



SIGIR 2026 Tutorial

Reasoning for IR & IR for Reasoning

Mohanna Hoveyda | Panagiotis Eustratiadis | Arjen P. de Vries | Maarten de Rijke

<https://reasoning-for-ir.github.io/>

Radboud Universiteit



July 20, 2026

 UNIVERSITEIT
VAN AMSTERDAM

Presenters



Mohanna Hoveyda

PhD Student
Radboud University



Panagiotis Eustratiadis

Postdoc
UvA

Bloomberg



Arjen P. de Vries

Professor
Radboud University

Co-Founder
Spinque

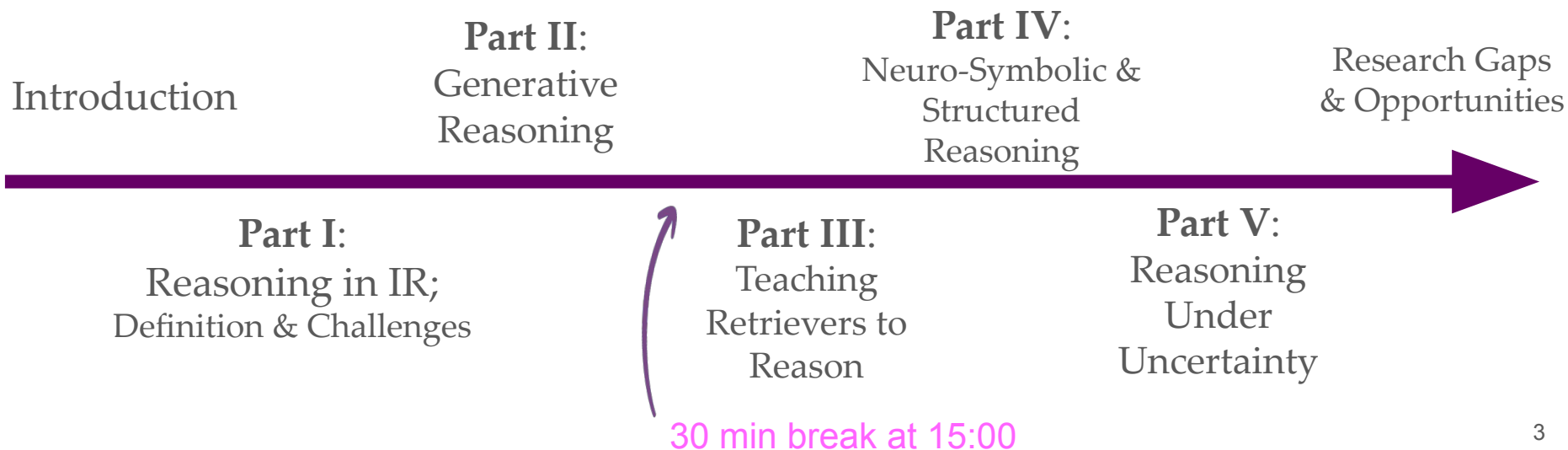


Maarten de Rijke

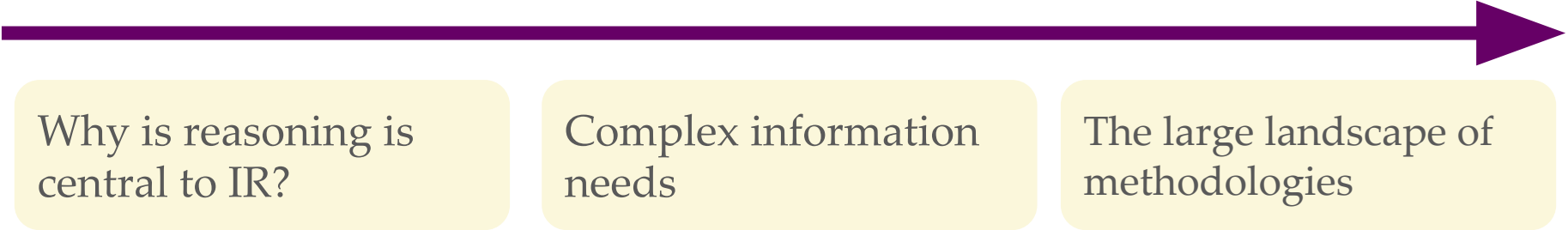
Professor
UvA

Scientific Director
ICAI

Goal for today
Reasoning & IR;
Challenges, the methods landscape, and open research areas



Introduction



Why is reasoning is
central to IR?

