Reasoning-Search-Augmented Large Language Models: A Comprehensive Survey

Biswas Poudel, Justin Liu, Sami Melhem, and **Xianshun Jiang**
Texas A&M University
College Station, TX
{bpoudel, j.liuu, samilmelhem, xianshun}@tamu.edu

Abstract

Large language models (LLMs) exhibit impressive emergent reasoning abilities, yet remain fundamentally constrained by their parametric memory and finite context windows. As a result, they hallucinate, struggle with up-to-date knowledge, and often misinterpret retrieved evidence. A recent line of work addresses these limitations by tightly coupling step-by-step reasoning with dynamic search, treating retrieval as an action within an agentic loop. This survey synthesises over forty systems that interleave planning, querying, retrieving, verifying, and stopping, rather than performing a single static retrieval. We organise the literature along training paradigms (prompt-based, supervised, reinforcement learning) and architectural choices (single-controller, modular, hierarchical, and tree-search). We further review benchmarks, metrics, and evaluation protocols, compare empirical performance, and highlight open challenges in reward design, long-horizon memory, efficiency, and safety. Our taxonomy and analysis aim to clarify the landscape of reasoning–search–augmented LLMs and outline promising directions for future work.Resources are maintained at https://github.com/notij/Awesome-Reasoning-Search-Augmented-LLMs

1 Introduction

Large language models (LLMs) have demonstrated remarkable emergent reasoning capabilities across mathematics, programming, and open-domain question answering. However, they are inherently limited by (i) static parametric knowledge, bounded by a training cut-off, and (ii) finite context windows. Their parameters are fixed at training time and their context windows are limited, which lead them to hallucinate or omit critical facts, when faced with complex, long-horizon, or recent questions queries. Chain-of-thought (CoT) prompting improves reasoning by instructing LLMs to produce intermediate steps, and self-consistency techniques sample multiple CoTs to select the most common answer (Besta et al., 2025).

Methods such as ReAct and Self-Ask interleave natural-language reasoning with calls to search APIs, using hand-crafted prompts and deterministic heuristics to decide when to search and what to retrieve (Yao et al., 2022). While effective in some settings, these approaches do not allow the model to *learn* when to search, what to retrieve, or how to integrate evidence. These methods depend on hand-crafted prompts fixed heuristics.

Parallel efforts in retrieval-augmented generation (RAG) constrain generation to retrieved documents, reducing hallucination relative to ungrounded LLMs, but RAG still suffers from retrieval irrelevance, residual hallucinations, and latency bottlenecks (Barnett et al., 2024). These limitations have sparked interest in agents that treat search as a trainable component of the reasoning process. The pioneering Search-R1 framework uses reinforcement learning (RL) teach an LLM to to autonomously generate multiple search queries during step-by-step reasoning and decide when to stop; it yields up to 41% higher exact match scores than static RAG baselines on open-domain and multi-hop QA benchmarks (Jin et al., 2025). Its successors refine this idea in several directions. ZeroSearch removes reliance on external APIs during training by simulating search with a smaller LLM, yet still finds that a learned retrieval module can match or surpass real-search performance (Sun et al., 2025). ReSearch trains solely via outcome-based RL, allowing the model to discover its own querying strategy (Chen et al., 2025), while StepSearch optimises a step-wise PPO objective with information-gain and redundancy-penalty rewards to improve multi-hop QA (Zheng et al., 2025a). Stratified GRPO addresses trajectory heterogeneity by stratifying advantages (Zhu et al., 2025). Later works like VerlTool modu-

larize the loop into planning, retrieving, verifying, and answering (Jiang et al., 2025a) or integrate hierarchical planning and tree search as in THOUGHTSCULPT (Chi et al., 2025)

1.1 Scope and Significance

These developments motivate our survey of reasoning-search-augmented LLMs—models that dynamically interleave reasoning (planning, decomposition, self-reflection) with active search. We formally define such systems as those implementing a loop of plan → query → retrieve → verify → stop:

- Plan: Generate or revise a task decomposition or search query based on current context.
- Query: Issue one or multiple search calls (one-shot, step-wise or batched) to an external environment (e.g., web, simulation, knowledge base).
- Retrieve: Fetch documents, snippets or structured data from the search environment; retrieval can be dense, sparse, hybrid or Monte-Carlo-tree–guided.
- Verify: Assess whether retrieved evidence supports the hypothesis; methods range from simple self-consistency checks, to learned verifiers, reward models or factuality scorers.
- Stop: Decide when to terminate the loop; stopping rules may be fixed (e.g., maximum turns), threshold-based, or learned from rewards.

By reasoning-search-augmented, we mean that the model's decision to search (or not) is conditional on the reasoning state, distinguishing these agents from vanilla RAG systems that perform a single retrieval before generation. Our focus is limited to text-based search agents that query web or local knowledge bases; we exclude code execution, multimodal tools, or purely structured search unless it is coupled with reasoning. Geographically, we cover work published between 2022 and 2025 in venues such as ACL, NeurIPS, ICML, AAAI and arXiv, spanning more than 40 distinct systems. Our survey is thus the first to:

- Provide background on reasoning topologies (CoT, ToT) and early tool-use frameworks (ReAct, Self-Ask), explaining their strengths and limitations.
- Define reasoning-search-augmented LLMs and formalise the plan → query → retrieve → verify → stop loop.
- Organise the literature into a two-axis taxonomy (training/architecture and loop design), which serves as the backbone of the survey.

We also discuss the open challenges in compute efficiency, reward design, long-horizon memory, structured retrieval, and safety.

2 Taxonomy of Reasoning–Search–Augmented LLMs

We organise the literature along two axes:

1. **Training / architectural paradigm**: RL-based vs. supervised vs. prompt-based; single-controller vs. modular vs. tree-search.
2. **Design of the plan → query → retrieve → verify → stop loop**: how queries are generated, what search environment is used (open web vs. local knowledge base), and whether verification or feedback modules are present.

