

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 FUTUREMIND: EQUIPPING SMALL LANGUAGE MODELS WITH STRATEGIC THINKING-PATTERN PRIORS VIA ADAPTIVE KNOWLEDGE DISTILLATION

Anonymous authors

Paper under double-blind review

ABSTRACT

Small Language Models (SLMs) are attractive for cost-sensitive and resource-limited settings due to their efficient, low-latency inference. However, they often struggle with complex, knowledge-intensive tasks that require structured reasoning and effective retrieval. To address these limitations, we propose FutureMind, a modular reasoning framework that equips SLMs with strategic thinking-pattern priors via adaptive knowledge distillation from large language models (LLMs). FutureMind introduces a dynamic reasoning pipeline composed of four key modules: Problem Analysis, Logical Reasoning, Strategy Planning, and Retrieval Guidance. This pipeline is augmented by three distinct retrieval paradigms that decompose complex queries into tractable subproblems, ensuring efficient and accurate retrieval execution. Extensive experiments on multi-hop QA benchmarks, including 2WikiMultihopQA, MuSiQue, Bamboogle, and Frames, demonstrate the superiority of FutureMind. It consistently outperforms strong baselines such as Search-o1, achieving state-of-the-art results under free training conditions across diverse SLM architectures and scales. Beyond empirical gains, our analysis reveals that the process of thinking-pattern distillation is restricted by the cognitive bias bottleneck between the teacher (LLMs) and student (SLMs) models. This provides new perspectives on the transferability of reasoning skills, paving the way for the development of SLMs that combine efficiency with genuine cognitive capability.

1 INTRODUCTION

In recent years, driven by massive datasets and scalable computing, Large Language Models (LLMs) have achieved outstanding problem-understanding and problem-solving performance on a wide range of general tasks such as commonsense inference (Yang et al., 2025), code generation (Guo et al., 2024), and mathematical reasoning (Shao et al., 2024) through pre-training (Raffel et al., 2020), instruction tuning (Wei et al., 2022a), reinforcement learning from human feedback (RLHF) (Touvron et al., 2023; OpenAI, 2023). However, once problems become time-sensitive or require domain-specific knowledge (Peng et al., 2023; Li et al., 2023b), model performance is constrained by their inherent, static parameters, exposing shortcomings like stale knowledge and insufficient domain coverage. This limitation highlights the necessity of introducing external knowledge sources during reasoning. Against this backdrop, Retrieval-Augmented Generation (RAG) (Gao et al., 2023; Xiong et al., 2025) has emerged: by supplying the model with retrieved documents before inference, it effectively enhances both the accuracy and domain adaptability of language models. Yet, single-step retrieval often struggles with knowledge-intensive, multi-hop reasoning tasks (Yang et al., 2018; Ho et al., 2020b). In response, recent studies have proposed "deep search" paradigms (Li et al., 2025c; Alzubi et al., 2025) that emphasize dynamic interaction between reasoning and retrieval: during problem solving, the model continuously decomposes the question, iteratively retrieves information, and aggregates evidence until the answer converges.

As problem complexity grows, increasing model size or memory alone is insufficient; effective reasoning also requires explicit "retrieval logic" to determine when, what, and how to retrieve relevant evidence (Schick et al., 2023; Zhang et al., 2024). Search-o1 (Li et al., 2025a) integrates retrieval into the chain-of-thought, while ReAct (Yao et al., 2023) formalizes a "reasoning–acting–observing–reasoning" paradigm for targeted external information (Li et al., 2025b). These

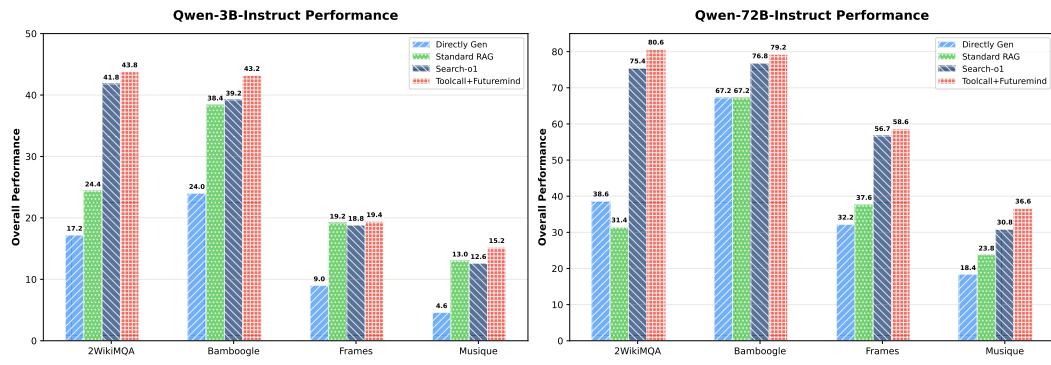


Figure 1: Overall performance comparison of FutureMind with other methods across four multi-hop QA benchmarks. The left panel depicts the performance on a 3B small language model (SLM), while the right panel illustrates the performance on a 72B large language model (LLM).

approaches illustrate a long-standing consensus: LLM capabilities should be activated dynamically and on demand during inference (Wang et al., 2024b; Jin et al., 2024b). For autonomous agents, this shifts the objective from answering directly to reasoning systematically—analyzing, decomposing, and integrating evidence for deeper insight and more robust strategies.

However, implementing explicit retrieval logic places substantial demands on model capabilities. LLMs are proficient in multi-turn reasoning and retrieval but incur prohibitive latency and computational costs (Wan et al., 2023; Wang, 2024). In contrast, SLMs offer notable advantages in efficiency, cost, and privacy, but their limited memory, weak context retention, and restricted structured reasoning hinder effective problem decomposition, iterative evidence retrieval, and multi-hop aggregation (Wang et al., 2024a; Xu et al., 2025). Consequently, achieving an optimal balance between reasoning effectiveness and computational efficiency remains a critical and unresolved challenge (Bai et al., 2024), particularly for resource-constrained models deployed in real-time or privacy-sensitive scenarios.

To this end, we propose **FutureMind**, a training-free modular reasoning framework that enables low-latency and high-accuracy complex reasoning without gradient updates, leveraging an adaptive thinking-pattern distillation strategy. The name FutureMind reflects our vision for future AI systems: even under constrained resources, the model can draw on distilled thinking-pattern priors to generalize to high-difficulty and unseen problems with free training. FutureMind decomposes reasoning into a four-stage pipeline—**Problem Analysis**, **Logical Reasoning**, **Strategy Planning**, and **Retrieval Guidance**—which sequentially address whether to retrieve, what to retrieve, how to integrate retrieved evidence, and how to generate a coherent answer. To further reduce retrieval overhead, we design three retrieval paradigms based on the decomposition of complex-question retrieval logic: (1) **Forward Stepwise Reasoning**—progressive expansion of sub-queries; (2) **Backward Constraint Focusing**—start from answer constraints and narrow search; (3) **Parallel Intersection Reasoning**—run parallel sub-searches and intersect evidence. By completing the reasoning strategy within a single turn, the framework endows models with clear planning, retrieval, and knowledge-synthesis capabilities, bridging the gap between reasoning depth and efficiency. The contributions of this paper are summarised as follows:

1. **A training-free modular reasoning framework:** We propose FutureMind, a four-stage pipeline (Problem Analysis, Logic Reasoning, Strategy Planning, Retrieval Guidance) supplemented by a dynamic thinking module that provides explicit knowledge support for structured reasoning. The framework is applicable to both LLMs and SLMs, balancing accuracy and efficiency, as shown in Figure 1.
2. **Composable retrieval strategies:** We design three adaptive retrieval paradigms (forward stepwise reasoning, backward constraint focusing, parallel intersection reasoning) that decompose complex multi-hop questions into manageable sub-queries and perform evidence integration efficiently.

108 3. **Systematic experiments and cognitive insights:** Experiments on four multi-hop QA
 109 benchmarks demonstrate that FutureMind consistently improves performance across models
 110 of various architectures and scales, with the largest gains on SLMs, establishing a new state
 111 of the art among training-free methods. Moreover, we identify a "cognitive-bias bottleneck":
 112 once the teacher's plan surpasses the student's capacity, distillation becomes lossy, snapping
 113 reasoning chains and amplifying noise. This emphasizes the importance of teacher-student
 114 compatibility over raw model size, offering guidance for the design of lightweight yet
 115 scalable reasoning systems.

116 117 **2 RELATED WORK**

118 119 **Large Language Models and Retrieval.** LLMs (Achiam et al., 2023; Team, 2024) exhibit strong
 120 reasoning and code-generation abilities (Guo et al., 2025; 2024), but remain prone to hallucination
 121 due to their reliance on static parametric knowledge (Zhang et al., 2023). To mitigate this, external
 122 search is widely adopted via (i) retrieval-augmented generation (RAG) (Gao et al., 2023), which
 123 integrates retrieved evidence into the generation process, and (ii) search-as-a-tool (Schick et al.,
 124 2023), where LLMs explicitly interact with a search engine through prompting (Trivedi et al.,
 125 2022; Schick et al., 2023) or fine-tuning (Schick et al., 2023). However, RAG's static, single-stage
 126 retrieval—i.e., a non-adaptive, one-shot lookup that ignores query complexity and intermediate
 127 generation signals—can return irrelevant or weakly informative passages, impeding compositional
 128 and multi-hop reasoning (Jin et al., 2024a); tool-based approaches, though more interactive, still
 129 struggle to retrieve evidence that is sufficiently relevant and precise for complex, multi-step inference.

130 131 **Small Language Models and Cognitive Transfer.** SLMs are attractive for cost-sensitive, low-
 132 latency, and privacy-preserving applications, yet they exhibit pronounced deficiencies in memory,
 133 context propagation, and structured, multi-step reasoning (Fu et al., 2023; Hsieh et al., 2023). To
 134 close this gap, **Cognitive-Transfer techniques** attempt to migrate reasoning behaviors from larger
 135 models. **CoT Distillation** transfers step-by-step traces (Wei et al., 2022b; Wang et al., 2023; Fu
 136 et al., 2023) but provides limited adaptivity and can be brittle under distributional or stylistic shift.
 137 **Prompt Distillation** reduces stylistic mismatch by extracting compact prompts (Li et al., 2023a;
 138 Chen & Feng, 2023), yet typically encodes mostly static knowledge templates that do not support
 139 dynamic planning. Retrieval-augmented transfers such as **Meta-RAG** improve efficiency through
 140 external knowledge (Mombaerts et al., 2024), but commonly treat retrieval as a fixed, non-adaptive
 141 pipeline and thus fail to fully integrate retrieval with adaptive reasoning. Overall, these approaches
 142 only partially mitigate SLMs' reasoning deficits and lack the generalizability and dynamic adaptivity
 143 required for robust problem decomposition, iterative retrieval, and multi-hop aggregation—motivating
 144 methods that endow SLMs with lightweight, structured reasoning routines and strategic retrieval
 145 policies.

146 147 **3 METHODOLOGY**

148 149 **3.1 OVERVIEW**

150 151 FutureMind is a modular reasoning framework that employs **adaptive knowledge distillation** to
 152 transfer structured reasoning and retrieval strategies from teacher models to student models. Unlike
 153 conventional distillation methods, which primarily focus on compressing knowledge representations,
 154 FutureMind targets the distillation of **systematic thinking patterns**. Specifically, it captures the
 155 complete logical chain from problem definition to retrieval guidance, abstracting these patterns
 156 into lightweight, reusable strategic thinking-pattern priors. This design enables student models,
 157 particularly SLM, to perform adaptive reasoning and deep, structured retrieval planning, thereby
 158 achieving superior retrieval performance even in resource-constrained environments.

159 160 As depicted in Figure 2, FutureMind is coordinated by the **Thinking Module**, which dynamically
 161 generates the optimal retrieval strategies based on task characteristics, data availability, and efficiency
 162 constraints. It of four core modules: **Problem Analysis**, **Logical Reasoning**, **Strategy Planning**,
 163 and **Retrieval Guidance**, achieving modularity, interpretability, and end-to-end optimization.