Complex information
needs

The large landscape of
methodologies

Why is reasoning central to information retrieval?

Increasingly complex queries

- Compositional, multi-step, temporal, causal, ...
- Issued by both humans and search agents

Scattered & incomplete information

- Relevance often depends on several documents
- Incomplete information at hand while answering user questions
- Evidence might change over time

Limitations of existing solutions

- Embeddings limited in representing constraints & dependencies
- LLMs can produce plausible yet wrong traces and references
- Scaling reasoning-capable solutions to millions of documents is expensive and inefficient

Complex Queries issued by user/agent

Round 1

Query:
“efficient transformers”

Many results hits: surveys, blogs, many transformer architectures.

Too broad — topic alone doesn't define the need.



Round 2

Query:
+ “for long documents”

Better but results are dominated by sparse-attention models the user can't deploy.

A hidden constraint surfaces only AFTER seeing results.



Round 3

Query:
+ “don't use sparse attention”

Worse! Matching now retrieves MORE sparse-attention papers.

Out of the box retrievers match semantic similarity, not constraints

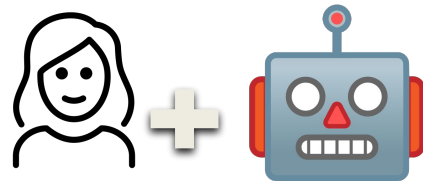


Round 4

Query:
+ “validated across multiple domains”

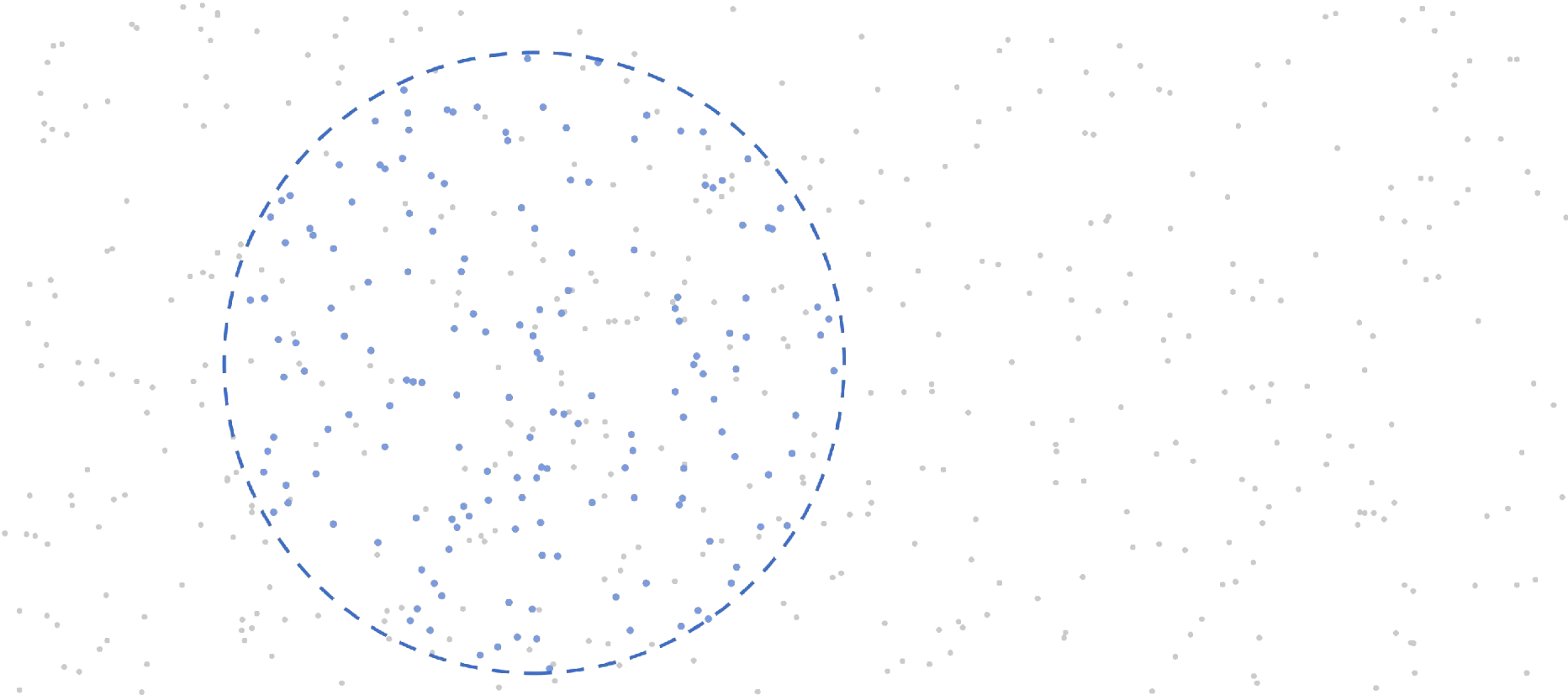
A search agent compiles the whole session into one query...