Although we present distinct categories, their boundaries are fluid. For example, an agent that uses a search tool can be viewed simultaneously as a retrieval method, a reasoning method, and a tool-use method. Where appropriate, we highlight such *hybrid architectures* or cross-reference systems that serve multiple functional roles.

Table **??** (not shown due to space) summarises representative systems, indicating: (i) presence of a controller or planner module, (ii) search-query generation strategy (chain-of-thought tagging, learned policy, or rule-based), (iii) search environment (web vs. local corpora vs. knowledge graphs), (iv) verification / feedback mechanisms, and (v) training paradigm (prompt-only, supervised fine-tuning, RL).

2.1 Reinforcement Learning for Search-Augmented Reasoning

Single-controller RL agents. These methods train a *single* LLM policy that interleaves natural-language reasoning and search actions.

Jin et al. (2025) introduce Search-R1, which augments an LLM with search actions and token masking and trains it via RL. The model learns to decide when to issue search queries and how to incorporate

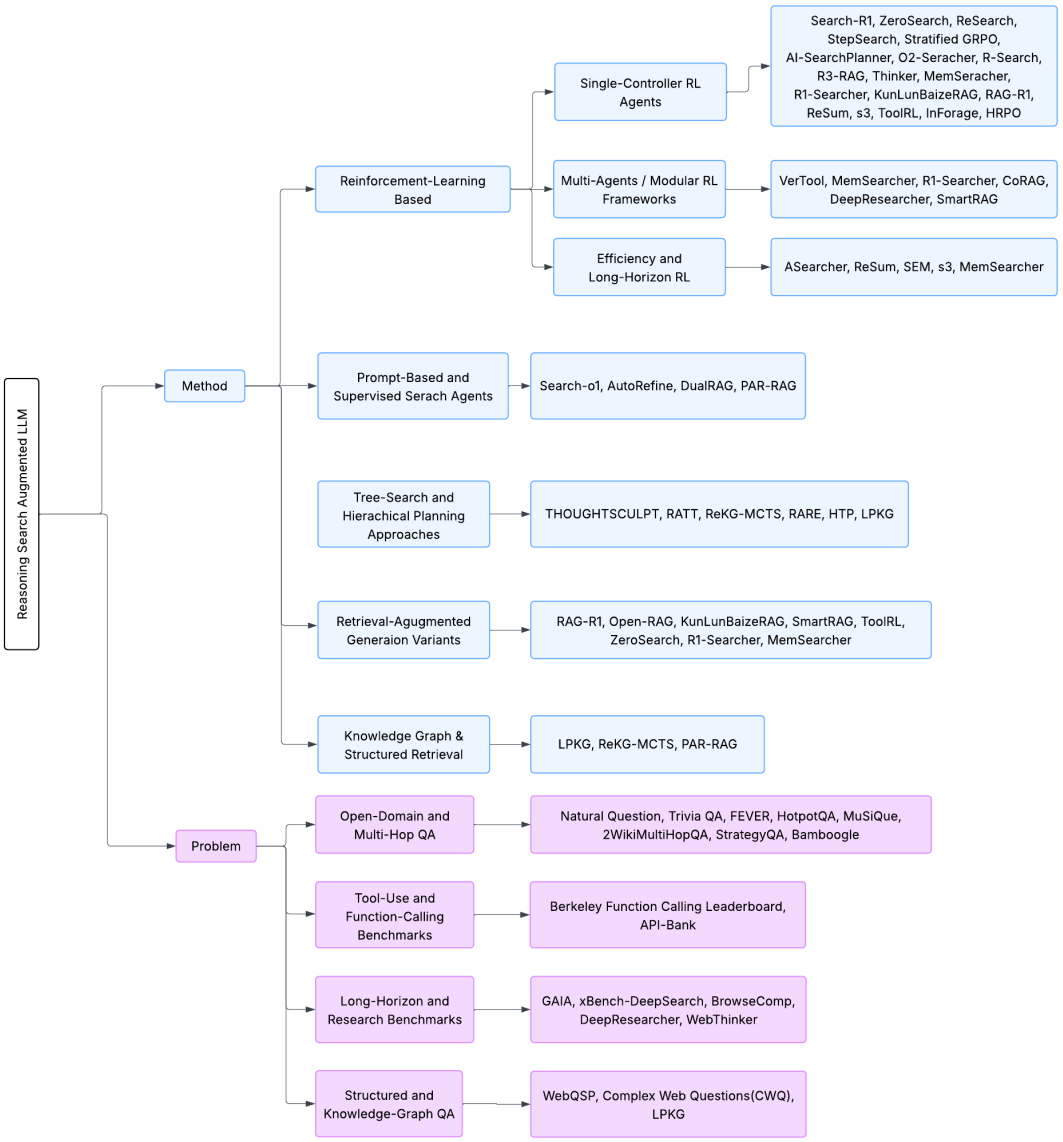

Figure 1: Taxonomy of Reasoning–Search–Augmented LLMs

retrieved snippets into its chain-of-thought, yielding large improvements over static RAG baselines. ZeroSearch avoids costly and noisy external APIs during training by simulating search with a smaller LLM, while still demonstrating that a trained retrieval module can match or exceed real-search performance at test time (Sun et al., 2025). ReSearch shows that RL alone—without supervised reasoning traces—is sufficient to learn when and how to search (Chen et al., 2025).

StepSearch optimises step-wise PPO with information-gain and redundancy penalties, encouraging the agent to issue informative, non-redundant queries and achieving sizeable gains on multi-hop QA (Zheng et al., 2025a). Stratified GRPO handles structural heterogeneity by computing stratified advantages across trajectories, improving stability and performance (Zhu et al., 2025). AI-SearchPlanner decouples planning and answer generation by training a small RL planner to choose search actions while keeping the QA model frozen (Mei et al., 2025b). O^2-Searcher trains an RL agent in a local simulated search environment for both open-ended and closed-ended questions, using unified rewards to outperform larger baselines with a 3B model (Mei et al., 2025a).