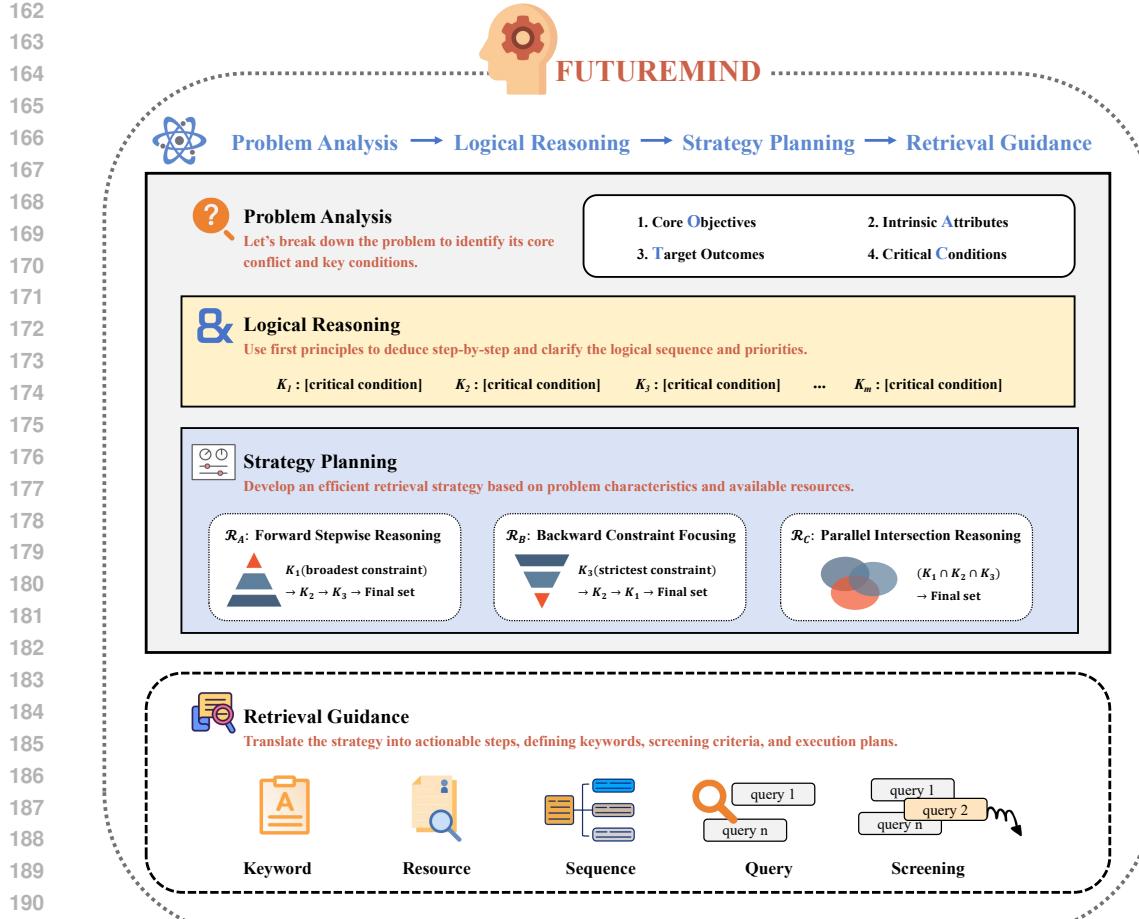


Figure 2: Overview of the FutureMind framework.

Formally, FutureMind is a four-stage pipeline coordinated by the Thinking Module \mathcal{M} :

$$F = \mathcal{M}(\mathcal{P}, \mathcal{L}, \mathcal{S}, \mathcal{R}), \quad (1)$$

where \mathcal{P} , \mathcal{L} , \mathcal{S} , and \mathcal{R} represent Problem Analysis, Logical Reasoning, Strategy Planning, and Retrieval Guidance, respectively.

For clarity and reproducibility, we further provide module-wise Instructions and execution examples in the Appendix E.5.1- E.5.4, illustrating how each component operates within the overall framework.

Subsequently, we provide a comprehensive overview of the four core modules of FutureMind.

3.2 MODULE DEFINITIONS

3.2.1 PROBLEM ANALYSIS \mathcal{P}

The Problem Analysis module initiates the reasoning pipeline by decomposing the input query x into its fundamental components. This decomposition yields a structured representation that enables subsequent reasoning and decision-making processes. Specifically, this module identifies the following key elements:

$$\mathcal{P}(x) \rightarrow (\mathcal{O}, \mathcal{A}, \mathcal{T}, \mathcal{C}), \quad (2)$$

where:

- \mathcal{O} represents the core objectives, which define the primary direction and desired outcomes of the problem-solving process.

- \mathcal{A} denotes the intrinsic attributes, characterizing the inherent properties and conditions of the problem.
- \mathcal{T} specifies the target outcomes, indicating the expected results or output types upon problem resolution.
- $\mathcal{C} = \{C_1, C_2, \dots, C_n\}$ captures the key dimensions, which are critical conditions or factors within the problem, each C_i representing a specific dimension.

By systematically decomposing the input query into these components, the Problem Analysis module establishes a clear, structured foundation for the subsequent reasoning stages.

3.2.2 LOGICAL REASONING \mathcal{L}

The Logical Reasoning module applies a *first-principles approach* to derive core problem mechanisms. It identifies the most fundamental principles and applies deductive reasoning to construct a coherent abstraction. Unlike heuristic or analogy-based reasoning, this approach grounds inference in causal structures, reducing reliance on incomplete prior knowledge. Formally:

$$\mathcal{L}(\mathcal{O}, \mathcal{A}, \mathcal{T}, \mathcal{C}) \rightarrow (\mathcal{M}, \mathcal{K}), \quad (3)$$

where:

- \mathcal{O} , \mathcal{A} , \mathcal{T} , and \mathcal{C} denote the core objectives, intrinsic attributes, target outcomes, and key dimensions extracted from Problem Analysis.
- \mathcal{M} represents the mechanistic understanding, capturing causal relations and fundamental principles governing the system.
- $\mathcal{K} = \{K_1, K_2, \dots, K_m\}$ is an ordered set of critical conditions, prioritized by logical dependency and discriminative importance.

Grounding reasoning in first principles improves interpretability and adaptability, especially in complex or knowledge-intensive tasks where heuristics are insufficient. By decomposing the problem into basic components, identifying key conditions, and arranging them in order of logical priority, the module converts the structured input from Problem Analysis into a mechanistic abstraction $(\mathcal{M}, \mathcal{K})$. This abstraction serves as the foundation for Strategy Planning and Retrieval Guidance.

3.2.3 STRATEGY PLANNING \mathcal{S}

The Strategy Planning module bridges the mechanistic insights from Logical Reasoning and the normative layer of Retrieval Guidance. It dynamically determines the optimal retrieval strategy \mathcal{R}^* based on the mechanistic understanding \mathcal{M} and the prioritized condition sequence $\mathcal{K} = \{K_1, K_2, \dots, K_m\}$. Formally:

$$\mathcal{S}(\mathcal{M}, \mathcal{K}) \rightarrow \mathcal{R}^*, \quad \mathcal{R}^* = \arg \min_{\mathcal{R} \in \mathcal{P}_{\text{cand}}} \mathcal{F}(\mathcal{R}; \mathcal{M}, \mathcal{K}), \quad (4)$$

where:

- \mathcal{M} : mechanistic understanding derived from logical reasoning.
- $\mathcal{K} = \{K_1, K_2, \dots, K_m\}$: prioritized sequence of conditions.
- $\mathcal{P}_{\text{cand}} = \{\mathcal{R}_A, \mathcal{R}_B, \mathcal{R}_C\}$: candidate pool of reasoning strategies.
- $\mathcal{F}(\cdot)$: a cost function evaluating efficiency, constraints, interdependencies, and data availability. In this work, \mathcal{F} specifically refers to the function implemented by the teacher model to conduct a preliminary evaluation of retrieval plans, aiming to generate recommendations for the optimal retrieval strategy.

Based on insights from knowledge-intensive tasks, we design three distinct retrieval paradigms (Figure 4) to enable more efficient and accurate execution. The Strategy Planning module dynamically selects among them based on the condition set topology \mathcal{K} .

Let \mathcal{U} denote the candidate space. For each condition K_i , we define an evaluation function $\phi(K_i, x) \in \{0, 1\}$ that checks if candidate $x \in \mathcal{U}$ satisfies K_i . Intermediate sets are defined as:

$$X_i = \{x \in \mathcal{U} \mid \phi(K_i, x) = 1\}, \quad X^* = \bigcap_{i=1}^m X_i. \quad (5)$$

270 **Strategy A: Forward Stepwise Reasoning (\mathcal{R}_A)** Applied when early conditions are broad yet
 271 effective in pruning. Constraints are applied sequentially from general to specific:
 272

$$273 \quad X_1 = \{x \in \mathcal{U} \mid \phi(K_1, x) = 1\}, \quad X_j = \{x \in X_{j-1} \mid \phi(K_j, x) = 1\}, \quad X^* = \bigcap_{i=1}^m X_i. \quad (6)$$

276 **Strategy B: Backward Constraint Focusing (\mathcal{R}_B)** Adopted when downstream conditions are
 277 highly selective. Reasoning starts with the tightest constraint and broadens progressively:
 278

$$279 \quad X_m = \{x \in \mathcal{U} \mid \phi(K_m, x) = 1\}, \quad X_j = \{x \in X_{j+1} \mid \phi(K_j, x) = 1\}, \quad X^* = \bigcap_{i=1}^m X_i. \quad (7)$$

281 **Strategy C: Parallel Intersection Reasoning (\mathcal{R}_C)** Best suited for independent or orthogonal
 282 conditions. All constraints are processed in parallel, then intersected:
 283

$$284 \quad X_i = \{x \in \mathcal{U} \mid \phi(K_i, x) = 1\}, \quad X^* = \bigcap_{i=1}^m X_i. \quad (8)$$

287 By deeply understanding knowledge-intensive problems, the module adaptively selects the optimal
 288 retrieval strategy, ensuring both efficient and precise retrieval execution.
 289

290 3.2.4 RETRIEVAL GUIDANCE \mathcal{R}

291 The Retrieval Guidance module serves as a normative layer that transforms abstract reasoning and
 292 the selected retrieval strategy into structured instructions for execution. Unlike direct retrieval, this
 293 module generates prescriptive guidelines that specify how retrieval should be performed.
 294

295 Given the mechanistic understanding \mathcal{M} , the prioritized conditions \mathcal{K} , and the chosen strategy \mathcal{R}^* ,
 296 the module outputs a set of retrieval guidelines:
 297

$$298 \quad \mathcal{R}(\mathcal{M}, \mathcal{K}, \mathcal{R}^*) \rightarrow \Gamma, \quad (9)$$

299 where $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_q\}$ is a set of normative principles guiding the retrieval process (e.g., priority
 300 order, source preferences, evaluation criteria).
 301

302 The guidance is structured into five key, complementary stages:
 303

- 304 • **Keyword Guidance.** Extract core entities, attributes, and relations from \mathcal{K} and specify the
 305 lexical and semantic variants that retrieval should prioritize. This guidance outlines the
 306 dimensions along which queries can vary, enabling adaptive retrieval across domains while
 307 maintaining alignment with the underlying reasoning structure.
 308
- 309 • **Resource Guidance.** Indicate categories of information sources (e.g., academic databases,
 310 industry reports, policy documents) ranked by relevance \mathcal{M} and credibility, guiding retrieval
 311 toward reliable knowledge domains.
 312
- 313 • **Sequence Guidance.** Provide recommendations on the ordering of retrieval steps in accord-
 314 ance with \mathcal{R}^* . For instance, a Forward Stepwise strategy begins with broad, high-recall
 315 repositories to establish initial coverage before moving to domain-specific collections. Con-
 316 versely, a Backward Constraint strategy starts with highly selective regulatory to anchor the
 317 search with high-precision evidence, then expands outward as needed.
 318
- 319 • **Query Guidance.** Provide structural templates for query formulation (e.g., Boolean patterns,
 320 semantic expansions, hierarchical constraints), emphasizing inclusiveness in initial searches
 321 and progressive narrowing in later stages. This guidance offers adaptable design principles
 322 rather than fixed query strings.
 323
- 324 • **Screening Guidance.** Define the principles for evaluating retrieved results, including
 325 their relevance to \mathcal{M} , source credibility, and methodological rigor. The module specifies
 326 evaluation criteria conceptually.
 327

328 By structuring guidance around keywords, resources, sequencing, query formulation, and evaluation,
 329 the module bridges cognitive strategy with retrieval while maintaining executional independence.
 330

324

4 EXPERIMENTS

325

4.1 EXPERIMENTAL SETUP

326 **Datasets.** We evaluate our approach on four widely-used multi-hop question answering (QA)
 327 benchmarks: **2WikiMultihopQA (2WikiMQA)** (Ho et al., 2020a), **Bamboogle** (Press et al., 2022),
 328 **MuSiQue** (Tang & Yang, 2024), and **FRAMES** (Krishna et al., 2024). Specifically, we randomly
 329 sample 500 instances from the validation sets of 2WikiMQA and MuSiQue, while evaluating on the
 330 full test sets of Bamboogle and FRAMES.