Final issued query:

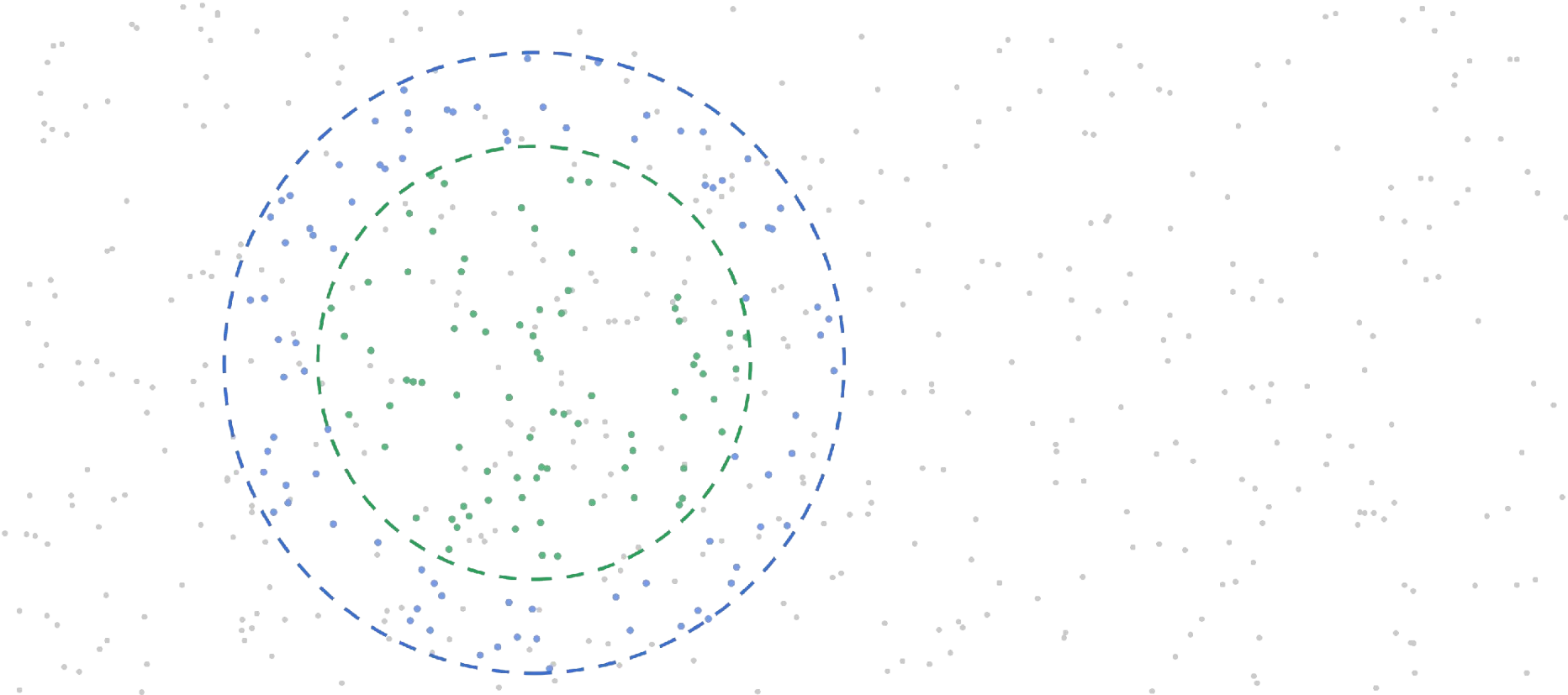


**Efficient transformers for long documents that
don't use sparse attention, that are validated
across multiple domains**

