge analysis. The second stage evaluates the efficacy of SocraticAgent in closing this gap (RQ2) by systematically comparing its performance against carefully chosen baselines that represent the primary paradigms in knowledge-intensive reasoning.

4.1 Tasks, Datasets, and Models

Tasks and Datasets. We evaluate our approach on three established benchmarks: **StrategyQA** [7], **CommonSenseQA** [24], and **TruthfulQA** [14]. These were specifically chosen because they demand multi-step reasoning on knowledge that is expected to be latent within a general-purpose LLM, rather than requiring timely or domain-specific external lookup. This selection isolates the internal recall problem from confounding variables like retrieval noise common in web-browsing tasks (e.g., HotpotQA, MuSiQue, or BrowseComp), making them the ideal environments to isolate and study the *internal recall* problem we diagnose. To enable our diagnostic analysis, we first established a ground-truth knowledge base. The canonical knowledge points required for each question

were extracted from the annotated references provided in prior work [9], using our strongest available model, `Qwen2.5-72B-It`, to ensure high-fidelity extraction.

Models Evaluated. To demonstrate the generality and model-agnostic nature of SocraticAgent, our evaluation spans a wide spectrum of prominent open-source and proprietary LLMs:

- **Open-Source Models:** We test models from two leading families: `Qwen2.5` (1.5B, 3B, 7B, 14B, 32B, 72B), the newer `Qwen3-30B-A3B`, and `LLaMA3.1` (8B, 70B). We also include variants of `Qwen2.5-7B` and `LLaMA3.1-8B` that have been fine-tuned on the DeepSeek-R1 reasoning dataset to test SocraticAgent on models already optimized for reasoning.
- **Proprietary Models:** We include `gemini-2.5-flash`, a highly capable model with native web-searching capacity, allowing for a direct comparison between our agent's internal deliberation and real-world external web retrieval.

4.2 Baselines and Evaluation Protocol

Baselines. To situate the contribution of our agent's procedural intervention, we compare it against five distinct baselines representing the dominant paradigms in knowledge-intensive reasoning:

(1) **Vanilla Baseline:** Our primary baseline is standard CoT prompting [29]. This represents the standard generation process where we hypothesize recall failures occur.
(2) **Oracle RAG (Ground Truth References):** To establish a practical upper bound, we provide the model with curated, ground-truth knowledge points from benchmark annotations [9]. This "oracle" baseline simulates a perfect RAG system that enables the model to access all necessary facts without noise or retrieval errors.
(3) **Noisy RAG (Web-Source References):** Compared to the oracle RAG, we further employ web retrieval to simulate a realistic, but potentially noisy, RAG implementation. This is only feasible with `gemini-2.5-flash`, which has native web browsing capabilities.
(4) **Parametric Recall:** Our primary conceptual competitor is a state-of-the-art method that also addresses internal recall but through parametric adaptation [9]. We report results directly from the original work to benchmark our zero-shot agent against this resource-intensive paradigm.
(5) **Process-Driven Refinement:** We additionally compare against inference-time methods including Self-Refine [17] and Multi-Agent Debate [3], which attempt to improve reasoning through iterative critique or consensus without explicit knowledge deconstruction.

Evaluation Protocol. Our protocol is divided into two parts, corresponding to our research questions. All automated evaluations use `Qwen2.5-72B-It` as an LLM-as-judge for consistency and scalability.

To ensure reliability, we manually validated a sample of 762 judgments (approx. 1% of data), achieving a 95.54% agreement rate with the LLM judge. Among 700 human-judged correct and 62 incorrect cases, the judge made 19/700 Type I errors (2.7%) and 15/62 Type II errors (24.2%), confirming the reliability of our metrics.

(1) *Diagnostic Analysis (for RQ1):* We first quantify the knowledge recall gap with two checks:
   - **Knowledge Possession Check:** To probe if an LLM "knows" a fact, we isolate each ground-truth knowledge point and prompt the model for a concise definition in the context of the original question. The LLM-as-judge then compares the model's definition to the ground-truth explanation, yielding a Yes/No verdict on factual correctness.
   - **Knowledge Coverage Analysis:** To measure if an LLM *uses* its knowledge, we prompt it for a standard CoT response. A separate LLM instance then extracts the knowledge points expressed in the response. Finally, the LLM-as-judge compares this extracted set to the ground-truth inventory to calculate a knowledge coverage percentage.
(2) *Agent Performance Evaluation (for RQ2):* To assess the effectiveness of the SocraticAgent's deterministic two-action policy, we measure end-to-end task accuracy on each dataset. The final answer generated by the agent (and each baseline) is compared against the ground-truth label.