331 **Metrics.** For evaluation, following prior work (Sun et al., 2025), we adopt two complementary
 332 metrics: **Coverage-based Exact Match** (ACC_E) and **LLM-as-Judge** (ACC_L). ACC_E measures
 333 whether the predicted answer fully covers the gold reference while allowing for paraphrastic variations;
 334 its detailed calculation formula is provided in Appendix G.1. In contrast, ACC_L employs GPT-4o-
 335 mini as an automatic evaluator to judge the semantic correctness of predicted answers relative to the
 336 gold reference. The full evaluation prompt for ACC_L is provided in Appendix F.1.

337 **Baselines.** We consider three categories of baselines: (1) **Naive Generation**: Generates answers
 338 without retrieval. (2) **Standard RAG** (Zhao et al., 2024): Retrieves documents using the original
 339 question as the query. (3) **Search-o1** (Li et al., 2025a): Performs self-initiated retrieval using prompts.

340 **Implementation Details.** We evaluate the effectiveness of the proposed **FutureMind** method using
 341 models at different architectures and scales (Qwen-2.5-3B/7B/14B/32B/72B-Instruct and Llama3.1-
 342 8B-Instruct). For generation, we set the maximum sequence length to 32768 tokens, with temperature
 343 = 0.0, top-p = 0.8, top-k = 20, and repetition penalty = 1.05 across all models. For retrieval, we
 344 employ the Google Web Search API, retrieving the top k = 10 results. In experiments, FutureMind
 345 leverages an enhanced version of **Toolcall (TC)**, a ReAct-Style orchestration framework¹. We modify
 346 the original TC framework by replacing its single search process with parallel search, enabling more
 347 efficient and robust aggregation of retrieved evidence. This configuration is referred to as **TC+FM**.
 348 Details of implementation are provided in Appendix E.

349

4.2 MAIN RESULTS

350 Table 1 compares the performance of different model architectures and scales across four multi-hop
 351 QA benchmarks, under four methods: naive generation, standard RAG, Search-o1, and ToolCall-
 352 driven FutureMind. Several key observations emerge from the results. Additional benchmark results
 353 are presented in Table 6 in Appendix B due to space limitations.

- 354 **1. Inherent Limitations of Baseline Methods.** Naive generation (internal knowledge only) yields
 355 the lowest accuracy, underscoring its inability to integrate external evidence. While standard
 356 RAG (retrieval-enabled) cannot reliably perform multi-step reasoning and may even underperform
 357 naive generation when reasoning integration fails. Search-o1 (reason-in-documents) enhances
 358 retrieval quality but remains limited: small models benefit minimally, and even larger models are
 359 constrained by these intrinsic summarization and integration capacity.
- 360 **2. Effectiveness of FutureMind with Adaptive Knowledge Distillation.** Unlike Search-o1’s
 361 fixed-prompt design, FutureMind employs adaptive knowledge, enabling a more flexible and
 362 effective problem-solving process that yields consistent performance gains. For instance, Qwen-
 363 3B improves on Frames (ACC_E : 11.77 → 18.84), Llama3.1-8B on Bamboogle (ACC_L : 52.00 →
 364 64.00), and Qwen-72B on 2WikiMQA (ACC_L : 75.40 → 80.60). By providing adaptive external
 365 strategy empowerment rather than depending solely on internal capability, FutureMind achieves
 366 stronger and more scalable reasoning, particularly under resource-constrained settings.
- 367 **3. Universal Applicability of FutureMind Across Model Architectures and Scales.** TC+FM*
 368 achieves state-of-the-art results in nearly all settings, delivering scalable improvements across both
 369 model architectures and parameter scales. This demonstrates FutureMind’s broad effectiveness in
 370 enhancing multi-hop reasoning via external strategy transfer, alleviating capability bottlenecks in
 371 resource-limited models, while maintaining strong performance in larger models.

372 ¹<https://github.com/QwenLM/Qwen-Agent/>

378
379 Table 1: Main results on four multi-hop QA benchmarks. **Bold** denotes the best performance and
380 underline indicates the second best. Rows with blue background correspond to our methods TC+FM*,
381 where FM* selects the best-performing FutureMind-enhanced variant for each base model.

382 Model	383 Method	2WikiMQA		Bamboogle		Frames		MuSiQue		384 AVG	
		385 ACC _E	386 ACC _L	387 ACC _E	388 ACC _L	389 ACC _E	390 ACC _L	391 ACC _E	392 ACC _L	393 ACC _E	394 ACC _L
385 Qwen-3B	Naive Gen	16.80	17.20	20.80	24.00	5.94	8.98	3.60	4.60	11.79	13.70
	Standard RAG	24.00	24.40	26.40	38.40	12.01	19.17	10.20	13.00	18.15	23.74
	Search-01	41.00	41.80	34.40	39.20	11.77	18.81	10.40	12.60	24.39	28.10
	TC+FM*	56.40	43.80	39.20	43.20	18.84	19.42	14.20	15.20	32.16	30.41
388 Qwen-7B	Naive Gen	29.40	25.20	34.40	37.60	11.29	16.87	7.60	10.80	20.67	22.62
	Standard RAG	30.20	29.80	42.40	<u>52.80</u>	15.78	24.76	13.20	16.80	25.39	31.04
	Search-01	57.80	59.80	43.20	51.20	24.63	38.34	20.80	23.80	36.61	43.29
	TC+FM*	62.00	64.00	58.40	64.80	25.12	34.71	<u>20.00</u>	<u>23.80</u>	41.38	46.83
392 Qwen-14B	Naive Gen	30.40	30.80	48.80	55.20	14.81	22.82	8.80	12.40	25.70	30.30
	Standard RAG	27.40	28.40	44.80	<u>56.00</u>	17.96	28.40	14.00	18.60	26.04	32.85
	Search-01	66.80	68.40	43.20	55.20	30.46	46.48	<u>20.60</u>	<u>25.60</u>	40.27	48.92
	TC+FM*	71.60	75.20	70.40	72.80	34.83	49.51	24.00	28.20	50.21	56.43
395 Qwen-32B	Naive Gen	30.80	31.30	54.40	60.80	15.66	24.51	10.80	15.20	27.91	32.95
	Standard RAG	24.60	24.40	52.80	61.60	19.78	30.95	16.20	19.60	28.35	34.14
	Search-01	68.60	71.60	60.80	67.20	34.34	54.12	22.80	27.80	46.63	55.18
	TC+FM*	74.40	77.80	68.80	72.80	37.15	53.86	26.00	30.40	51.59	58.71
400 Qwen-72B	Naive Gen	38.20	38.60	60.00	67.20	21.12	32.16	12.80	18.40	33.03	39.09
	Standard RAG	31.00	31.40	59.20	67.20	25.97	37.62	19.00	23.80	33.79	40.01
	Search-01	72.60	75.40	67.20	72.80	37.37	56.67	24.60	30.80	50.44	58.92
	TC+FM*	74.20	80.60	75.20	79.20	41.38	58.59	28.40	36.60	54.80	63.75
402 Llama3.1-8B	Naive Gen	38.20	38.60	60.00	67.20	21.12	32.16	12.80	18.40	33.03	39.09
	Standard RAG	29.20	30.40	39.20	47.20	15.05	22.82	12.20	15.20	23.91	28.90
	Search-01	54.00	56.00	46.40	52.00	24.88	37.62	15.40	18.20	35.17	40.95
	TC+FM*	55.20	56.80	58.40	<u>64.00</u>	27.43	39.92	21.80	25.20	40.71	46.48

405
406 Table 2: Impact of Teacher Model Scale on Student Performance in multi-hop QA benchmarks. **Bold**
407 denotes the best performance and underline indicates the second best. Rows with blue background
408 correspond to the best teacher model scale for each student model.

409 Model	410 Method	2WikiMQA		Bamboogle		Frames		MuSiQue		411 Avg	
		412 ACC _E	413 ACC _L	414 ACC _E	415 ACC _L	416 ACC _E	417 ACC _L	418 ACC _E	419 ACC _L	420 ACC _E	421 ACC _L
413 Qwen-3B	TC	54.20	42.40	37.60	40.00	17.96	22.94	13.00	12.60	30.69	29.49
	TC+FM (3B)	42.00	30.80	28.00	30.40	11.53	10.32	7.40	8.60	22.23	20.03
	TC+FM (7B)	53.00	39.60	30.40	36.00	15.78	16.26	11.20	11.40	27.60	25.82
	TC+FM (14B)	55.20	45.60	40.80	42.40	17.11	18.46	12.20	12.80	31.33	29.82
	TC+FM (32B)	49.20	37.00	36.00	37.60	12.26	12.01	10.20	10.40	26.92	24.25
	TC+FM (72B)	56.40	43.80	<u>39.20</u>	43.20	17.84	19.42	14.20	15.20	31.91	30.41
417 Qwen-7B	TC	56.80	56.20	49.60	54.40	23.78	<u>32.28</u>	16.40	18.80	36.65	40.42
	TC+FM (3B)	60.20	60.00	49.60	50.40	23.09	30.83	17.00	19.60	37.97	40.21
	TC+FM (7B)	60.20	<u>61.80</u>	53.60	57.60	24.39	33.55	17.60	21.20	38.95	43.04
	TC+FM (14B)	62.00	64.00	58.40	64.80	<u>25.12</u>	34.71	20.00	23.80	41.38	46.83
	TC+FM (32B)	57.80	57.60	52.00	58.40	22.57	29.98	15.20	19.40	36.89	41.35
	TC+FM (72B)	60.40	60.00	<u>56.80</u>	61.60	26.58	34.71	18.20	21.20	40.50	44.38

422 4.3 IMPACT OF TEACHER MODEL DESIGN ON TEACHER–STUDENT COGNITIVE ALIGNMENT

423 We systematically analyze the impact of teacher model design on student performance in knowledge
424 distillation (Table 2). Several consistent patterns emerge:

425 1. **Small-scale teachers models degrade student performance.** In TC+FM, using a 3B teacher
426 for Qwen-3B reduces average performance (ACC_E: 30.69 → 22.23, ACC_L: 29.49 → 20.03),
427 indicating that low-capacity teachers may generate noisy or misleading planning signals, hindering
428 transfer effectiveness.

432 Table 3: Ablation study of modular components on multi-hop QA benchmarks. **Bold** denotes the
 433 performance with all modules enabled.