R-Search employs multi-reward RL, jointly optimising answer quality, evidence quality, and output format, and reports up to 32.2% improvement over strong baselines (Zhao et al., 2025). R3-RAG introduces a two-stage pipeline: a cold-start stage for supervised reasoning and retrieval, followed by RL

fine-tuning using answer and relevance rewards (Li et al., 2025c). Thinker combines hierarchical problem decomposition with a knowledge-boundary detector; RL tuning improves exact match (EM) compared to purely prompt-based reasoning (Xu et al., 2025). MemSearcher compresses the reasoning history into a compact memory and trains via multi-context GRPO, yielding double-digit relative gains (Yuan et al., 2025).

Further, R1-Searcher and ParallelSearch factor the agent into sub-components or batch multiple queries to trade off latency and accuracy (Song et al., 2025a). KunLunBaizeRAG combines reinforcement alignment, iterative search–think enhancement, and local routing to yield strong gains on knowledge-intensive tasks (Li et al., 2025a). RAG-R1 incentivises LLMs to transition from single-query to multi-query retrieval, reducing latency by 11.1% and improving accuracy by up to 13.7% (Tan et al., 2025). ReSum periodically summarises context during RL training, enabling long-horizon search with improved performance over ReAct-like baselines (Wu et al., 2025). SEM post-trains LLMs to decide when to search via GRPO, reducing redundant calls while maintaining accuracy (Sha et al., 2025).

The s3 framework decouples the searcher from the generator and trains only the searcher with a "Gain Beyond RAG" reward, achieving competitive results using only a few thousand training samples (Jiang et al., 2025b). ToolRL focuses on generic tool use, designing reward functions for tool selection and call correctness; RL training improves tool-use accuracy by a significant margin (Qian et al., 2025). Scent of Knowledge (InForage) uses rewards inspired by Information Foraging Theory, explicitly rewarding outcome quality and information gain (Qian and Liu, 2025). Hybrid Latent Reasoning leverages latent policy states and a gating mechanism with RL to improve performance on knowledge- and reasoning-intensive tasks (Yue et al., 2025).

Multi-agent and modular RL frameworks. Several approaches decompose the reasoning–search loop into specialised modules.

VerlTool provides a unified agent framework with separate planner, executor, and verifier modules, trained with RL and designed for multi-tool scenarios (Jiang et al., 2025a). MemSearcher combines a memory compressor with a search-and-reason agent and uses multi-context GRPO to jointly optimise memory management and retrieval (Yuan et al., 2025). R1-Searcher splits the system into planner, searcher, and verifier sub-agents and uses reward-trace distillation to improve coordination (Song et al., 2025a).

CoRAG (Chain-of-Retrieval Augmented Generation) iteratively refines queries and retrieves chains of documents, improving EM on KILT-style benchmarks by over ten points (Muhamed et al., 2025). DeepResearcher scales multi-agent research systems in real web environments via RL, training agents that iterate between search, summarisation, and drafting (Zheng et al., 2025b). SmartRAG jointly trains retriever and generator with RL, deciding when to retrieve and how to rewrite queries (Gao et al., 2025b). ASearcher (Beyond Ten Turns) focuses on fully asynchronous RL for long-horizon search, allowing more than one hundred search calls and achieving large gains on long-horizon benchmarks (Gao et al., 2025a).

Efficiency and long-horizon RL. A recurring theme in RL-based systems is the tension between exploration and efficiency. ReSum uses periodic summarisation to enable indefinite exploration while controlling context size (Wu et al., 2025); SEM and s3 explicitly penalise unnecessary searches or minimise training data (Sha et al., 2025; Jiang et al., 2025b). MemSearcher's compressed memory design likewise improves efficiency in long dialogues or research tasks (Yuan et al., 2025). These works illustrate that the *structure* of memory and search—not just the reward—is critical to scalability.

2.2 Prompt-Based and Supervised Search Agents

Prompt-based and supervised methods eschew RL in favour of carefully designed prompts or supervised fine-tuning.

Search-o1 uses a rule-based plan–query–retrieve–revise loop that requires no additional training, outperforming zero-shot baselines on QA tasks while remaining simple to implement (Li et al., 2025b). AutoRefine (Search-and-Refine During Think) introduces explicit knowledge-refinement steps between search calls and adds retrieval-specific objectives, improving multi-hop QA (Shi et al., 2025). DualRAG fine-tunes both a reasoning-augmented querying module and a progressive knowledge-aggregation module, which operate iteratively to

refine evidence sets (Cheng et al., 2025).

PAR-RAG (Plan-then-Act-and-Review RAG) decomposes questions into sub-questions, retrieves evidence at multiple granularities, and employs verification steps to correct reasoning errors; a top-down planning component constrains the reasoning path and improves robustness (Zhang et al., 2025b). More generally, Search–Refine frameworks insert static `Search:` and `Refine:` tags into chain-of-thought prompts, improving interpretability but remaining less adaptive than RL-based policies.

2.3 Tree-Search and Hierarchical Planning

Tree-search and hierarchical planning methods explore reasoning spaces more systematically.

THOUGHTSCULPT combines Monte-Carlo Tree Search (MCTS) with an LLM-powered thought generator and evaluator, allowing the agent to explore and prune candidate reasoning paths and achieve state-of-the-art gains on several reasoning and generation tasks (Chi et al., 2025). RATT constructs retrieval-augmented thought trees, integrating retrieval at each branch to maintain factual correctness and coherence (Zhang et al., 2025a).

ReKG-MCTS performs MCTS over knowledge graphs, guided by an LLM but without RL, and outperforms other training-free methods on structured QA benchmarks (**?**). RARE extends rStar MCTS by adding retrieval-augmented actions and a factuality scorer to evaluate reasoning paths (Tran et al., 2025). HyperTree Planning proposes hypertree-structured planning outlines that support hierarchical divide-and-conquer reasoning and demonstrates substantial gains on a travel-planning benchmark (Gui et al., 2025). LPKG fine-tunes LLMs using planning traces mined from knowledge graphs, enabling explicit plan generation and structured retrieval for complex questions (Wang et al., 2024).

2.4 Retrieval-Augmented Generation Variants

Several systems redesign the RAG pipeline to integrate reasoning and retrieval more tightly, while remaining closer to the RAG paradigm than to fully agentic RL.