5 RESULTS AND ANALYSIS

Our experiments are designed to empirically test our core theses: first, to diagnose and quantify the hypothesized knowledge recall gap (RQ1), and second, to evaluate the efficacy of SocraticAgent's procedural intervention in closing this gap (RQ2). The results validate our claims, showing that a deliberate agentic process unlocks significant performance gains by improving an LLM's access to its own latent knowledge.

5.1 Diagnosis: Quantifying the Chasm Between Knowledge and Recall (RQ1)

Our first objective was to move beyond anecdotal evidence and empirically validate our central premise: that many reasoning failures stem from poor recall, not an absence of knowledge. To do so, we conducted a two-stage diagnostic analysis using ground-truth knowledge points extracted from prior work [9]. The results reveal a stark and consistent paradox.

First, our *Knowledge Possession Check*, visualized in Figure 2, confirms that modern LLMs have successfully encoded the factual knowledge required for these commonsense reasoning tasks. The results are striking: even in a single attempt, models like `LLaMA3.1-8B` and `Qwen2.5-72B` articulate isolated facts with high accuracy (92.8% and 96.7%, respectively). More tellingly, a more robust *Majority Vote @ 4* metric, which assesses correctness over four generations, reveals near-perfect possession rates across all models, ranging from 95.6% to 99.4%. This provides strong evidence that the fundamental knowledge is not just present, but robustly encoded within the models' parameters.

However, the *Knowledge Coverage Analysis* (Table 1) reveals the counterintuitive failure. When prompted for a standard CoT [29] response, these same models spontaneously apply only a fraction of their latent knowledge. For example, on StrategyQA, `LLaMA3.1-8B` utilizes only 62.54% of the necessary knowledge it was just proven to possess (92.8%). This creates a significant, 30-percentage-point chasm between knowledge possession and its application. Troublingly, this is not merely a retrieval problem that better RAG can

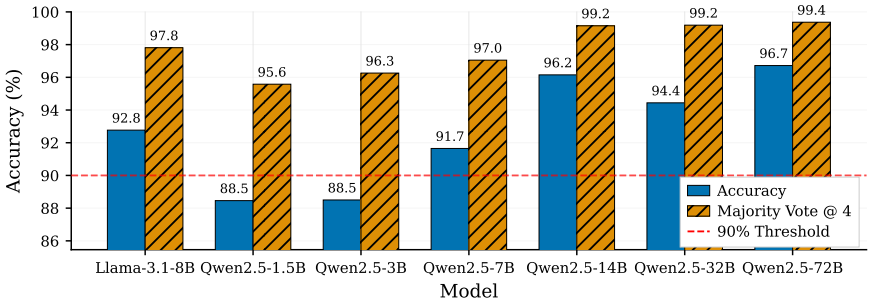

Figure 2: Knowledge Possession Check. Blue and orange bars denote single-attempt and robust "Majority@4" accuracy, respectively. Knowledge is robustly encoded across models, with nearly all exceeding 90% initially and approaching perfect scores with multiple attempts.

Table 1: Knowledge Coverage Analysis. Coverage measures the percentage of necessary knowledge points that a model spontaneously uses in a given reasoning process. The low coverage for vanilla baseline demonstrates a systemic failure to recall existing knowledge, a gap that is not fully closed even with perfect external information (Oracle RAG).

Model	Dataset	Vanilla Baseline Coverage (%)	Oracle RAG Coverage (%)
LLaMA3.1-8B	CommonSenseQA	61.24	72.47
	StrategyQA	62.54	71.73
	TruthfulQA	57.92	67.85
Qwen2.5-7B	CommonSenseQA	59.14	74.42
	StrategyQA	64.15	72.63
	TruthfulQA	59.08	70.40

solve. Even when provided with a perfect external "cheat sheet" via Oracle RAG, coverage for `LLaMA3.1-8B` on StrategyQA improves but remains far from complete at only 71.73%.