Model	Method	2WikiMQA		Bamboogle		Frames		MuSiQue		Avg	
		ACC _E	ACC _L								
Qwen-7B	All Modules	62.00	64.00	58.40	64.80	25.12	34.71	20.00	23.80	41.38	46.83
	- w/o Problem Analysis	58.20	57.40	56.00	62.40	24.93	34.57	16.00	21.00	38.78	43.84
	- w/o Logical Reasoning	60.40	59.00	49.60	53.60	24.59	34.21	19.20	23.00	38.45	42.45
	- w/o Strategy Planning	56.40	57.20	49.30	51.20	22.63	33.10	17.60	19.80	36.48	40.33
	- w/o Retrieval Guidance	59.20	60.40	57.60	60.00	23.39	33.34	18.40	20.60	39.65	43.59

442 Table 4: Ablation study of retrieval strategies on multi-hop QA benchmarks. **Bold** denotes the
 443 performance with all three retrieval strategies distilled from the Qwen-14B teacher.

Model	Method	2WikiMQA		Bamboogle		Frames		MuSiQue		Avg	
		ACC _E	ACC _L								
Qwen-7B	All Strategies	62.00	64.00	58.40	64.80	25.12	34.71	20.00	23.80	41.38	46.83
	- w/o Strategy A	57.40	57.80	54.20	60.00	24.64	32.16	16.60	21.20	38.21	42.79
	- w/o Strategy B	58.80	58.00	57.60	61.60	25.09	34.47	18.40	22.60	39.97	44.67
	- w/o Strategy C	60.40	59.00	54.40	60.80	24.72	33.37	17.40	21.40	39.23	43.64

453 2. **Mid-scale teachers provide optimal alignment.** Both student models benefit most from the
 454 14B teacher(Qwen-3B: ACC_E: 31.33, ACC_L: 29.82; Qwen-7B: ACC_E: 41.38, ACC_L: 46.83),
 455 outperforming the 32B variant and matching or exceeding the 72B model on average.

456 3. **Cognitive compatibility outweighs raw scale.** Although the 72B teacher excels in certain sub-
 457 tasks, it does not consistently surpass the 14B teacher on average (Qwen-7B student: 72B teacher
 458 ACC_E: 40.50, ACC_L: 44.38 < 14B teacher ACC_E: 41.38, ACC_L: 46.83), suggesting that cognitive
 459 alignment between teacher and student plays a more critical role in distillation effectiveness than
 460 raw scale.

461 The "cognitive bias bottleneck" further demonstrates that overly complex teacher plans may fail to
 462 transfer reasoning capabilities to smaller students, as strategic information loss can disrupt critical
 463 reasoning chains or amplify noise. Therefore, in knowledge distillation, prioritizing teacher–student
 464 compatibility is more important than considering raw model size. Future work should systematically
 465 quantify planning quality and evaluate generalization across tasks and alignment strategies to ensure
 466 scalable reasoning in lightweight models.

467 Beyond scale, the teacher's architecture, reasoning orientation, and instruction tuning must be
 468 compatible with the student. Experiments in Appendix B show that teachers with architectures
 469 well-aligned to the student consistently enable more effective thinking-pattern distillation, leading to
 470 improved multi-hop reasoning performance.

4.4 ABLATION STUDIES

475 We first evaluate the contributions of the four core modules of FutureMind: Problem Analysis, Logical
 476 Reasoning, Strategy Planning, and Retrieval Guidance. As shown in Table 3, removing any module
 477 leads to noticeable performance drops, confirming their complementary roles. Among them, Strategy
 478 Planning has the largest impact, highlighting its central role in converting structured reasoning into
 479 effective retrieval actions.

480 Next, we analyze the three retrieval strategies within the Strategy Planning module. Table 4 shows that
 481 removing any single strategy—Forward Stepwise Reasoning (\mathcal{R}_A), Backward Constraint Focusing
 482 (\mathcal{R}_B), or Parallel Intersection Reasoning (\mathcal{R}_C)—degrades performance. Overall, \mathcal{R}_A is most critical,
 483 \mathcal{R}_C contributes more on datasets with independent conditions, and \mathcal{R}_B benefits constraint-focused
 484 cases.

485 Finally, we evaluate the overall impact of FutureMind by comparing enhanced ToolCall alone versus
 ToolCall-driven FutureMind. Figure 3 shows that removing the FutureMind module consistently

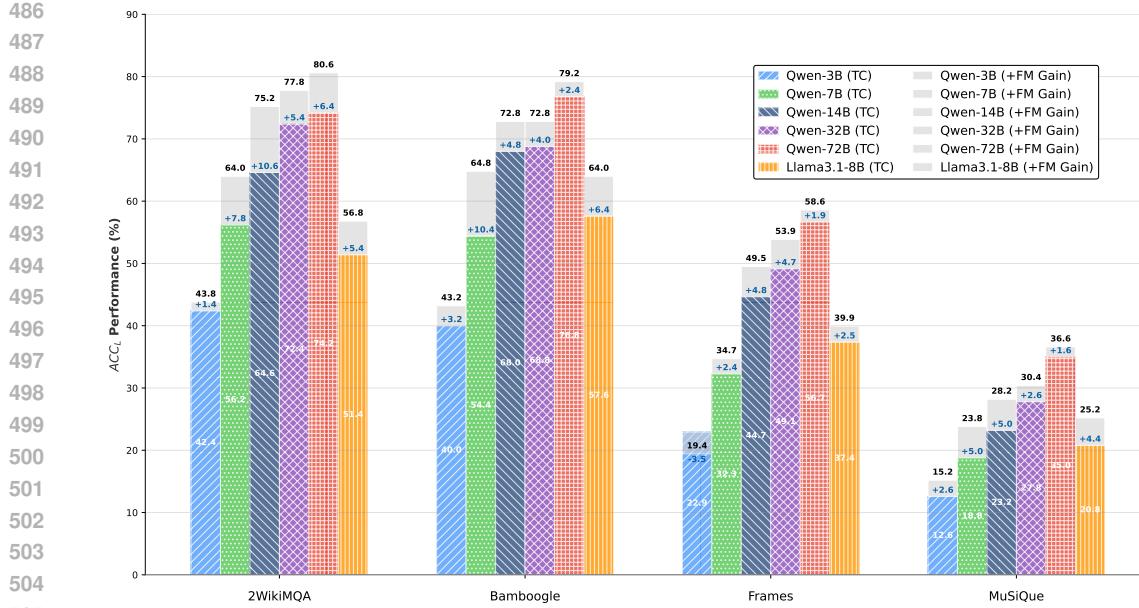


Figure 3: Ablation study of enhanced ToolCall across different models.

lowers performance, demonstrating its essential role in coordinating multi-hop reasoning and retrieval planning. The detailed numerical results are provided in Table 7.

Together, these ablations show that FutureMind, along with its modules and retrieval strategies, establishes an effective and coherent reasoning framework.

5 CONCLUSION

We introduce **FutureMind**, a training-free modular reasoning framework that enables both large and small language models to perform efficient, accurate, and structured reasoning. By decomposing reasoning into **Problem Analysis**, **Logical Reasoning**, **Strategy Planning**, and **Retrieval Guidance**, and leveraging adaptive retrieval strategies, FutureMind provides clear guidance on when, what, and how to retrieve evidence, effectively balancing reasoning depth and efficiency.

Evaluations on four multi-hop QA benchmarks show consistent gains across model scales and architectures, with the largest improvements in SLMs. We further identify a *cognitive bias bottleneck* in teacher-student thinking-pattern distillation: overly complex teacher plans can overwhelm student capacity, causing strategic information loss and degraded reasoning. Effective distillation requires both scale and architectural alignment, as structurally compatible teachers with aligned reasoning orientations enable more effective strategy transfer.

In summary, FutureMind shows that structured, adaptive reasoning is achievable even for SLMs, turning them into cognitively capable agents through strategic thinking-pattern priors.

ETHICS STATEMENT

We strictly adhere to the ICLR Code of Ethics. This work only uses four publicly available multi-hop QA datasets and does not involve any personal or sensitive information, nor does it recruit human subjects. There are no conflicts of interest or foreseeable ethical risks associated with this study. Our research does not introduce ethical concerns beyond the scope of standard multi-hop question answering tasks.

540 REPRODUCIBILITY STATEMENT
541

542 We will publicly release all experimental code and data processing scripts upon paper acceptance
543 to ensure reproducibility. The four datasets used are publicly available (e.g., 2WikiMultihopQA,
544 MuSiQue, Bamboogle, Frames), and the appendix details data preprocessing, retrieval configurations,
545 and relevant hyperparameters. The main text provides sufficient descriptions of the model architecture,
546 reasoning pipeline, and evaluation metrics to allow other researchers to replicate our results.
547

548 REFERENCES
549

550 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman,
551 Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report.
552 *arXiv preprint arXiv:2303.08774*, 2023.

553 Salaheddin Alzubi, Creston Brooks, Purva Chiniya, Edoardo Contente, Chiara von Gerlach, Lu-
554 cas Irwin, Yihan Jiang, Arda Kaz, Windsor Nguyen, Sewoong Oh, et al. Open deep search:
555 Democratizing search with open-source reasoning agents. *arXiv preprint arXiv:2503.20201*, 2025.
556

557 Guangji Bai, Zheng Chai, Chen Ling, Shiyu Wang, Jiaying Lu, Nan Zhang, Tingwei Shi, Ziyang Yu,
558 Mengdan Zhu, Yifei Zhang, et al. Beyond efficiency: A systematic survey of resource-efficient
559 large language models. *arXiv preprint arXiv:2401.00625*, 2024.

560 Feng Chen and Yujian Feng. Chain-of-thought prompt distillation for multimodal named entity
561 recognition and multimodal relation extraction. *arXiv preprint arXiv:2306.14122*, 2023.

563 Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language
564 models towards multi-step reasoning. In *International Conference on Machine Learning*, pp.
565 10421–10430. PMLR, 2023.

566 Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Haofen
567 Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey.
568 *arXiv preprint arXiv:2312.10997*, 2, 2023.

570 Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi,
571 Yu Wu, YK Li, et al. Deepseek-coder: When the large language model meets programming—the
572 rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

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

577 Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop
578 qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*,
579 2020a.

580 Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop
581 qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International
582 Conference on Computational Linguistics*, pp. 6609–6625, 2020b.

584 Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner,
585 Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger
586 language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*,
587 2023.

588 Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. Long-context llms meet rag: Overcoming
589 challenges for long inputs in rag. In *The Thirteenth International Conference on Learning
590 Representations*, 2024a.

592 Haolin Jin, Linghan Huang, Haipeng Cai, Jun Yan, Bo Li, and Huaming Chen. From llms to llm-
593 based agents for software engineering: A survey of current, challenges and future. *arXiv preprint
arXiv:2408.02479*, 2024b.

594 Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam
 595 Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-
 596 augmented generation. *arXiv preprint arXiv:2409.12941*, 2024.

597 Lei Li, Yongfeng Zhang, and Li Chen. Prompt distillation for efficient llm-based recommendation. In
 598 *Proceedings of the 32nd ACM international conference on information and knowledge management*,
 599 pp. 1348–1357, 2023a.

600 Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang,
 601 and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *CoRR*,
 602 abs/2501.05366, 2025a. doi: 10.48550/ARXIV.2501.05366. URL <https://doi.org/10.48550/arXiv.2501.05366>.

603 Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and
 604 Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*, 2025b.

605 Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and
 606 Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability.
 607 *CoRR*, abs/2504.21776, 2025c. doi: 10.48550/ARXIV.2504.21776. URL <https://arxiv.org/abs/2504.21776>.

608 Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. Large language models in finance: A survey.
 609 In *Proceedings of the fourth ACM international conference on AI in finance*, pp. 374–382, 2023b.

610 Laurent Mombaerts, Terry Ding, Adi Banerjee, Florian Felice, Jonathan Taws, and Tarik Borogovac.
 611 Meta knowledge for retrieval augmented large language models. *arXiv preprint arXiv:2408.09017*,
 612 2024.

613 OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774.
 614 URL <https://doi.org/10.48550/arXiv.2303.08774>.

615 Cheng Peng, Xi Yang, Aokun Chen, Kaleb E Smith, Nima PourNejatian, Anthony B Costa, Cheryl
 616 Martin, Mona G Flores, Ying Zhang, Tanja Magoc, et al. A study of generative large language
 617 model for medical research and healthcare. *NPJ digital medicine*, 6(1):210, 2023.