RAG-R1 transitions from single-query to multi-query retrieval through RL, improving both accuracy and latency (Tan et al., 2025). Open-RAG converts a dense model into a sparse mixture-of-experts with hybrid adaptive retrieval, showing that an open-source 7B model can surpass larger closed-source baselines on knowledge-intensive tasks (Islam et al., 2024). KunLunBaizeRAG combines reasoning alignment, search–think iterative enhancement, and network-local routing to boost EM and LLM-judged scores (Li et al., 2025a). EviNote-RAG inserts an intermediate note-taking stage and uses an entailment-based evidence-quality reward to improve F1 on multi-hop QA benchmarks (Dai et al., 2025).

SmartRAG and ToolRL jointly learn retrieval, reasoning, and tool use, demonstrating generalisation to unseen tools and APIs (Gao et al., 2025b; Qian et al., 2025). ZeroSearch, ParallelSearch, and MemSearcher sit at the boundary between classic RAG and RL agents: they leverage simulation, batched retrieval, and memory compression to improve efficiency while maintaining agentic behaviour (Sun et al., 2025; Yuan et al., 2025; Song et al., 2025a).

2.5 Knowledge Graph and Structured Retrieval

Structured retrieval settings often involve explicit knowledge graphs, APIs, or function calls. LPKG leverages planning traces derived from knowledge graphs, training models to decompose complex questions into structured sub-queries (Wang et al., 2024). ReKG-MCTS and RARE augment MCTS with knowledge-graph traversal and retrieval actions (**?**Tran et al., 2025). PAR-RAG integrates vector-based retrieval, knowledge graphs, and citation-based evidence selection within a unified planning and verification framework (Zhang et al., 2025b).

These methods highlight that reasoning–search agents must ultimately cope with heterogeneous sources, including structured databases, APIs, and multimodal content.

3 Benchmarks, Datasets, and Metrics

Evaluating reassoning–search-augmented models requires benchmarks that test both information retrieval and reasoning, along with metrics for answer quality, search behavior, and tool-use. Here, we summarize key datasets and evaluation protocols.

3.1 Open-Domain QA Datasets

Natural Questions (NQ): This is an open-domain dataset with hundreds of thousands of anonymized Google Queries. Models must extract precise answers from full Wikipedia articles, making this a strong retrieval and comprehension benchmark. (Kwiatkowski et al., 2019)

TriviaQA: A collection of nearly 100k question-answer pairs, each with multiple evidence documents and requiring retrieval from unstructured sources.(Joshi et al., 2017)

3.2 Multi-Hop QA Datasets

HotpotQA: About 113k multi-hop questions with sentence-level supporting facts, directly designed to test multi-step reasoning. (Yang et al., 2018)
FEVER: Fact verification tasks with 195k human-written claims, where systems must verify or refute claims, citing supporting evidence (Thorne et al., 2018).
MuSiQue: Multi-hop questions composed to avoid exploitable shortcuts, ensuring truly multi-step reasoning.(Trivedi et al., 2022)
2WikiMultiHopQA: A multi-hop dataset requiring at least two reasoning hops per question, which explicit reasoning chains provided. (Ho et al., 2020)
StrategyQA: Short, seemingly simple questions whose reasoning steps are implicit and provides explicit decompositions for evaluation.(Geva et al., 2021)
Bamboogle: A small multi-hop dataset that curates questions that even search engines fail to answer. They are useful for testing compositional robustness. (Press et al., 2023)

3.3 Tool-Use and Function-Calling Benchmarks

API-Bank: A large-scale benchmark that introduces dozens of APIs and annotated tool-use dialogues across hundreds of domains. It can measure whether a model can plan, select, and call sequential tool execution.(Li et al., 2023)
Berekeley Function Calling Leaderboard (BFCL): Built on API-BAnk, the Berekeley Function Calling Leaderboard evaluates models on thousands of real-world functions, including serial and parallel calls, with an abstract syntax tree (AST) metric to verify parameter correctness.(Patil et al., 2025)

3.4 Long-Horizon and Research Benchmarks

GAIA: This benchmark contains hundreds of multi-step questions, each requiring multiple tool calls and long-term planning.(Mialon et al., 2024)
xBench-DeepSearch and BrowseComp: Long-horizon research questions that require dozens of queries and extensive summarization. (Wei et al., 2025)
ReSum: A summarization benchmark; 30B model trained on only 1k samples attains competitive performance on long-horizon search tasks by periodically summarizing context (Wu et al., 2025) be
DeepResearcher/ Related Environments: These benchmarks embed agents in real browser environments, combining retrieval, summarization, and drafting. Human evaluators judge the final reports for quality and novelty. (Zheng et al., 2025b; Gao et al., 2025a).

3.5 Structured and Knowledge-Graph QA

WebQSP: Structured QA benchmark that requires traversing knowledge graphs such as Freebase, converting questions into graph-structured queries.(Yih et al., 2016)
Complex Web Questions (CWQ): Extension of WebQSP requiring multi-hop reasoning over Freebase.(Talmor and Berant, 2018)
ReGK-MCTS: Applies Monte Carlo Tree Search (MCTS) over knowledge graphs, guided by LLMs, and outperforms training-free baselines on these datasets (Song et al., 2025b).
LPKG: Constructs training data by extracting patterns from knowledge graphs and verbalizing them into question-plan-answer triples. It also introduces the CLQA-Wiki benchmark to test plan generation and execution (Wang et al., 2024).

3.6 Metrics and Evaluation Protocols

Most studies report standard QA metrics such as exact match (EM) and token-level F1 on final answers. However, these metrics ignore search behaviour and tool use. Consequently, researchers augment them with:

- **Supporting-fact F1**, measuring overlap with annotated evidence sentences (e.g., HotpotQA, FEVER).
- **Average number of search steps or tool calls**, used to compare one-shot and step-wise retrieval policies (e.g., StepSearch, SEM).
- **Retrieval latency and throughput**, crucial for systems that batch or parallelise queries (e.g., ParallelSearch, ASearcher).
- **Reward curves and returns**, especially in RL work, where outcome-based and process-based rewards (for evidence quality, information gain, or efficiency) are tracked over training (Jin et al., 2025; Zhao et al., 2025; Qian and Liu, 2025).