These findings provide a direct and quantitative answer to RQ1. The performance bottleneck in complex reasoning can indeed be causally attributed to a failure of internal knowledge recall. This failure is so profound that it persists even when the knowledge is explicitly provided in context, suggesting the root cause is a failure in the reasoning *process* itself. This empirically validates the problem that our SocraticAgent is designed to solve procedurally.

5.2 Intervention: SocraticAgent Systematically Closes the Recall Gap (RQ2)

We evaluated SocraticAgent's ability to mitigate this failure. As shown in Table 2, the agent delivers consistent, significant improvements over the vanilla baseline across diverse models. For instance, it boosts `LLaMA3.1-8B-It` mean accuracy from 65.90% to **76.78%** and `Qwen2.5-7B-It` from 71.11% to **76.14%**.

This success directly validates the agent's causal mechanism. Whereas our diagnostic showed that a standard CoT process utilizes only 57-64% of available knowledge (Table 1), the agent's deterministic two-action policy intervenes directly. The `Deconstruct` action compels the LLM to populate the agent's internal 'recalled knowledge context' $\mathcal{K}$, transforming latent knowledge into an explicit representation. The subsequent `GroundReasoning` action then constrains the LLM's generation to this context, ensuring the surfaced facts are integrated into the final reasoning chain. This procedural scaffolding bridges the diagnosed recall gap, providing a firm, positive answer to RQ2: an autonomous agent can systematically improve reasoning at inference time by guiding an LLM to better utilize its own latent knowledge.

5.3 The Ordeal: Process-Driven Recall vs. Resource-Intensive Paradigms

The true test of SocraticAgent lies not just in outperforming a simple baseline, but in how its procedural approach contends with the field's dominant and resource-intensive paradigms. We conducted a systematic comparison against three key challengers: (*i*) a real-world **Noisy RAG** system using web retrieval, (*ii*) a state-of-the-art **Parametric Specialization** method that fine-tunes models for recall [9], and (*iii*) an **Oracle RAG** system given perfect external knowledge. The results demonstrate the profound efficiency of an agentic process.

First, against noisy, real-world retrieval, the value of focused internal deliberation becomes clear. For `gemini-2.5-flash`, enabling its native web search (Noisy RAG) slightly harmed its mean performance, dropping it from 75.07% to 74.23%, likely due to irrelevant search results distracting the model. In stark contrast, SocraticAgent, which guides the model to query its own internal knowledge, boosted performance to **78.47%**. For commonsense reasoning, clean access to internal knowledge appears more effective than an unguided search of the external world.

Second, against parametric specialization, SocraticAgent proves to be a highly competitive and far more general alternative. On StrategyQA, SocraticAgent with `LLaMA3.1-8B-It` achieves **77.44%** accuracy. This significantly closes the performance gap to the fine-tuned SOTA result of 82.3% from prior work [9], and our agent achieves this at inference time, without any of the data collection, training costs, or model-specific lock-in associated with fine-tuning. This demonstrates that an intelligent process can often be a powerful substitute for brute-force parametric adaptation.

Third, against purely process-driven methods on `Qwen2.5-7B-It`, SocraticAgent (76.14%) outperforms both Self-Refine (69.05%) [17] and Multi-Agent Debate (71.73%) [3]. This confirms that structured

Table 2: Main experimental results showing accuracy (%) on three commonsense reasoning benchmarks. Our proposed method, SocraticAgent, is compared against a vanilla baseline, Oracle RAG ("w/ Ground Truth Ref"), Noisy RAG ("w/ Web Search"), and a SOTA Parametric Fine-Tuning method. SocraticAgent consistently outperforms the vanilla baseline and noisy RAG, and provides a competitive, zero-shot alternative to expensive fine-tuning and oracle-level retrieval.