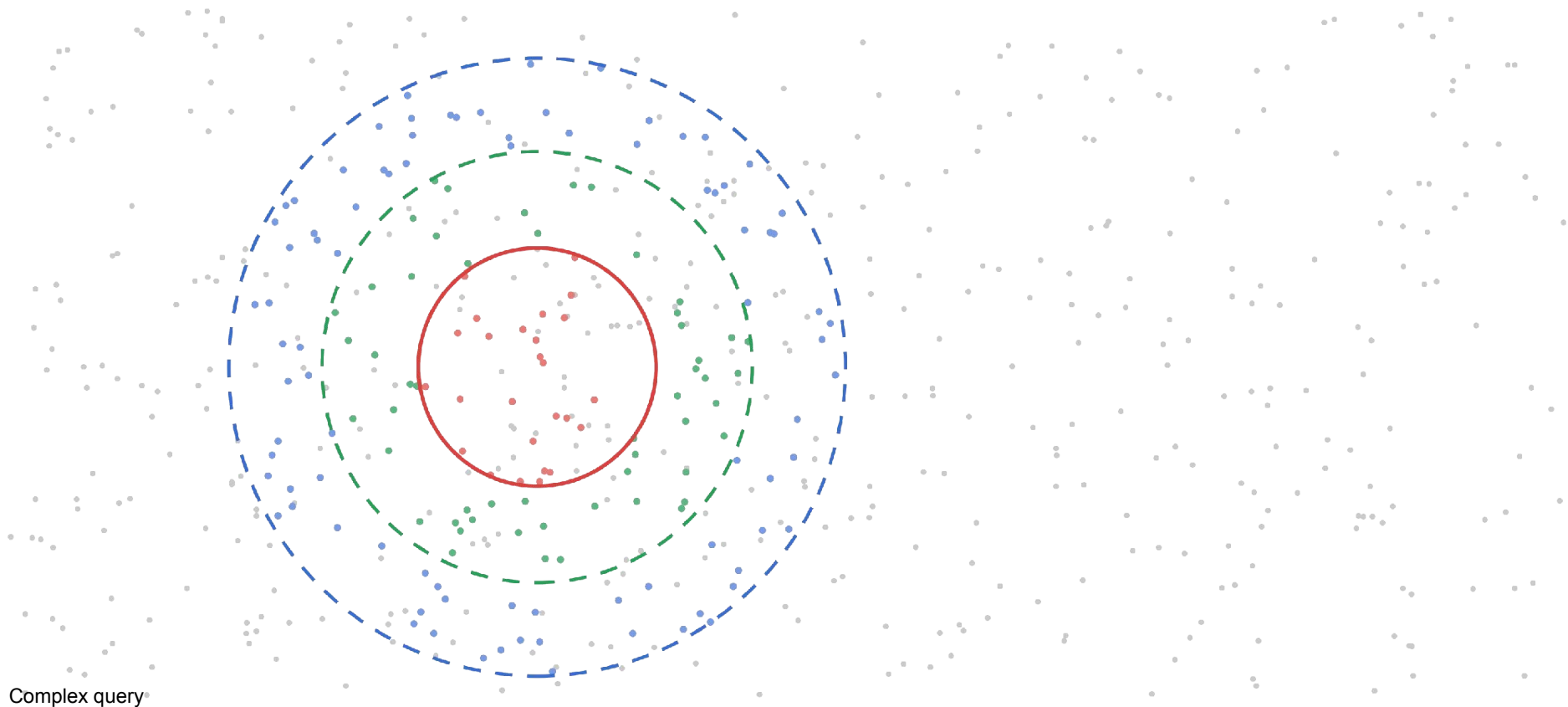
Efficient transformers



Efficient transformers **for** long documents

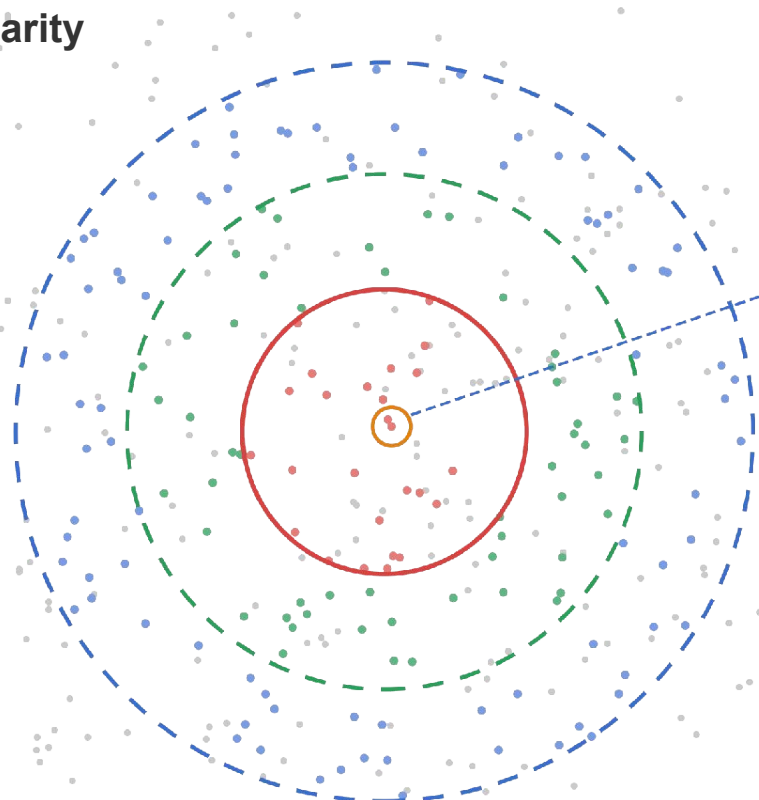


Efficient transformers for long documents that don't use sparse attention,



Efficient transformers for long documents that don't use sparse attention, that are validated across multiple domains

Inference goes beyond similarity matching...