- **Tool-call success rates and AST accuracy**, essential for API-Bank and BFCL style evaluations.
- **Pass@k or Avg@k** scores for long-horizon tasks (e.g., GAIA, xBench), which allow partial credit over multiple attempts.
- **Human evaluation** of report quality, novelty, or safety, particularly for research and planning tasks (Zheng et al., 2025b; Gao et al., 2025a).

These metrics highlight that evaluating reasoning–search agents requires assessing not only correctness but also *efficiency*, *evidence quality*, and *tool-use reliability*. We recommend that future work report both QA metrics and search/tool-specific measures, and clearly specify the evaluation setting (number of search turns, allowed tools, maximum context length).

4 Comparative Performance and Trends

Over the past three years, the field has transitioned from static RAG to systems that treat search as an explicit action within a broader reasoning loop. Early RL agents such as Search-R1 demonstrate that even simple outcome-based rewards can substantially improve EM on open-domain and multi-hop QA (Jin et al., 2025). Subsequent systems refine this recipe: StepSearch uses step-wise information-gain rewards and reports 4–11 point EM improvements (Zheng et al., 2025a); Stratified GRPO normalises advantages within trajectory strata to stabilise RL and adds several points of accuracy (Zhu et al., 2025); ZeroSearch shows that search can be efficiently simulated during training without sacrificing test-time performance (Sun et al., 2025).

Two broad trends emerge:

From one-shot to iterative search. RL agents increasingly move from single-shot retrieval to iterative, multi-step search. ParallelSearch and StepSearch batch or sequence queries, reducing latency while improving accuracy (Zheng et al., 2025a; Song et al., 2025a). ReSum pushes this further by periodically summarising retrieved context, enabling dozens of search calls in long-horizon settings without running out of context (Wu et al., 2025). MemSearcher and SEM likewise show that carefully managing memory and search frequency can yield efficient yet strong agents (Yuan et al., 2025; Sha et al., 2025).

Modularity and hierarchy. Systems are becoming more modular and hierarchical. R1-Searcher decouples planning, search, and verification and trains them with reward-trace distillation (Song et al., 2025a). MemSearcher compresses history into a dedicated memory module and still outperforms larger baselines (Yuan et al., 2025). Thinker decomposes problems into sub-questions and uses RL to decide when external search is required (Xu et al., 2025). CoRAG adopts a collaborative scheme in which multiple agents iteratively refine queries and evidence sets (Muhamed et al., 2025). These designs improve interpretability and robustness, but at the cost of increased training complexity and more intricate engineering.

Beyond RL, planning and tree-search methods (THOUGHTSCULPT, RATT, ReKG-MCTS, RARE, HyperTree Planning) explore reasoning spaces through explicit search over thoughts or knowledge-graph paths, avoiding RL but relying on strong search heuristics and domain-specific templates (Chi et al., 2025; Zhang et al., 2025a; **?**; Tran et al., 2025; Gui et al., 2025). RAG variants such as RAG-R1, Open-RAG, KunLunBaizeRAG, and EviNote-RAG complement these agentic systems by rethinking retrieval strategies and evidence aggregation (Tan et al., 2025; Islam et al., 2024; Li et al., 2025a; Dai et al., 2025).

4.1 Critical Analysis

This survey reveals several key insights:

Longer is not always better. Studies on efficient reasoning show that overly long chains-of-thought can hurt performance; agents that adaptively determine when to search and stop often outperform those that merely increase step count. Explicit stopping criteria or penalties for redundant search (e.g., SEM, s3, Scent of Knowledge) exemplify this trade-off (Sha et al., 2025; Jiang et al., 2025b; Qian and Liu, 2025).

Balancing exploration and exploitation. RL agents must explore diverse search paths without wasting queries. Step-wise rewards, information-gain objectives, and multi-reward strategies—as in StepSearch, R-Search, R3-RAG, and Scent of Knowledge—seek this balance (Zheng et al., 2025a; Zhao et al., 2025; Li et al., 2025c; Qian and Liu, 2025). Yet, designing incentives that foster

both thoroughness and efficiency remains a challenge.

Simulation vs. real-world search. Simulated search (ZeroSearch, Scent of Knowledge) reduces costs and avoids noisy APIs, but may not capture the complexity and adversarial nature of real web search (Sun et al., 2025; Qian and Liu, 2025). Hybrid approaches, combining simulated and real retrieval (e.g., KunLunBaizeRAG, DeepResearcher), appear promising (Li et al., 2025a; Zheng et al., 2025b).

Planning and RL as complementary. Planning methods offer explicit decompositions and structure (THOUGHTSCULPT, HyperTree Planning), while RL learns adaptive policies over these frameworks. Integrating planners to propose search paths and RL to refine when to explore, backtrack, or stop is a natural next step (Chi et al., 2025; Gui et al., 2025).

5 Challenges and Future Directions

We identified the following open problems and promising future directions.

Compute and data efficiency. RL agents often require millions of tokens and repeated search interactions, leading to high training costs even with simulated retrieval or stratified advantages (Zhu et al., 2025; Sun et al., 2025; Wu et al., 2025; Gao et al., 2025a). Future research should pursue more sample-efficient RL (e.g., off-policy, model-based, meta-learning), improved synthetic data, and transferable policies across tasks and tools.

Reward design and evaluation. Designing rewards that align with human intent is challenging: optimizing for correctness can cause hallucinations or redundant search, while efficiency may truncate reasoning. Information-theoretic objectives and human feedback offer alternatives (Zhao et al., 2025; Qian and Liu, 2025). Metrics should move beyond exact match to also capture evidence quality, efficiency, and tool-use robustness; tool-call and AST-based measures, as in BFCL-style benchmarks, are key steps.