Model	Method	CommonSenseQA	StrategyQA	TruthfulQA	Mean
SOTA Parametric Method (from [9])					
LLaMA3.1-8B (Fine-Tuned)	Disentangling Reasoning and Knowledge	78.0	82.3	86.6	82.3
Qwen2.5-7B (Fine-Tuned)	Disentangling Reasoning and Knowledge	78.6	83.2	81.2	81.0
Zero-Shot Methods					
Gemini-2.5-Flash	Vanilla Baseline	68.96	76.38	79.88	75.07
	Noisy RAG (w/ Web)	67.40	76.64	78.66	74.23
	Oracle RAG (w/ Ref)	90.75	83.42	92.54	88.90
	SocraticAgent (ours)	**81.63**	**78.66**	**75.12**	**78.47**
LLaMA3.1-8B-It	Vanilla Baseline	72.38	73.26	52.06	65.90
	Oracle RAG (w/ Ref)	87.36	79.86	86.19	84.47
	Self-Refine	71.11	49.32	57.34	59.26
	Multiagent Debate	68.81	50.94	58.71	59.49
	SocraticAgent (ours)	**81.05**	**77.44**	**71.85**	**76.78**
Qwen2.5-7B-It	Vanilla Baseline	79.12	70.98	63.23	71.11
	Oracle RAG (w/ Ref)	89.11	73.81	88.61	83.84
	Self-Refine	77.42	70.53	59.20	69.05
	Multiagent Debate	78.88	71.15	65.17	71.73
	SocraticAgent (ours)	**81.62**	**77.57**	**69.24**	**76.14**
Qwen2.5-1.5B-It	Vanilla Baseline	65.40	59.10	40.87	55.13
	Oracle RAG (w/ Ref)	85.19	71.79	72.68	76.55
	SocraticAgent (ours)	**80.60**	**75.14**	**73.92**	**76.55**
Qwen3-30B-A3B-It-2507	Vanilla Baseline	74.62	79.60	57.34	70.52
	Oracle RAG (w/ Ref)	79.77	82.53	68.53	76.94
	SocraticAgent (ours)	**80.09**	**78.39**	**71.68**	**76.72**

knowledge deconstruction is significantly more effective here than generic iterative refinement or consensus strategies.

Finally, the comparison against oracle-level retrieval reveals our most striking result. For smaller models with less developed native reasoning, the deliberation structure imposed by our agent is remarkably powerful. On Qwen2.5-1.5B-It, the smallest model we tested, SocraticAgent achieves a mean accuracy of **76.55%**, which is identical to the performance achieved by the Oracle RAG baseline (**76.55%**). This finding is profound: for an agent with inherently brittle reasoning, a superior internal *process* can be as valuable as being given a perfect external *answer key*.

5.4 Analysis: Interaction with Model Capabilities

A deeper analysis reveals two crucial insights about how SocraticAgent's procedural scaffolding interacts with a model's inherent capabilities. First, our agentic approach acts as a powerful equalizer, disproportionately empowering smaller, more accessible models. The performance uplift is most pronounced at the lower end of the parameter scale, with a massive +21.42 point improvement for Qwen2.5-1.5B and a +6.2 point improvement for Qwen3-30B-A3B-It-2507 (w/o its reasoning mode). This equalizing effect is so profound that, as shown in Section 5.3, it allows the Qwen2.5-1.5B model to achieve a mean accuracy of **76.55%**, a score identical to that achieved by the Oracle RAG baseline. This finding has significant implications, as it demonstrates that for some tasks, a superior agent architecture can be as effective as providing a small model with a perfect external answer key, offering a path to democratize high-level reasoning.

Second, our experiments reveal a nuanced interaction with models that have already been fine-tuned for reasoning. When we applied SocraticAgent to reasoning-specialized variants of LLaMA3.1 and Qwen2.5. For DeepSeek-R1-Distill-Llama-8B, mean performance improved slightly to 77.25%, suggesting a positive synergy. However, for DeepSeek-R1-Distill-Qwen-7B, performance unexpectedly decreased to 73.90%. We hypothesize this is due to procedural interference: the model's parametrically-encoded reasoning style likely clashed with the explicit two-action process imposed by our agent, leading to a less coherent outcome. This finding is notable because it clarifies the agent's primary role. SocraticAgent's strength lies in its ability to bring structure to general-purpose models at inference time, offering a flexible and cost-effective alternative to, rather than a universal augmentation for, the resource-intensive process of parametric specialization.