618 Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring
 619 and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*,
 620 2022.

621 Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi
 622 Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-
 623 text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL <https://jmlr.org/papers/v21/20-074.html>.

624 Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke
 625 Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach
 626 themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551,
 627 2023.

628 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Huawei Zhang,
 629 Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical
 630 reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

631 Zhongxiang Sun, Qipeng Wang, Weijie Yu, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Yang
 632 Song, and Han Li. Rearter: Retrieval-augmented reasoning with trustworthy process rewarding. In
 633 *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in
 634 Information Retrieval*, pp. 1251–1261, 2025.

635 Yixuan Tang and Yi Yang. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop
 636 queries. *arXiv preprint arXiv:2401.15391*, 2024.

637 Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.
 638 *arXiv preprint arXiv:2403.05530*, 2024.

648 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée
 649 Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand
 650 Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language
 651 models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/ARXIV.2302.13971. URL <https://doi.org/10.48550/arXiv.2302.13971>.

653 Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval
 654 with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint*
 655 *arXiv:2212.10509*, 2022.

656

657 Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen
 658 Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. Efficient large language
 659 models: A survey. *CoRR*, abs/2312.03863, 2023. doi: 10.48550/ARXIV.2312.03863. URL
 660 <https://doi.org/10.48550/arXiv.2312.03863>.

661 Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, Tzuhalo Mo, Qiuhalo Lu, Wanjing Wang, Rui
 662 Li, Junjie Xu, Xianfeng Tang, et al. A comprehensive survey of small language models in the era
 663 of large language models: Techniques, enhancements, applications, collaboration with llms, and
 664 trustworthiness. *ACM Transactions on Intelligent Systems and Technology*, 2024a.

665 Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai
 666 Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents.
 667 *Frontiers of Computer Science*, 18(6):186345, 2024b.

668

669 Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. Scott: Self-
 670 consistent chain-of-thought distillation. *arXiv preprint arXiv:2305.01879*, 2023.

671 Zeyu Wang. Causalbench: A comprehensive benchmark for evaluating causal reasoning capabilities
 672 of large language models. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language*
 673 *Processing (SIGHAN-10)*, pp. 143–151, 2024.

674

675 Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan
 676 Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In
 677 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event,*
 678 *April 25-29, 2022*. OpenReview.net, 2022a. URL <https://openreview.net/forum?id=gEZrGCozdqR>.

679

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

683 Guangzhi Xiong, Qiao Jin, Xiao Wang, Yin Fang, Haolin Liu, Yifan Yang, Fangyuan Chen, Zhixing
 684 Song, Dengyu Wang, Minjia Zhang, et al. Rag-gym: Optimizing reasoning and search agents with
 685 process supervision. *arXiv preprint arXiv:2502.13957*, 2025.

686 Weihong Xu, Haein Choi, Po-kai Hsu, Shimeng Yu, and Tajana Simunic. Slim: A heterogeneous
 687 accelerator for edge inference of sparse large language model via adaptive thresholding. *ACM*
 688 *Transactions on Embedded Computing Systems*, 2025.

689

690 Meng Yang, Yihao Wang, and Yu Gu. Language-based reasoning graph neural network for common-
 691 sense question answering. *Neural Networks*, 181:106816, 2025.

692

693 Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov,
 694 and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question
 695 answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*
 696 *Processing*, pp. 2369–2380, 2018.

697

698 Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao.
 699 React: Synergizing reasoning and acting in language models. In *International Conference on*
 700 *Learning Representations (ICLR)*, 2023.

701

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao,
 Yu Zhang, Yulong Chen, et al. Siren’s song in the ai ocean: a survey on hallucination in large
 language models. *arXiv preprint arXiv:2309.01219*, 2023.

702 Table 5: Effect of Teacher Model Architecture on Student Performance. We evaluate student
 703 models (3B–72B) under different teacher model architectures, specifically Qwen2.5-72B-Instruct
 704 (denoted as Q2.5) and Llama3.1-70B-Instruct (denoted as L3.1) within the Toolcall-driven FutureMind
 705 (TC+FM). **Bold** denotes the best performance and underline indicates the second best. Rows with
 706 blue background correspond to the best teacher model architecture for each student model.

Model	Method	2WikiMQA		Bamboogle		Frames		MuSiQue		Avg	
		ACC _E	ACC _L								
Qwen-3B	TC	54.20	42.40	37.60	40.00	17.96	22.94	13.00	12.60	<u>30.69</u>	29.49
	TC+FM (L3.1)	47.80	34.00	27.20	27.20	11.53	9.83	10.20	10.40	24.18	20.36
	TC+FM (Q2.5)	56.40	43.80	39.20	43.20	17.84	19.42	14.20	15.20	31.91	30.41
Qwen-7B	TC	56.80	56.20	49.60	54.40	23.78	32.28	16.40	18.80	<u>36.65</u>	40.42
	TC+FM (L3.1)	53.60	50.20	48.80	51.20	19.54	26.58	14.20	17.20	34.03	36.30
	TC+FM (Q2.5)	57.80	57.60	52.00	58.40	22.57	29.98	15.20	19.40	36.89	41.34
Qwen-14B	TC	64.40	64.60	64.00	68.00	34.22	44.66	20.40	23.20	<u>45.76</u>	50.11
	TC+FM (L3.1)	63.40	63.80	63.60	67.20	28.16	37.86	22.80	25.20	44.49	48.52
	TC+FM (Q2.5)	70.00	71.60	64.00	68.00	35.92	48.06	25.80	28.20	48.93	53.96
Qwen-32B	TC	71.20	72.40	66.40	68.80	36.28	49.15	24.60	27.80	<u>49.62</u>	54.54
	TC+FM (L3.1)	63.40	61.40	66.40	68.00	28.79	38.74	23.20	26.20	<u>45.45</u>	48.59
	TC+FM (Q2.5)	74.40	77.80	68.80	72.80	34.15	47.86	26.00	30.40	50.84	57.21
Qwen-72B	TC	71.60	74.20	68.80	76.80	40.04	56.67	27.40	35.00	<u>51.96</u>	60.67
	TC+FM (L3.1)	67.40	69.80	69.60	76.00	33.98	47.33	23.20	29.60	48.54	55.68
	TC+FM (Q2.5)	74.20	80.60	75.20	79.20	41.38	58.59	27.40	34.60	54.55	63.25

725 Zihan Zhang, Meng Fang, and Ling Chen. Retrievalqa: Assessing adaptive retrieval-augmented
 726 generation for short-form open-domain question answering. In Lun-Wei Ku, Andre Martins, and
 727 Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024,*
 728 *Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 6963–6975. Association for
 729 Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.415. URL <https://doi.org/10.18653/v1/2024.findings-acl.415>.

731 Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling
 732 Yang, Wentao Zhang, and Bin Cui. Retrieval-augmented generation for ai-generated content:
 733 A survey. *CoRR*, abs/2402.19473, 2024. doi: 10.48550/ARXIV.2402.19473. URL <https://doi.org/10.48550/arXiv.2402.19473>.

A THE USE OF LARGE LANGUAGE MODELS (LLMs)

739 During the writing process of this paper, we utilized the large language models (LLMs) as an auxiliary
 740 tool, solely for polishing the grammar and expression of the paper to enhance its standardization and
 741 readability. The research conception, core content, experimental design, and conclusions of the paper
 742 were all independently completed by the authors, with the large language models not participating in
 743 any substantive aspects of the research or creative process.

B ADDITIONAL EXPERIMENTAL RESULTS

747 As shown in Table 5, teacher architecture has a decisive impact on student adaptation under the
 748 TC+FM framework: across all scales (3B–72B), Qwen2.5-72B consistently surpasses Llama3.1-70B
 749 in both ACC_E and ACC_L. This pattern demonstrates that architectural alignment is the key of
 750 effective cognitive transfer in multi-hop reasoning.

C TOOLS DESIGN

754 To enable efficient and accurate retrieval-driven reasoning, we adopt the ReAct framework as
 755 implemented in Qwen-Agent and develop two complementary tools. The first is a **Parallel Search**
 756 **Tool C.1** built on the Google Search API that performs parallel retrievals and makes retrieval decisions,

756 Table 6: Additional results on four multi-hop QA benchmarks.
757

758 Model	759 Method	760 2WikiMQA		761 Bamboogle		762 Frames		763 MuSiQue		764 AVG	
		765 ACC _E	766 ACC _L	767 ACC _E	768 ACC _L	769 ACC _E	770 ACC _L	771 ACC _E	772 ACC _L	773 ACC _E	774 ACC _L
775 Qwen-14B	776 TC	777 64.40	778 64.60	779 64.00	780 68.00	781 34.22	782 44.66	783 20.40	784 23.20	785 45.76	786 50.12
	787 TC+FM (3B)	788 68.40	789 70.00	790 58.40	791 64.00	792 32.89	793 43.33	794 22.80	795 27.00	796 45.62	797 51.08
	798 TC+FM (7B)	799 69.00	800 71.20	801 63.20	802 72.00	803 33.37	804 46.84	805 20.60	806 24.40	807 46.54	808 53.61
	809 TC+FM (14B)	810 71.60	811 75.20	812 70.40	813 72.80	814 34.83	815 49.51	816 24.00	817 28.20	818 50.21	819 56.43
	820 TC+FM (32B)	821 71.80	822 74.40	823 64.80	824 72.00	825 35.25	826 46.24	827 22.20	828 27.20	829 48.51	830 54.96
	831 TC+FM (72B)	832 70.00	833 71.60	834 64.00	835 68.00	836 35.92	837 48.06	838 25.80	839 28.20	840 48.93	841 53.96
775 Qwen-32B	776 TC	777 71.20	778 72.40	779 66.40	780 68.80	781 36.28	782 49.15	783 24.60	784 27.80	785 49.62	786 54.54
	787 TC+FM (3B)	788 70.40	789 70.00	790 60.80	791 67.20	792 32.77	793 44.53	794 22.60	795 24.80	796 46.64	797 51.63
	798 TC+FM (7B)	799 71.80	800 73.20	801 61.60	802 67.80	803 34.83	804 45.14	805 22.90	806 25.80	807 47.28	808 53.49
	809 TC+FM (14B)	810 74.60	811 76.40	812 70.40	813 75.20	814 35.19	815 47.33	816 23.20	817 29.80	818 50.85	819 57.18
	820 TC+FM (32B)	821 69.60	822 70.00	823 64.00	824 73.60	825 31.10	826 42.89	827 21.00	828 24.60	829 46.43	830 52.77
	831 TC+FM (72B)	832 74.40	833 77.80	834 68.80	835 72.80	836 37.15	837 53.86	838 26.00	839 30.40	840 51.59	841 59.71
775 Qwen-72B	776 TC	777 71.60	778 74.20	779 68.80	780 76.80	781 40.04	782 56.67	783 27.40	784 35.00	785 51.96	786 60.67
	787 TC+FM (3B)	788 71.20	789 75.20	790 65.60	791 72.00	792 38.23	793 53.64	794 24.60	795 29.40	796 49.91	797 57.56
	798 TC+FM (7B)	799 72.80	800 76.80	801 71.20	802 76.00	803 37.13	804 55.70	805 24.60	806 30.40	807 51.93	808 59.73
	809 TC+FM (14B)	810 71.00	811 73.40	812 74.40	813 77.60	814 35.56	815 50.12	816 23.60	817 30.00	818 51.14	819 57.78
	820 TC+FM (72B)	821 74.20	822 80.60	823 75.20	824 79.20	825 41.38	826 58.59	827 28.40	828 36.60	829 54.80	830 63.75

775 Table 7: Ablation study of enhanced ToolCall across different models.
776

777 Method	778 Model	779 2WikiMQA		780 Bamboogle		781 Frames		782 MuSiQue		783 Avg	
		784 ACC _E	785 ACC _L	786 ACC _E	787 ACC _L	788 ACC _E	789 ACC _L	790 ACC _E	791 ACC _L	792 ACC _E	793 ACC _L
784 ToolCall	785 Qwen-3B	786 54.20	787 42.40	788 37.60	789 40.00	790 17.96	791 22.94	792 13.00	793 12.60	794 30.69	795 29.49
	796 Qwen-7B	797 56.80	798 56.20	799 49.60	800 54.40	801 23.78	802 32.28	803 16.40	804 18.80	805 36.65	806 40.42
	807 Qwen-14B	808 64.40	809 64.60	810 64.00	811 68.00	812 34.22	813 44.66	814 20.40	815 23.20	816 45.76	817 50.11
	818 Qwen-32B	819 71.20	820 72.40	821 66.40	822 68.80	823 36.28	824 49.15	825 24.60	826 27.80	827 49.62	828 54.54
	829 Qwen-72B	830 71.60	831 74.20	832 68.80	833 76.80	834 40.04	835 56.67	836 27.40	837 35.00	838 51.96	839 60.67
	840 Llama3.1-8B	841 51.20	842 51.40	843 55.20	844 57.60	845 24.51	846 37.38	847 17.40	848 20.80	849 37.08	850 41.80

785 enabling high-efficiency searching. The second is **FutureMind Tool C.2**, a module instantiated with
786 a larger language model that strengthens structured reasoning and retrieval logic, enabling more
787 precise, targeted retrieval. Together, these components decouple the complex retrieval process from
788 the reasoning logic, enabling efficient and accurate retrieval execution, thereby facilitating better
789 evidence integration and yielding more accurate final answers.