Long-horizon reasoning and memory. Current summarisation and memory compression methods (e.g., ReSum, MemSearcher) are heuristic and risk losing important details (Wu et al., 2025; Yuan et al., 2025). Learned memories, hierarchical attention, and retrieval-based episodic recall may better support long-term reasoning. Jointly learning memory and search policies remains an underexplored opportunity.

Structured and multimodal retrieval. Most work centers on unstructured text, yet practical tasks span code, tables, images, and APIs. Early efforts in knowledge-graph reasoning and function calling (e.g., ReKG-MCTS, ToolRL) highlight the need for agents to integrate diverse modalities and tools within unified reasoning loops (**?**Wang et al., 2024; Qian et al., 2025; Gao et al., 2025b).

Alignment, safety, and robustness. Agents may retrieve biased, misleading, or harmful content. Verification and consensus modules (e.g., R1-Searcher, CoRAG) offer partial safeguards (Song et al., 2025a; Muhamed et al., 2025), but fully aligning agent behavior with human values is unsolved. Progress is needed in adversarial scenarios, robust calibration, and user-in-the-loop verification.

Human–AI collaboration. Ultimately, reasoning–search agents should support humans in complex tasks by soliciting clarifications, exposing their reasoning, and adapting to user feedback. Multi-agent research systems and long-horizon planning frameworks (e.g., DeepResearcher) suggest that conversational planning and feedback are vital for usefulness and trust (Zheng et al., 2025b; Gao et al., 2025a).

6 Conclusion

Static LLMs are limited by parametric memory and context windows, while standard RAG approaches lack deep reasoning over retrieved content. Reasoning–search–augmented LLMs overcome these barriers by integrating search as a core action within an agentic planning and verification loop.

The systems surveyed here span RL-based agents, modular frameworks, planning and tree-search strategies and advanced RAG variants demonstrate that searching *with* reasoning yields substantial gains on knowledge-intensive, long-horizon tasks, while highlighting open challenges in efficiency, reward desing, memory, structured retrieval, and human collaboration.

We hope that this survey and taxonomy will help guide future research and development of more reliable, efficient, and trustworthy reasoning-search-augmented LLMs.

References

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven failure points when engineering a retrieval augmented generation system. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, pages 194–199.

Maciej Besta, Florim Memedi, Zhenyu Zhang, Robert Gerstenberger, Guangyuan Piao, Nils Blach, Piotr Nyczyk, Marcin Copik, Grzegorz Kwaśniewski, Jurgen Müller, et al. 2025. Demystifying chains, trees, and graphs of thoughts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. Research: Learning to reason with search for llms via reinforcement learning.

Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazhen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. 2025. DualRAG: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31877–31899, Vienna, Austria. Association for Computational Linguistics.

Yizhou Chi, Kevin Yang, and Dan Klein. 2025. ThoughtSculpt: Reasoning with intermediate revision and search. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7685–7711, Albuquerque, New Mexico. Association for Computational Linguistics.

Yuqin Dai, Guoqing Wang, Yuan Wang, Kairan Dou, Kaichen Zhou, Zhanwei Zhang, Shuo Yang, Fei Tang, Jun Yin, Pengyu Zeng, Zhenzhe Ying, Can Yi, Changhua Meng, Yuchen Zhou, Yongliang Shen, and Shuai Lu. 2025. Evinote-rag: Enhancing rag models via answer-supportive evidence notes.

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

Jingsheng Gao, Linxu Li, Ke Ji, Weiyuan Li, Yixin Lian, yuzhuo fu, and Bin Dai. 2025b. SmartRAG: Jointly learn RAG-related tasks from the environment feedback. In *The Thirteenth International Conference on Learning Representations*.

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*, 9:346–361.

Runquan Gui, Zhihai Wang, Jie Wang, Chi Ma, Huiling Zhen, Mingxuan Yuan, Jianye HAO, Defu Lian, Enhong Chen, and Feng Wu. 2025. Hypertree planning: Enhancing LLM reasoning via hierarchical thinking. In *Forty-second International Conference on Machine Learning*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shayekh Bin Islam, Md Asib Rahman, K S M Tozammel Hossain, Enamul Hoque, Shafiq Joty, and Md Rizwan Parvez. 2024. Open-RAG: Enhanced retrieval augmented reasoning with open-source large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14231–14244, Miami, Florida, USA. Association for Computational Linguistics.

Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, Tianyu Pang, and Wenhu Chen. 2025a. Verltool: Towards holistic agentic reinforcement learning with tool use.

Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025b. s3: You don't need that much data to train a search agent via RL. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21610–21628, Suzhou, China. Association for Computational Linguistics.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Cheng Li, Jiexiong Liu, Yixuan Chen, Qihang Zhou, and KunLun Meta. 2025a. Kunlunbaizerag: Reinforcement learning driven inference performance leap for large language models.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. API-bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3102–3116, Singapore. Association for Computational Linguistics.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025b. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5420–5438, Suzhou, China. Association for Computational Linguistics.

Yuan Li, Qi Luo, Xiaonan Li, Bufan Li, Qinyuan Cheng, Bo Wang, Yining Zheng, Yuxin Wang, Zhangyue Yin, and Xipeng Qiu. 2025c. R3-rag: Learning step-by-step reasoning and retrieval for llms via reinforcement learning.

Jianbiao Mei, Tao Hu, Daocheng Fu, Licheng Wen, Xuemeng Yang, Rong Wu, Pinlong Cai, Xinyu Cai, Xing Gao, Yu Yang, Chengjun Xie, Botian Shi, Yong Liu, and Yu Qiao. 2025a. O^2-searcher: A searching-based agent model for open-domain open-ended question answering.

Lang Mei, Zhihan Yang, and Chong Chen. 2025b. Aisearchplanner: Modular agentic search via pareto-optimal multi-objective reinforcement learning.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*.

Aashiq Muhamed, Mona T. Diab, and Virginia Smith. 2025. CoRAG: Collaborative retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 265–276, Albuquerque, New Mexico. Association for Computational Linguistics.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru WANG, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. ToolRL: Reward is all tool learning needs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Hongjin Qian and Zheng Liu. 2025. Scent of knowledge: Optimizing search-enhanced reasoning with information foraging. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zeyang Sha, Shiwen Cui, and Weiqiang Wang. 2025. Sem: Reinforcement learning for search-efficient large language models.

Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. 2025. Search and refine during think: Facilitating knowledge refinement for improved retrieval-augmented reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. 2025a. R1-searcher: Incentivizing the search capability in llms via reinforcement learning.

Xiaozhuang Song, Shufei Zhang, and Tianshu Yu. 2025b. ReKG-MCTS: Reinforcing LLM reasoning on knowledge graphs via training-free Monte Carlo tree search. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 9288–9306, Vienna, Austria. Association for Computational Linguistics.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Fei Huang, and Yan Zhang. 2025. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Zhiwen Tan, Jiaming Huang, Qintong Wu, Hongxuan Zhang, Chenyi Zhuang, and Jinjie Gu. 2025. Rag-r1: Incentivizing the search and reasoning capabilities of llms through multi-query parallelism.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Hieu Tran, Zonghai Yao, Zhichao Yang, Junda Wang, Yifan Zhang, Shuo Han, Feiyun Ouyang, and Hong Yu. 2025. RARE: Retrieval-augmented reasoning enhancement for large language models. In *Proceedings of the 63rd Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 18305–18330, Vienna, Austria. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Junjie Wang, Mingyang Chen, Binbin Hu, Dan Yang, Ziqi Liu, Yue Shen, Peng Wei, Zhiqiang Zhang, Jinjie Gu, Jun Zhou, Jeff Z. Pan, Wen Zhang, and Huajun Chen. 2024. Learning to plan for retrieval-augmented large language models from knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7813–7835, Miami, Florida, USA. Association for Computational Linguistics.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Minhao Cheng, Shuai Wang, Hong Cheng, and Jingren Zhou. 2025. Resum: Unlocking long-horizon search intelligence via context summarization.

Jun Xu, Xinkai Du, Yu Ao, Peilong Zhao, Yang Li, Ling Zhong, Lin Yuan, Zhongpu Bo, Xiaorui Wang, Mengshu Sun, Zhengke Gui, Dalong Zhang, Zhaoyang Wang, Qiwei Wang, Yangyang Hou, Zhiying Yin, Haofen Wang, Huajun Chen, Lei Liang, and Jun Zhou. 2025. Thinker: Training llms in hierarchical thinking for deep search via multi-turn interaction.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

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*.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

Qianhao Yuan, Jie Lou, Zichao Li, Jiawei Chen, Yaojie Lu, Hongyu Lin, Le Sun, Debing Zhang, and Xianpei Han. 2025. Memsearcher: Training llms to reason, search and manage memory via end-to-end reinforcement learning. *arXiv preprint arXiv:2511.02805*.

Zhenrui Yue, Bowen Jin, Huimin Zeng, Honglei Zhuang, Zhen Qin, Jinsung Yoon, Lanyu Shang, Jiawei Han, and Dong Wang. 2025. Hybrid latent reasoning via reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Jinghan Zhang, Xiting Wang, Weijieying Ren, Lu Jiang, Dongjie Wang, and Kunpeng Liu. 2025a. Ratt: a thought structure for coherent and correct llm reasoning. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

Ningning Zhang, Chi Zhang, Zhizhong Tan, Xingxing Yang, Weiping Deng, and Wenyong Wang. 2025b. Credible plan-driven rag method for multi-hop question answering.

Qingfei Zhao, Ruobing Wang, Dingling Xu, Daren Zha, and Limin Liu. 2025. R-search: Empowering llm reasoning with search via multi-reward reinforcement learning.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. 2025a. StepSearch: Igniting LLMs search ability via step-wise proximal policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21816–21841, Suzhou, China. Association for Computational Linguistics.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025b. DeepResearcher: Scaling deep research via reinforcement learning in real-world environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 414–431, Suzhou, China. Association for Computational Linguistics.

Mingkang Zhu, Xi Chen, Bei Yu, Hengshuang Zhao, and Jiaya Jia. 2025. Stratified grpo: Handling structural heterogeneity in reinforcement learning of llm search agents.

A Appendix

Individual Contributions

Biswas Poudel

- Spearheaded the conceptualization of the survey, actively forming the team and defining research scope regarding reasoning-search-augmented LLMs.

- Served as the primary point of contact for the group, coordinating directly with the professor and TA to incorporate feedback and refine the project focus (with thanks to Dr. Zhang for helping redefine the project scope).
- Oversaw the project's operational aspects, including scheduling meetings, managing the timeline, breaking down and defining smaller tasks, and delegating them to ensure efficient collaboration.
- Led and directed the literature collection and review of over 40 publications from venues such as NeurIPS, ACL, ACM, IEEE, ICML, and arXiv, rigorously filtering the literature to focus on agents that treat retrieval as a dynamic action within an agentic loop, thus distinguishing them from static RAG systems.
- Architected the survey's two-axis taxonomy, classifying models by training paradigm (RL, supervised, prompt-based) and architectural design (single-controller, modular, tree-search), which served as the backbone for organizing the report.
- Authored "Introduction, Motivation & Scope", and "Taxonomy" sections in final report.
- Collaborated on refining the slides and co-presented the project alongside Justin.

Sami Melhem

- Led a comprehensive literature review by systematically searching, collecting, and organizing all relevant papers from arXiv and major ML/NLP venues, ensuring the survey reflected the full landscape of methods that enhance LLMs with explicit reasoning, tool use, or structured intermediate steps.
- Closely read and analyzed each paper, then distilled the technical content into clear, structured summaries that captured core ideas, architectures, training strategies, evaluation setups, and limitations; synthesized these into comparative writeups highlighting common patterns, key design trade-offs, and gaps in the existing reasoning-augmented LLM literature.
- Built and maintained an internal, tagged catalog of works (e.g., by reasoning paradigm, supervision type, task domain, parser or tool interface, evaluation metric), helping the team quickly navigate the space of papers, align terminology, and identify under-explored areas and open problems to emphasize in the final manuscript.
- Designed and authored the taxonomy diagram that links all parsers discussed in the survey into a unified conceptual framework, organizing them into coherent categories and interaction layers, and iteratively refining the visualization based on co-author feedback to clearly communicate each parser's role, dependencies, and place within end-to-end reasoning-augmented LLM pipelines.