6 DISCUSSION AND LIMITATIONS

Our results provide strong empirical evidence that SocraticAgent's procedural intervention systematically closes the knowledge recall gap. More profoundly, the findings illuminate the value of an agent-driven process, challenging prevailing assumptions about how to best elicit robust reasoning from LLMs.

Agentic Process as a General Alternative to Parametric Specialization. A dominant paradigm for unlocking an LLM's internal knowledge involves costly parametric adaptation, where models

are fine-tuned on specialized datasets to bake in recall behaviors. Our findings demonstrate that a zero-shot, agentic process offers a powerful and far more general alternative. As shown in Table 2, SocraticAgent, through its simple, deterministic policy, enables `LLaMA3.1-8B` to close a significant portion of the performance gap with a state-of-the-art fine-tuned model on StrategyQA (77.44% vs. 82.3%). It achieves this without any data curation, compute costs, or loss of generality inherent to fine-tuning. This suggests that for many reasoning tasks, the desired behavior can be elicited through superior cognitive governance at inference time rather than through a more specialized model. This is a crucial finding for the Agent community, as it champions the role of agent architecture and deliberate action sequencing as a primary tool for achieving intelligent behavior, independent of model retraining.

The Power of Structured Internal Deliberation. Our experiments also draw a sharp distinction between different methods of knowledge access. The results from `gemini-2.5-flash` are particularly revealing. Unguided external web search (Noisy RAG) slightly degraded performance to 74.23% mean accuracy, likely due to the introduction of irrelevant information. In contrast, SocraticAgent's internally-focused deliberation, governed by its two-action policy, boosted performance to 78.47%. This highlights a key insight: for commonsense reasoning, where knowledge is often already latent, a structured query of the agent's "inner world" is more effective than a noisy query of the external world. Furthermore, the comparison with the Oracle RAG baseline yields a striking second insight. For the `Qwen2.5-1.5B` model, the procedural scaffolding provided by SocraticAgent achieved a mean accuracy of 76.55%, identical to its performance when provided with a perfect, externally-curated answer key. This result is profound: for an agent with inherently brittle reasoning, a superior internal process can be as valuable as perfect external knowledge. The agent's structured deliberation effectively compensates for its inherent brittleness, offering a path to democratize high-level reasoning for smaller, more efficient models and broadening access to advanced AI capabilities.

Boundaries and Future Agent Architectures. While our approach demonstrates significant promise, its boundaries define fertile ground for future research into more advanced agentic systems.

- **From Fixed Policies to Learned Meta-Policies:** SocraticAgent's efficacy is predicated on knowledge being present within the LLM, a boundary condition validated by our *Knowledge Possession Check*. For tasks requiring timely or private information, RAG remains indispensable. This defines a clear research direction: developing hybrid agents that learn a *meta-policy* to dynamically decide when to execute an internal action like `Deconstruct` versus when to query an external source. Such an agent could optimize its knowledge-gathering strategy based on the specific problem it faces.
- **From Single-Agent Recall to Multi-Agent Dialectics:** The agent's `Deconstruct` action relies on the LLM to faithfully report its knowledge. While our results show the net effect is strongly positive, a risk of hallucination exists. This suggests a compelling extension toward a multi-agent system where a second "critic" agent is tasked with validating or critiquing the knowledge recalled by the first. This would create a more robust, dialectical reasoning process, improving the fidelity of the surfaced knowledge before it is used for grounding.
- **From Deterministic Action to Learned Reasoning Policies:** SocraticAgent employs a fixed, deterministic two-action policy. While simple and effective, this is not necessarily optimal for all problems. Some questions might not require the deconstruction step, while others could benefit from an iterative cycle of recall and reasoning. This opens a compelling avenue for applying reinforcement learning to train an agent that learns its own optimal cognitive policy. The agent could learn to select from a broader set of cognitive actions (e.g., recall, critique, reason, elaborate) to maximize correctness and efficiency, with our current fixed policy serving as a strong initial baseline.

7 CONCLUSION

In this work, we challenged the prevailing assumption that reasoning failures in Large Language Models are primarily caused by knowledge deficits. We provided the first empirical diagnosis and quantification of a "knowledge recall gap," a consistent, significant disparity between an LLM's vast repository of latent knowledge and its ability to apply that knowledge in practice. This finding reframes a significant challenge in agent reasoning from a problem of knowledge injection to one of knowledge access.