790 C.1 PARALLEL SEARCH TOOL

791 Parallel Search Tool (Google) Introduction	
Name:	Parallel_Search (google)
Description:	You should invoke the Parallel_Search Tool (google) whenever the user's query falls into one of the following categories:
1.	Your internal knowledge base and training data are insufficient to answer the question accurately.
2.	The user asks about a specific example, product, or piece of information that you can retrieve in greater detail via the web.
3.	The question involves the latest data, dynamic information, or any knowledge that postdates your training cutoff and requires real-time updates.
4.	The answer exists in external knowledge sources you cannot directly access; you must search to retrieve it.
5.	Although you possess general knowledge of the topic, an online search would yield more detailed or up-to-date information (e.g. current buzzwords or trending topics).

```

810
811 6. You encounter an unfamiliar term or concept and must avoid fabrication by verifying it
812  through the search tool.
813 7. You need to consult a product manual or official specification to support your response.
814
815
816  The search tool supports both parallel and sequential queries:
817 1. If multiple searches are independent, you may issue them in parallel.
818 2. If queries depend on each other (i.e. require ordered steps), perform them sequentially.
819
820 Parameters:
821 {
822     "type": "object",
823     "properties": {
824         "queries": {
825             "type": "array",
826             "items": { "type": "string" },
827             "description": (
828                 "List of search keywords:  
"- Parallel search: supply multiple keywords at once;  
"- Iterative search: supply a single-element array."
829             ),
830             "examples": [
831                 {
832                     "queries": [
833                         "Xiaomi SU7 Ultra official price",
834                         "Tesla Model S latest price"
835                     ]
836                 },
837                 {
838                     "queries": ["Mishi wolffin fish namer"]
839                 }
840             ],
841             "required": ["queries"]
842         }
843     }
844

```

C.2 FUTUREMIND TOOL

FutureMind Tool Instruction

Name: FutureMind

Description:

Upon receiving a query, the FutureMind Tool is invoked first to obtain a **systematic thinking pattern and solution roadmap** for the problem.

For retrieval-oriented questions:

1. It produces a structured problem-solving workflow and retrieval strategy.
2. It explicitly delineates the logical chain from "Problem Definition" through "Condition Decomposition" to "Conclusion Derivation."
3. It provides executable search sequences and combined query conditions as retrieval guidance, thereby enhancing information acquisition efficiency and ensuring retrieval accuracy.

Parameters:

```
{

```

```

864
865     "type": "object",
866     "properties": {
867         "query": {
868             "type": "string",
869             "description": "A query that requires systematic
870             problem analysis and retrieval-strategy formulation"
871         }
872     },
873     "required": ["query"]
874
875
876
877
```

D TOOLCALL TEMPLATE

Toolcall Template

Tools

You may call one or more functions to assist with the user query.

You are provided with function signatures within `<tools></tools>` XML tags:

```

885
886 <tools>
887 {tool_descs}
888 </tools>
```

For each function call, return a json object with function name and arguments within `<tool_call></tool_call>` XML tags:

```

891
892 <tool_call>
893 {"name": <function-name>, "arguments": <args-json-object>}
894 </tool_call>
```

E FUTUREMIND: DESCRIPTION AND IMPLEMENTATION DETAILS

E.1 OVERVIEW OF FUTUREMIND

As depicted in Figure 2, FutureMind is a lightweight, training-free reasoning framework that transfers systematic thinking patterns from teacher models to smaller student models via **adaptive thinking-pattern distillation**. It decomposes tasks into four staged modules—Problem Analysis, Logical Reasoning, Strategy Planning, and Retrieval Guidance—and emits concise, auditable plans that specify whether to retrieve, what to retrieve, and how to integrate evidence. To reduce retrieval overhead, the framework supports three composable paradigms (Figure 4): Forward Stepwise Reasoning, Backward Constraint Focusing, and Parallel Intersection Reasoning. Overall, FutureMind enables resource-constrained models to perform structured, low-latency retrieval and reasoning with improved accuracy and interpretability.

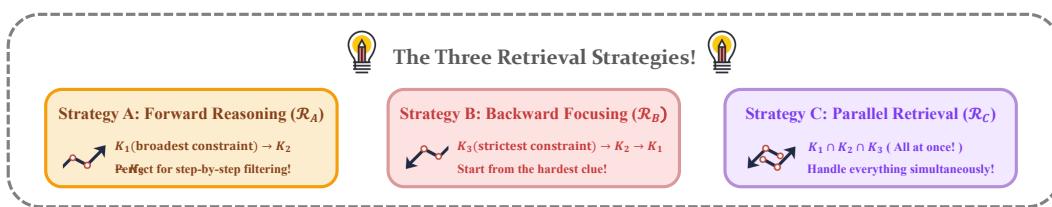


Figure 4: Three adaptive retrieval paradigms employed in FutureMind.

918 E.2 INSTRUCTION FOR ENHANCED TOOLCALL USING PARALLEL SEARCH TOOL C.1
919

920 Instruction for Enhanced ToolCall Using Parallel Search Tool

921

922 **System:**
923 You are a helpful assistant.
924

925 **Tools:**
926 You may call one or more functions to assist with the user query. You are provided with
927 function signatures within <tools></tools> XML tags:
928

```
929 <tools>
930   {
931     "type": "function",
932     "function": {
933       "name": "google_search",
934       "description": "Parallel Search Tool of
935                     description",
936       "parameters": "Parallel Search Tool of
937                     parameters",
938     }
939   }
940 </tools>}
```

941 For each function call, return a json object with function name and arguments within
942 <tools></tools> XML tags:
943

```
944 <tool_call>
945   {"name": <function-name>, "arguments": <args-json-object>}
946 </tool_call>
```

947 **User:**
948 **Question: {user question}**
949 You FIRST think about the reasoning process as an internal monologue and then provide the
950 final answer. The reasoning process MUST BE enclosed within <think> </think> tags. The
951 final answer MUST BE put in <answer> </answer> tags.
952

953 E.3 INSTRUCTION FOR TOOLCALL-DRIVEN FUTUREMIND USING PARALLEL SEARCH
954 TOOL C.1
955

956 Instruction for ToolCall-driven FutureMind Using Parallel Search Tool

957

958 **System:**
959 You are a helpful assistant.
960

961 **Tools:**
962 You may call one or more functions to assist with the user query.
963 You are provided with function signatures within <tools></tools> XML tags:
964

```
965 <tools>
966   {
967     "type": "function",
968     "function": {
969       "name": "Parallel Search",
970       "description": "Parallel Search Tool of
971                     description",
```

```

972
973             "parameters": "Parallel Search Tool of
974                 parameters",
975             }
976         }
977     </tools>
978
979     <tools>
980         {
981             "type": "function",
982             "function":
983                 {
984                     "name": "futuremind",
985                     "description": "Futuremind Tool of
986                         description",
987                     "parameters": "Futuremind Tool of parameters",
988                 }
989         }
990     </tools>
991
992     For each function call, return a json object with function name and arguments within
993     <tools></tools> XML tags:
994
995     <tool_call>
996         {
997             "name": <function-name>,
998             "arguments": <args-json-object>
999         }
1000     </tool_call>
1001
1002     user:
1003     Question: {user question}
1004     You FIRST think about the reasoning process as an internal monologue and then provide the
1005     final answer. The reasoning process MUST BE enclosed within <think> </think> tags. The
1006     final answer MUST BE put in <answer> </answer> tags. First, invoke the "futuremind" tool
1007     to obtain a systematic thinking strategy and solution roadmap for the given Question (original
1008     question text verbatim). After that, you may call the search tool multiple times as needed to
1009     gather or verify information until you have sufficient material to answer. Once all necessary
1010     information is confirmed, provide the final answer using concise, focused language without
1011     unnecessary elaboration.
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

```

1010 E.4 RELATED METHODS: ENHANCED TOOLCALL (TC) AND TOOLCALL-DRIVEN 1011 FUTUREMIND (TC+FM)

1012 **Enhanced ToolCall (TC).** Enhanced ToolCall (TC) is implemented on top of the ReAct framework,
1013 following the Toolcall Template D. To support efficient information access, we employ the **Parallel**
1014 **Search Tool** C.1, which enables parallel retrieval while constraining the reasoning process to a
1015 maximum of 10 steps. This design ensures both efficiency and controllability. The reasoning process
1016 follows the predefined instructions provided in E.2.

1017
1018
1019
1020
1021 **ToolCall-driven FutureMind (TC+FM).** ToolCall-driven FutureMind (TC+FM) extends this
1022 design by incorporating **FutureMind (FM) Method** into the enhanced ToolCall framework. For
1023 more efficient and robust aggregation of retrieved evidence, we employ the **Parallel Search Tool** C.1
1024 and **FutureMindTool** C.2, which enables parallel retrieval while constraining the reasoning process
1025 to a maximum of 10 steps. This design further improve retrieval robustness while maintaining
controllable reasoning. The reasoning process follows the predefined instructions provided in E.3.

1026 E.5 MODULE OF FUTUREMIND: PROBLEM ANALYSIS-LOGICAL REASONING-STRATEGY
 1027 PLANNING-RETRIEVAL GUIDANCE
 1028

1029 In this section, we first introduce the instructions utilized by each module, followed by a detailed
 1030 walkthrough using a concrete example. Specifically, we employ the query “A painter whose father
 1031 died of heart disease, who has an elder sister, and who had five children with his wife before their
 1032 marriage broke down. Later, he experienced three subsequent relationships. A literary work was later
 1033 written based on this person. What is the title of that work?” to illustrate the step-by-step execution
 1034 of each component within the FutureMind framework.

1035
 1036 E.5.1 PROBLEM ANALYSIS MODULE
 1037

1038 Problem Analysis Module Instructions
 1039

1040 **Name of Module:** Problem Analysis
 1041

1042 **Instruction:** Please first identify the core conflict, intrinsic attributes, and target outcomes of
 1043 the problem; extract key dimensions $C_1, C_2, C_3, C_4, \dots$ (e.g., causal relationships, constraints,
 1044 objective functions).

1045 Problem Analysis Module Execution Example
 1046

1047 **1. Core Objectives**

1048 Precisely identify the painter who satisfies the specified biographical constraints and report
 1049 the single literary work (title and author) that is documented as based on that painter, together
 1050 with concise supporting evidence.

1051 **2. Intrinsic Attributes**

1052 Verifiable, atomic facts to check: (a) profession = painter; (b) father’s cause of death = heart
 1053 disease; (c) exactly one sister; (d) five children with his wife; (e) the marriage later dissolved;
 1054 (f) subsequently experienced three distinct significant romantic relationships; (g) existence of
 1055 a notable literary work explicitly linked to this person.