Xianshun Jiang

- Explored benchmarking comparisons for different frameworks, but determined that exhaustive benchmark comparison was infeasible and not necessary for a survey paper (informed by https://arxiv.org/pdf/2502.17504).
- Drafted the GitHub repository, organizing all frameworks and collecting URLs for papers, GitHub, datasets, and models, following the format of https://github.com/yuzhimanhua/Awesome-Scientific-Language-Models.
- Drafted parts of the report, contributed to the appendix by sorting models, base models, datasets, and keywords, and created a taxonomy diagram for the paper using draw.io.
- Assisted in drafting the presentation slides.

Justin Liu

- Contributed to the Datasets/Benchmarks/Metrics section, helping organize the content and adding new material based on research papers gathered. Organized the sections following the structure of the survey paper at https://arxiv.org/pdf/2502.17504.
- Attempted to gather data from research paper experiments to build a benchmark suite for the survey; however, the large number of frameworks made unification infeasible.
- Helped refine and tune the slides for the final presentation.

Model	Year	Base Model	Dataset	Keywords
Search-R1 (Jin et al., 2025)	2025	llama3, Qwen2.5	Natural Questions, HotpotQA	Scalable RL Training Framework; Search Engine Calling Interleaved LLM
ZeroSearch (Sun et al., 2025)	2025	llama3, Qwen2.5	Natural Questions, HotpotQA	Supervised Fine-tuning
ReSearch (Chen et al., 2025)	2025	Qwen2.5	FlashRAG (38 datasets)	Tool Call for LLMs
StepSearch (Zheng et al., 2025a)	2025	Qwen2.5	MuSiQue	Step-Wise Proximal Policy Optimization
Stratified GRPO (Zhu et al., 2025)	2025	Qwen2.5	Natural Questions, HotpotQA	Structural Heterogeneity
AI-SearchPlanner (Mei et al., 2025b)	2025	Qwen2.5, Qwen3 Deepseek-V3 Deepseek-R1	Natural Questions, HotpotQA	Search Planning
O2-Searcher (Mei et al., 2025a)	2025	Qwen2.5	Natural Questions, HotpotQA	Searching-based Agent; Open-Domain QA
R-Searcher (Zhao et al., 2025)	2025	Qwen2.5	2WikiMultiHopQA	Reasoning–Search Integration
R3-RAG (Li et al., 2025c)	2025	llama3.1, Qwen2.5	Synthesized Trajectories	Learn Optimal Reasoning–Retrieval Strategies
Thinker (Xu et al., 2025)	2025	Qwen2.5	Natural Questions HotpotQA, WebQA	Deep Thinking and Reasoning
MemSearcher (Yuan et al., 2025)	2025	Qwen2.5	Natural Questions, HotpotQA	Multi-context GRPO
R1-Searcher (Song et al., 2025a)	2025	llama3.1, Qwen2.5	HotpotQA, 2WikiMultiHopQA	Two-stage Outcome-based RL
KunLunBaizeRAG (Li et al., 2025a)	2025	Baize	FlashRAG (38 datasets)	DAPO; Search–Think Iterative Enhancement
RAG-R1 (Tan et al., 2025)	2025	Qwen2.5	HotpotQA, 2WikiMultiHopQA	Multi-query Parallelism
ReSum (Wu et al., 2025)	2025	Websailor	SailorFog-QA	Multi-Agent RL; Agentic RAG; Search Agent
SEM (Sha et al., 2025)	2025	Qwen	MuSiQue, MMLU	Post-train
s3 (Jiang et al., 2025b)	2025	Qwen2.5 Claude-3-Haiku	Natural Questions, HotpotQA	Gain Beyond RAG Reward
ToolRL (Qian et al., 2025)	2025	llama3.2, Qwen2.5	ToolACE, Hammer, xLAM	Tool Selection; Application Tasks
InForage (Qian and Liu, 2025)	2025	Qwen2.5	WebQA (self-crawled)	Human-guided Dataset; Real-world Web Tasks
HRPO (Yue et al., 2025)	2025	Qwen2.5	MMLU, GSM8K, MATH	Hybrid Latent Reasoning
VerlTool (Jiang et al., 2025a)	2025	Qwen2.5	-	Diverse Tool Use
CoRAG (Muhamed et al., 2025)	2025	llama3.1	2WikiMultiHopQA, MuSiQue HotpotQA, bamboogol	Dynamic Query Reformulation
DeepResearcher (Zheng et al., 2025b)	2025	Qwen2.5	NQ, HotpotQA TriviaQA, 2WikiMultiHopQA	Cognitive Behaviors
SmartRAG (Gao et al., 2025b)	2025	llama2 Flan-T5 Large	PopQA, AmbigNQ HotpotQA, TriviaQA OpenBookQA, MedQA-en ARC-c	Joint Learning of RAG Tasks
ASearcher (Gao et al., 2025a)	2025	Qwen2.5	HotpotQA, 2WikiMultiHopQA	Asynchronous RL; Prompt-based Agent
Search-o1 (Li et al., 2025b)	2025	Qwen2.5, QwQ llama3.3, GPT-4o Deepseek-R1	GPQA, MATH500 NQ, HotpotQA	Agentic Search-Enhanced Reasoning
AutoRefine (Shi et al., 2025)	2025	Qwen2.5	FlashRAG (38 datasets)	Knowledge Refinement
DualRAG (Cheng et al., 2025)	2025	Qwen2.5	HotpotQA, MuSiQue 2WikiMultiHopQA	Dual-process Reasoning–Retrieval
PAR-RAG (Zhang et al., 2025b)	2025	GPT-4o	MuSiQue, TriviaQA	Retrieval-Augmented Multi-hop QA

Table 1: Methods for Reasoning-Search-Augmented Large Language Models