To address this gap, we introduced SocraticAgent, a novel, zero-shot autonomous agent that orchestrates an LLM's internal cognitive process. Through a deterministic, two-action policy of knowledge deconstruction and grounded reasoning, our agent procedurally bridges the gap between possession and application. Our comprehensive experiments demonstrate that this agentic intervention not only significantly boosts reasoning performance across a diverse suite of models but also provides an efficient and general alternative to resource-intensive paradigms such as fine-tuning.

The core contribution of this research is the demonstration that an agent's deliberative process can serve as a powerful and effective substitute for parametric memory adaptation and can be more valuable than unguided external retrieval for commonsense reasoning. Our most striking finding is that for smaller agents, the structured deliberation governed by our agent's policy can match the performance benefit of being supplied with a perfect external answer key. This work presents a paradigm shift, positioning agent-driven, procedural interventions as a first-class method for unlocking the immense, untapped potential already stored within LLMs. With this foundation, we envision a new generation of intelligent agents that leverage learned meta-policies, multi-agent dialectics, and adaptive reasoning strategies to achieve robust, reliable, and generalizable intelligence.

ETHICAL CONSIDERATIONS

This study relies solely on public benchmarks (CommonsenseQA, StrategyQA, and TruthfulQA) without involving human subjects or sensitive data. We adhere to all model usage policies and utilized LLMs for language refinement and code debugging.

ACKNOWLEDGMENTS

This work was supported by the Key R&D Program of Zhejiang, 2025C01104.

REFERENCES

[1] Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. 2023. Large Language Models as Tool Makers. In *The Twelfth International Conference on Learning Representations*.

[2] Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing Factual Knowledge in Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 6491–6506. https://doi.org/10.18653/v1/2021.emnlp-main.522

[3] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning* (Vienna, Austria) *(ICML'24)*. JMLR.org, Article 467, 31 pages.

[4] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*. Association for Computing Machinery, New York, NY, USA, 6491–6501. https://doi.org/10.1145/3637528.3671470

[5] Yuchen Fan, Kaiyan Zhang, Heng Zhou, Yuxin Zuo, Yanxu Chen, Yu Fu, Xinwei Long, Xuekai Zhu, Che Jiang, Yuchen Zhang, Li Kang, Gang Chen, Cheng Huang, Zhizhou He, Bingning Wang, Lei Bai, Ning Ding, and Bowen Zhou. 2025. SSRL: Self-Search Reinforcement Learning. arXiv:2508.10874 [cs.CL]

[6] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs]

[7] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics* 9 (April 2021), 346–361. https://doi.org/10.1162/tacl_a_00370

[8] Haoyu Han, Yaochen Xie, Hui Liu, Xianfeng Tang, Sreyashi Nag, William Headden, Yang Li, Chen Luo, Shuiwang Ji, Qi He, and Jiliang Tang. 2025. Reasoning with Graphs: Structuring Implicit Knowledge to Enhance LLMs Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 25698–25714. https://doi.org/10.18653/v1/2025.findings-acl.1319

[9] Mingyu Jin, Weidi Luo, Sitao Cheng, Xinyi Wang, Wenyue Hua, Ruixiang Tang, William Yang Wang, and Yongfeng Zhang. 2025. Disentangling Memory and Reasoning Ability in Large Language Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vienna, Austria, 1681–1701. https://doi.org/10.18653/v1/2025.acl-long.84

[10] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models Are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*, Vol. 35. 22199–22213.

[11] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, 9459–9474.

[12] Kuan Li, Liwen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Shuai Wang, and Minhao Cheng. 2025. LaRA: Benchmarking Retrieval-Augmented Generation and Long-Context LLMs – No Silver Bullet for LC or RAG Routing. In *Forty-Second International Conference on Machine Learning*.

[13] Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Association for Computational Linguistics, Miami, Florida, US, 881–893. https://doi.org/10.18653/v1/2024.emnlp-industry.66

[14] Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 3214–3252.

[15] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173. https://doi.org/10.1162/tacl_a_00638

[16] Jun-Yu Ma, Zhen-Hua Ling, Ningyu Zhang, and Jia-Chen Gu. 2024. Neighboring Perturbations of Knowledge Editing on Large Language Models. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 33839–33854.