1056 **3. Target Outcomes**

1057 Output (1) the literary work’s title and author; (2) a compact evidence map (3 bullet points)
 1058 showing which of the above intrinsic attributes (by label) are corroborated by authoritative
 1059 sources linking the painter to the work.

1060 **4. Critical Conditions**

1061 Let the four critical conditions be defined as:

- 1062 • **C_1 (Identity / Profession):** "Painter" — restricts the candidate set to artists/painters.
- 1063 • **C_2 (Family / Life Events):** Father died of heart disease; has one sister; had five
 1064 children with his wife; the marriage later dissolved. These are distinctive and
 1065 searchable biographical facts.
- 1066 • **C_3 (Romantic History):** Later experienced three significant romantic relationships
 1067 (multiple partners), with possible implications of relocation or cross-cultural life
 1068 (common in comparable cases, though not explicitly stated).
- 1069 • **C_4 (Literary Adaptation Clue):** A literary work was written based on this person —
 1070 the target is the work’s title (and its author), and the work is presumably well-known.

1073
 1074 E.5.2 LOGICAL REASONING MODULE
 1075

1076 Logical Reasoning Module Instructions
 1077

1078 **Name of Module:** Logical Reasoning
 1079

1079 **Instruction:** Apply first principles to reverse-engineer the underlying logic, determine critical

1080
 1081 conditional elements (denoted **K₁, K₂, K₃, K₄**), and order them by importance or logical
 1082 sequence.
 1083
 1084

1085 Logical Reasoning Module Execution Example

1086
 1087 Use first-principles decomposition to organize the constraints and decide the most efficient
 1088 solving sequence.

- 1089 • **K₁**: Had five children with his wife and the marriage ultimately broke down. Very
 1090 specific life events that exclude most painters.
- 1091 • **K₂**: Subsequently had multiple (approximately three) significant romantic relation-
 1092 ships. Together with **K₁** this narrows the candidate set substantially.
- 1093 • **K₃**: Father died of heart disease (additional temporal/place details increase discrim-
 1094 inative power). Serves as auxiliary confirmation.
- 1095 • **K₄**: Explicitly identified as the prototype or clear inspiration for a notable literary
 1096 work — provides the final verification link.

1097
 1098 First search for individuals satisfying **K₁**; verify **K₂** and **K₃** in parallel; finally confirm the
 1099 linkage to a notable literary work (**K₄**) to identify the unique match.

1100 1101 E.5.3 STRATEGY PLANNING MODULE

1102 Strategy Planning Module Instructions

1103
 1104 **Name of Module:** Strategy Planning

1105
 1106 **Instruction:** Thinking Strategies (Select one or more as appropriate).

1107
 1108 **Strategy A: Forward Stepwise Reasoning (Progressive Filtering)**

- 1109 • Sequentially filter from basic to decisive conditions:
 - 1110 a. Retrieve the set A meeting base condition **K₁**;
 - 1111 b. From A, filter the subset B meeting secondary condition **K₂**;
 - 1112 c. From B, filter the subset C meeting key condition **K₃**;
 - 1113 d. Within C, verify candidates satisfying decisive condition **K₄**.

1114
 1115 **Strategy B: Backward Constraint Focusing (Reverse Narrowing)**

- 1116 • Reverse-derive from stringent constraints to base compatibility:
 - 1117 a. Prioritize retrieving set A satisfying the strictest constraint (e.g., **K₃**) to narrow scope
 1118 quickly;
 - 1119 b. From A, filter set B meeting critical feature condition (e.g., **K₄**);
 - 1120 c. From B, verify set C satisfying prerequisite condition (e.g., **K₂**);
 - 1121 d. Within C, confirm final candidates compatible with baseline condition (e.g., **K₁**).

1122
 1123 **Strategy C: Parallel Intersection Reasoning (Parallel Filtering)**

- 1124 • Treat conditions as independent dimensions in parallel:
 - 1125 a. Retrieve sets A, B, C, D each satisfying conditions **K₁, K₂, K₃, K₄** respectively;
 - 1126 b. Compute intersection ($A \cap B \cap C \cap D$) to extract solutions meeting all conditions
 1127 simultaneously.

1128
 1129 **Choose the strategy (or combination) based on problem characteristics, data
 1130 availability, and efficiency requirements.**

1134
1135

Strategy Planning Module Execution Example

1136

Selected Strategy: Strategy B — Backward Constraint Focusing (recommended)

1137

Rationale: The “literary adaptation” condition (K_4) is the strongest and most selective constraint. Since only a small number of painters are explicitly used as prototypes for notable literary works, starting from this constraint yields the fastest and most discriminative narrowing of the search space.

1140

Strategy B: Backward Constraint Focusing (Reverse Narrowing)

1141

• Reverse-derive from the most stringent constraint (K_4):

1142

- a. Begin with K_4 : retrieve or enumerate well-known literary works explicitly based on, or inspired by, real painters.
- b. For each identified literary work, extract the painters documented or claimed as prototypes.
- c. For each candidate painter, verify biographical compatibility with K_1 (profession), K_2 (family/life events), and K_3 (romantic history).
- d. Remove any candidate failing one or more conditions; if no candidate remains, return to step (a) and expand the set of literary works.
- e. The painter who satisfies K_1 – K_3 and matches K_4 constitutes the final solution.

1143

Apply Strategy B exclusively. Start from the strongest constraint (literary adaptation), identify candidate works and their prototypes, then filter by K_1 – K_3 to obtain the unique match.

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

E.5.4 RETRIEVAL GUIDANCE MODULE

1157

Retrieval Guidance Module Instructions

1158

Name of Module: Retrieval Guidance

1159

Instruction: Thinking Strategies (Select one or more as appropriate).

1160

Keyword Guidance

1161

Derive core entities, attributes, and relationships from the problem analysis (e.g., “AI + ethical risks + regulatory policy”).

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

Retrieval Guidance Module Execution Example

1. Keyword Design (Hierarchical)

- Level 1 (Directly locate works/prototypes): “novel based on painter”, “fiction inspired by painter”

1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226

- Level 2 (Combine with biographical details): “painter five children divorced wife”, “father died of heart disease painter”, “painter with sister”
- Level 3 (Verify person and work combinations): “<Work Title> prototype”, “<Painter Name> depicted as <Work Title>”

2. Resource Selection (Priority)

- Authoritative biographies and academic publications (artist biographies, art history books)
- Literary studies and annotations (research articles on novel prototypes, author memoirs)
- Recognized encyclopedias and databases (e.g., Wikipedia for preliminary screening, with professional biographies as final reference)
- Cultural and historical journals, author studies, and book reviews (to verify claims of works based on real persons)

3. Retrieval Sequence (Recommended)

- First retrieve well-known novels or works based on painters to generate candidate works meeting the key literary adaptation condition
- Examine annotations, author notes, and academic reviews of these works to identify the associated prototype painters
- Retrieve biographies of candidate painters to verify biographical details such as number of children, marital status, father’s cause of death, and romantic history
- Cross-validate with independent authoritative sources for final confirmation

4. Query Examples (Facilitating Iterative Filtering)

- “novel inspired by painter” OR “fiction based on painter”
- “painter AND five children AND divorced wife”
- “<Work Title> prototype <Painter Name>” (used to verify known candidates)

5. Screening Criteria (Preliminary Inclusion / Exclusion)

- Inclusion: Reliable sources such as biographies, academic papers, authoritative book reviews, and author statements; multiple independent sources citing the same prototype and facts
- Exclusion: Unverified blogs or secondhand claims without sources; circular references lacking original evidence; candidates with biographical details contradicting key facts (e.g., children count, marital status)

F INSTRUCTION TEMPLATES

F.1 INSTRUCTIONS FOR EVALUATION

In this work, we use LLM-as-Judges (GPT-4o-mini) to evaluate multi-hop question answering (QA) benchmarks: 2WikiMultihopQA, Bamboolle, MuSiQue, and FRAMES. The specific instructions are as follows.

Instruction for Judge

Given a Question and its Golden Answer, verify whether the Predicted Answer is correct. The prediction is correct if it fully aligns with the meaning and key information of the Golden Answer. Respond with True if the prediction is correct and False otherwise.

[Question:] {user question}
[Golden Answer] {reference answer}

1242

1243

[Predicted Answer] {assistant's answer}

1244

1245

1246

1247

F.2 INSTRUCTION FOR NAIVE GENERATION

1248

1249

Instruction for Naive Generation

1250

1251

Please answer the below questions. You should think step by step to solve it. The final answer MUST BE put in <answer> </answer> tags.

1252

1253

[Question:] {user question}

1254

1255

1256

1257

F.3 INSTRUCTION FOR STANDARD RAG

1258

1259

Instruction for Standard RAG

1260

1261

You are a knowledgeable assistant that utilizes the provided documents to answer the user's question accurately.

1262

1263

Guidelines:

- Analyze the provided documents to extract relevant information. Synthesize the information to formulate a coherent and accurate answer.
- Ensure that your response directly addresses the user's question using the information from the documents.

[Question:] {user question}

[Documents:] {documents}

1264

1265

1266

1267

1268

1269

1270

1271

F.4 INSTRUCTION FOR SEARCH-O1

1272

1273

1274

1275

1276

Instruction for Search-o1

1277

You are a reasoning assistant with the ability to perform web searches to help you answer the user's question accurately. You have special tools:

To perform a search: write <begin_search_query> your query here <end_search_query>. Then, the system will search and analyze relevant web pages, then provide you with helpful information in the format <begin_search_result> ...search results... <end_search_result>. You can repeat the search process multiple times if necessary. The maximum number of search attempts is limited to {MAX_SEARCH_LIMIT}.

Once you have all the information you need, continue your reasoning.

Example:

Question: “...”

Assistant thinking steps:

- I might need to look up details about ...

Assistant:

<begin_search_query>...<end_search_query>

(System returns processed information from relevant web pages)

Assistant continues reasoning with the new information...

Remember:

- Use <begin_search_query> to request a web search and end with <end_search_query>.
- When done searching, continue your reasoning.

1296 **G FORMULA DESCRIPTION**
12971298 **G.1 FORMULA EXPLANATION OF ACC_E**
12991300 **Formula:**

1301
$$ACC_E = \begin{cases} 1 & \exists g \in G, \text{norm}(pred) \supseteq \text{norm}(g) \\ 0 & \text{otherwise} \end{cases}$$

1302
1303

1304 where $pred$ is the predicted answer; $G = \{g_1, \dots, g_k\}$ is the gold answer set; $\text{norm}(\cdot)$ denotes a
1305 normalization procedure that handles paraphrasing, punctuation, and related surface variations; and
1306 the relation $A \supseteq B$ indicates that A fully covers the core semantics of B . Hence, $ACC_E = 1$ when
1307 the normalized prediction semantically covers at least one normalized gold answer, and 0 otherwise.
13081309 **H OVERALL TOKEN CONSUMPTION AND API COST OF GPT-4O-MINI AND**
1310 **GOOGLE SEARCH API**
13111312 **H.1 COST ANALYSIS OF LARGE MODEL EVALUATION**
13131314 For the GPT-4o-mini model, the total number of tokens consumed during the scoring process can be
1315 divided into three main components: the initial scoring prompt F.1, the question, the golden predicted
1316 answer, and the GPT-4o-mini model’s output (True/False).
13171318 We statistically analyzed the input token usage across different datasets, as shown in Table 8. Across
1319 the four datasets (2WikiMQA, Bamboogle, Frames, and MuSiQue), the total input token count
1320 amounts to approximately 772,000 tokens, while the total output token count is around 2,000 tokens.
13211322 Given the official pricing of GPT-4o-mini — \$0.15 per million input tokens and \$0.60 per million
1323 output tokens — the total API cost for one complete experimental setting across all datasets is
1324 estimated to be approximately \$0.117. In total, we conducted 84 such experimental runs, resulting in
an overall evaluation cost of approximately \$9.828 for the GPT-4o-mini scoring process.
13251326 Table 8: Overall token consumption of GPT-4o-mini across different datasets in a single experimental
1327 setting. All values are reported in units of *10,000 tokens (w)*.
1328