[17] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, 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. In *Advances in Neural Information Processing Systems*, Vol. 36. 46534–46594.

[18] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and Editing Factual Associations in GPT. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, 17359–17372.

[19] Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2022. Mass-Editing Memory in a Transformer. In *The Eleventh International Conference on Learning Representations*.

[20] Elena Musi, Nadin Kökciyan, Khalid Al Khatib, Davide Ceolin, Emmanuelle Dietz, Klara Maximiliane Gutekunst, Annette Hautli-Janisz, Cristián Santibáñez, Jodi Schneider, Jonas Scholz, Cor Steging, Jacky Visser, and Henning Wachsmuth. 2025. Toward Reasonable Parrots: Why Large Language Models Should Argue with Us by Design. In *Proceedings of the 12th Argument mining Workshop*, Elena Chistova, Philipp Cimiano, Shohreh Haddadan, Gabriella Lapesa, and Ramon Ruiz-Dolz (Eds.). Association for Computational Linguistics, Vienna, Austria, 24–31. https://doi.org/10.18653/v1/2025.argmining-1.3

[21] Agada Joseph Oche, Ademola Glory Folashade, Tirthankar Ghosal, and Arpan Biswas. 2025. A Systematic Review of Key Retrieval-Augmented Generation (RAG) Systems: Progress, Gaps, and Future Directions. arXiv:2507.18910 [cs]

[22] Derek Powell, Walter Gerych, and Thomas Hartvigsen. 2024. TAXI: Evaluating Categorical Knowledge Editing for Language Models. In *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics, Bangkok, Thailand, 15343–15352. https://doi.org/10.18653/v1/2024.findings-acl.906

[23] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. [n.d.]. Reflexion: Language Agents with Verbal Reinforcement Learning. In *NeurIPS'23*, Vol. 36. 8634–8652.

[24] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4149–4158. https://doi.org/10.18653/v1/N19-1421

[25] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

[26] Yifei Wang, Yuheng Chen, Wanting Wen, Yu Sheng, Linjing Li, and Daniel Dajun Zeng. 2024. Unveiling Factual Recall Behaviors of Large Language Models through Knowledge Neurons. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Miami, Florida, USA, 7388–7402.

[27] Yiqun Wang, Chaoqun Wan, Sile Hu, Yonggang Zhang, Xiang Tian, Yaowu Chen, Xu Shen, and Jieping Ye. 2025. Tracing and Dissecting How LLMs Recall Factual Knowledge for Real World Questions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vienna, Austria, 23246–23271. https://doi.org/10.18653/v1/2025.acl-long.1133

[28] Taylor Webb, Shanka Subhra Mondal, and Ida Momennejad. 2025. A Brain-Inspired Agentic Architecture to Improve Planning with LLMs. *Nature Communications* 16, 1 (Sept. 2025), 8633. https://doi.org/10.1038/s41467-025-63804-5

[29] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, Vol. 35. 24824–24837.

[30] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Advances in Neural Information Processing Systems*, Vol. 36. Curran Associates, Inc., 11809–11822.

[31] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*.

[32] Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. [n.d.]. RankRAG: Unifying Context Ranking with Retrieval-Augmented Generation in LLMs. *NeurIPS'24* 37 ([n. d.]), 121156–121184.

[33] Ningyu Zhang, Bozhong Tian, Siyuan Cheng, Xiaozhuan Liang, Yi Hu, Kouying Xue, Yanjie Gou, Xi Chen, and Huajun Chen. 2024. InstructEdit: Instruction-Based Knowledge Editing for Large Language Models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI '24)*. Jeju, Korea, 6633–6641. https://doi.org/10.24963/ijcai.2024/733

[34] Bingxi Zhao, Lin Geng Foo, Ping Hu, Christian Theobalt, Hossein Rahmani, and Jun Liu. 2025. LLM-based Agentic Reasoning Frameworks: A Survey from Methods to Scenarios. arXiv:2508.17692 [cs]

[35] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024. Language Agent Tree Search Unifies Reasoning, Acting, and Planning in Language Models. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24, Vol. 235)*. JMLR.org, Vienna, Austria, 62138–62160.