1329 Dataset	Tokens Consumed (w)
1330 2WikiMQA	16.5 w
1331 Bamboogle	3.7 w
1332 Frames	38.0 w
1333 MuSiQue	19.0 w
1334 Total	77.2 w

1336 **H.2 COST ANALYSIS OF SEARCH TOOL INVOCATION**
13371338 This cost estimation focuses exclusively on the direct expenses incurred by the *Enhanced ToolCall*
1339 method and the *ToolCall-driven FutureMind* method. For both approaches, the majority of the cost
1340 originates from API calls made by the *parallel Search Tool*. According to the official pricing of the
1341 Custom Google Search JSON API, the cost is set at \$5 per 1,000 queries.
13421343 Across the four datasets (2WikiMQA, Bamboogle, Frames, and MuSiQue), both the Enhanced ToolCall
1344 and ToolCall-driven FutureMind methods perform approximately 4.2k search engine queries on
1345 average. Therefore, the estimated cost for executing all experiments across the four datasets with
1346 either method is roughly \$21. Considering the total number of experimental runs, the cumulative
1347 search-related expenditure is estimated to be approximately \$1,365.
13481349 In addition, we provide a concrete example to illustrate this estimation. In this case, the student
model is *Qwen2.5-3B*, and the teacher models are *Qwen2.5-72B*, *Qwen2.5-32B*, *Qwen2.5-14B*, and
Qwen2.5-7B, respectively. The detailed statistics are presented in Table 9, which reports the total
1350

1350
1351 Table 9: Total number of search queries across datasets for each teacher–student model pair under
1352 Baseline and FutureMind settings.
1353

Dataset	Qwen-72B → 3B		Qwen-32B → 3B		Qwen-14B → 3B		Qwen-7B → 3B	
	TC	TC+FM	TC	TC+FM	TC	TC+FM	TC	TC+FM
Wiki	1032	1297	1028	1120	987	1322	1005	1259
Bamboogle	208	235	203	184	196	286	202	214
Frames	2030	1923	2000	1210	1693	2031	2016	1492
Musique	1056	1217	1032	821	1021	821	1040	997
Total	4326	4672	4263	3335	3897	4460	4263	3962

1362
1363 number of search queries across datasets for each teacher–student model pair under both the *Enhanced*
1364 *ToolCall*(TC) and *ToolCall*-driven FutureMind(TC+FM) settings.
1365

1366
1367 **I DISCUSSION: FUTUREMIND’S FLEXIBILITY IN CORRECTING THE LOGICAL**
1368 **PATH**
1369

1370 This section discusses how **FutureMind** exhibits flexibility in correcting and refining the logical
1371 reasoning path. This capability represents one of the core design goals of FutureMind: to enable
1372 dynamically adaptive reasoning that balances structured planning with real-time flexibility.
1373

1374 Rather than functioning as a one-shot, fixed-plan generator, FutureMind is designed as a **dynamically**
1375 **callable tool** (clarified in Appendix E.3) that can be invoked iteratively as needed. This design
1376 preserves structured planning capabilities while maintaining flexibility in reasoning, thereby enabling
1377 more adaptive inference strategies. Its core mechanisms include:

- 1378 • On-demand triggering. FutureMind can be called on demand when the student model
1379 detects issues such as an excessively large candidate space, high retrieval noise, or invalid
1380 intermediate states. In these situations, FutureMind provides refined or alternative reasoning
1381 strategies to guide subsequent exploration.
- 1382 • Strategy diversity with decentralized control. FutureMind provides dynamic and diverse
1383 reasoning strategies, while the ultimate selection and adjustment of the reasoning path are
1384 made by the student model based on real-time retrieval results and available computational
1385 budget, preserving flexibility.

1386 Overall, this interaction forms a closed adaptive loop of **execution bottleneck** → **on-demand**
1387 **invocation** → **strategy optimization**, mirroring the iterative problem-solving logic of human experts.
1388

1389 **Illustrative Example.** The following case demonstrates this mechanism through a knowledge-
1390 intensive reasoning example: Who is the poet that was a friend of the author of One Hundred Years
1391 of Solitude and also won a Nobel Prize?
1392

- 1393 • **First FutureMind call.** During the initial exploration, the model recognized that retrieving
1394 "friends" could lead to a large number of redundant candidates. It therefore made the first
1395 FutureMind call and obtained a **forward stepwise reasoning strategy**: Identify the author
1396 → Retrieve well-known literary associates → Filter for poets → Verify Nobel laureateship.
- 1397 • **Second FutureMind call.** During execution, even after narrowing the set of "associates",
1398 dozens of candidates remained, indicating high downstream reasoning difficulty. The
1399 model triggered a second FutureMind call and received a recommended **backward con-**
1400 **straint-focusing strategy**: List Nobel Prize-winning poets → Filter candidates with docu-
1401 mented associations to García Márquez → Confirm friendship via credible sources.
1402
- 1403 • After receiving this guidance, the model refined the second step to "filter candidates with
publicly verifiable interaction records with García Márquez" and proceeded. It first identified

1404
 1405 **Pablo Neruda** (1971 Nobel Prize in Literature, poet) as the core candidate and confirmed
 1406 their long-term friendship via Márquez's essay **García Márquez on Pablo Neruda**. Verifi-
 1407 cation through literary archives ultimately yielded the correct answer **Neruda**.
 1408

1409 **J CASE STUDY**
 1410

1411 We present a case study to illustrate the full pipeline of the proposed FutureMind method 2. The
 1412 case is drawn from the MuSiQue benchmark, which poses a question that the Qwen2.5-7B-Instruct
 1413 model alone can hardly answer correctly. We then demonstrate how the proposed ToolCall-driven
 1414 FutureMind employs Qwen2.5-14B-Instruct as a teacher model to guide Qwen2.5-7B-Instruct in
 1415 solving the problem, and provide the full reasoning trajectory leading to the final answer.
 1416

1417 Case Trajectory in MuSiQue
 1418

1419 **Question:** When was the Palau de la Generalitat built in the city in which died the man
 1420 called Martin of the region of Spain where Berdejo is located?
 1421

1422 **Step 1**
 1423

1424 **<think>**
 1425 I'm not fully familiar with this question, so I'll first call the FutureMind tool to get a
 1426 systematic thinking pattern and roadmap.
 1427 **</think>**

1428 **<tool_call>**
 1429 {"name": "FutureMind", "arguments": { "query": ["When was the Palau de la Generalitat
 1430 built in the city in which died the man called Martin of the region of Spain where Berdejo is
 1431 located?"]}}
 1432 **</tool_call>**

1433 **<tool_response>**
 1434 1. **Problem Analysis** - Identify the construction date of the Palau de la Generalitat in the city
 1435 where Martin of the region of Spain, where Berdejo is located, died.
 1436

1437 2. **Logical Reasoning** - Key conditional elements
 1438 K1 (correct identification of Martin and his region in Spain)
 1439 K2 (connection between Martin and the city of death)
 1440 K3 (location of Berdejo in Spain)
 1441 K4 (construction date of the Palau de la Generalitat in the identified city)

1442 3. **Strategy Planning** - Selected strategy A (Forward Stepwise Reasoning)
 1443 a. Retrieve information on Martin of the region in Spain where Berdejo is located (K1)
 1444 b. From the information, identify the city where Martin died (K2)
 1445 c. Verify the connection between Berdejo and the identified region in Spain (K3)
 1446 d. Within the identified city, find the construction date of the Palau de la Generalitat (K4)

1447 4. **Retrieval Guidance**
 1448 **Keywords:** Martin of [region in Spain], Berdejo, Palau de la Generalitat, construction
 1449 date
 1450 **Resources:** Wikipedia, Encyclopedia Britannica, Spanish historical archives, local
 1451 government websites of the identified region and city
 1452 **Sequence:** Historical biographies → regional information → local historical records
 1453 → architectural or government websites
 1454 **Query:** "Martin of [region in Spain] biography"; "Berdejo location in Spain"; "Palau
 1455 de la Generalitat construction date" in the identified city
 1456 **Screening:** Exclude non-relevant biographies or historical records; limit to sources
 1457 with verifiable information on the construction date.

```

1458
1459 </tool_response>
1460
1461 Step 2
1462 <think>
1463 To answer the question, the goal is to identify the construction date of the Palau de la
1464 Generalitat in the city where Martin, from the region in Spain where Berdejo is located, died.
1465 The key steps are: 1. First, gather information about Martin and the region where Berdejo is
1466 located. This will allow the identification of the city where Martin died. 2. Then, validate the
1467 connection between Berdejo's location and the identified region in Spain. 3. After identifying
1468 the city where Martin died, find the construction date of the Palau de la Generalitat in that
1469 city. The selected strategy is Forward Stepwise Reasoning, and the reasoning involves
1470 retrieving historical and regional data followed by architectural information for the Palau de
1471 la Generalitat. The next step is to search for relevant data using the following queries.
1472 </think>
1473
1474 <tool_call>
1475 {"name": "google_search", "arguments": {"query": ["Martin of [region in Spain] biography",
1476 "Berdejo location in Spain", "Palau de la Generalitat construction date"]}}
1477 </tool_call>
1478 <tool_response>
1479 Martin (born 1356, Gerona, Catalonia [Spain]—died May 31, 1410, Barcelona)
1480 was the king of Aragon from 1395 and of Sicily (as Martin II from 1409)
1481 (https://www.britannica.com/biography/Martin-king-of-Aragon-and-Sicily)
1482 From March 2 to 14, 2024, the Palau de la Generalitat de Valencia opens its
1483 doors and invites you to explore this historic building that houses the Presidency
1484 (https://www.visitvalencia.com/en/events-valencia/palau-de-la-generalitat-opens-its-doors-celebrate-fallas)
1485 ...
1486 </tool_response>
1487 Step 3
1488 <think>
1489 The search results reveal the following: 1. Several individuals named Martin exist in different
1490 regions of Spain, but Martin I of Aragon (1392–1409) seems the most relevant. However,
1491 further information about his death location is needed. 2. Berdejo is a town in the Zaragoza
1492 province of Aragon, Spain. 3. The Palau de la Generalitat in Barcelona dates back to the 15th
1493 century. Given the lack of a direct link between Martin I and Berdejo, we will refine the
1494 search to focus on identifying the city where Martin I died, and its potential connection to
1495 Berdejo. The next search query will be: Martin I of Aragon death place.
1496 </think>
1497
1498 <tool_call>
1499 {"name": "google_search", "arguments": {"query": ["Martin I of Aragon death place"]}}
1500 </tool_call>
1501 <tool_response>
1502 Matino is the son of King Martin I of Aragon. In the 1390, Matino married the Sicilian
1503 Queen and cousin Maria and became co-king. After Matino's death in 1409, his father
1504 became King of Sicily, known as Matino (https://www.wikiwand.com/zh-cn/articles)
1505 ...
1506 </tool_response>
1507
1508 Step 4
1509 <think>
1510 Martin I of Aragon died in Barcelona on May 31, 1410. The question asks for the construction
1511

```

1512

1513 date of the Palau de la Generalitat in Barcelona, where Martin died, and relates it to Berdejo,
1514 located in Zaragoza, Aragon. Since Berdejo is in a different region (Aragon) and the Palau de
1515 la Generalitat is in Barcelona (Catalonia), we need to focus on the construction of the Palau
1516 de la Generalitat in Barcelona. This was completed in the 15th century, and since Martin I
1517 died in 1410, the building was already constructed by then.

1518 </think>

1519

1520 <answer>The Palau de la Generalitat in Barcelona, Spain, was built in the 15th century, long
1521 before Martin I of Aragon died in 1410. <answer> ✓

1522

1523

1524

1525

1526

1527

1528

1529

1530

1531

1532

1533

1534

1535

1536

1537

1538

1539

1540

1541

1542

1543

1544

1545

1546

1547

1548

1549

1550

1551

1552

1553

1554

1555

1556

1557

1558

1559

1560

1561

1562

1563

1564

1565