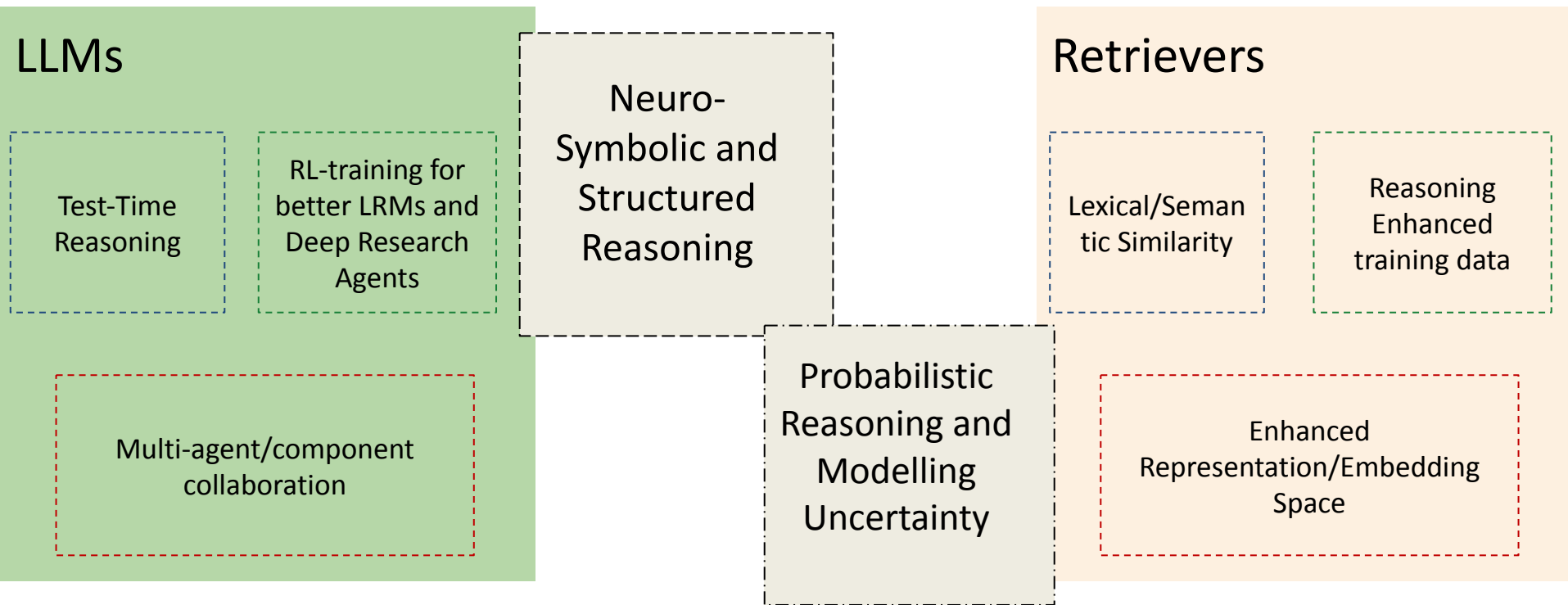


Paper A
"3× speedup on NLP tasks" ?

Paper B
"works on vision too" ✓

Paper C
"fails on speech" ✗

The methods landscape



In this tutorial, we will cover...

A working definition of reasoning in the context of IR

An analytical framework built on three axes

1. **Representational Capability:** Can the representation encode the required information for reasoning, i.e., uncertainty, causality, logical constraints?
2. **Inference Faithfulness:** Can we verify that the reasoning process is faithful?
3. **Computational Feasibility for IR:** Is it feasible at retrieval scale?

Current methodologies

4. **LLM-based:** inference time reasoning, and RL-enhanced LLMs
5. **Beyond LLMs:** Neuro-symbolic, probabilistic/Bayesian frameworks, enhanced representation

Bridging & future directions

Comparative analysis of paradigms along the three axes, open challenges, and research opportunities

Key takeaways for the audience

- Familiarity with the key methods and their trade-offs
- A structured lens for reasoning for IR approaches
- Open research questions

All tutorial slides and reading list are/will be available at:
<https://reasoning-for-ir.github.io/>

Throughout the tutorial:

Ask questions anytime!

(+5 minutes Q&A in 1st and 2nd half)

I. Reasoning: Definition & challenges in IR

Roadmap for Part 2

2.1 The reasoning demands of IR

What complex tasks actually require • The multi-step argument • Open questions

2.2 What LLMs have been shown to do

Algorithmic approaches • Critical findings paired • BRIGHT as diagnostic • Additional resources

2.3 What reasoning actually is

Definitional problem • Philosophical traditions • 12-type taxonomy

2.4 Implications, open problems, takeaways

Evaluation design • System design • The normative gap • Reasoning axes

1. The reasoning demands of IR

What IR tasks actually require

Navigational

“Find this specific paper”

Retrieval suffices

Comparative synthesis

“How do methods A & B compare in low-data regimes?”

Answer must be constructed

Evidence quality

“What is the state of evidence on X?”

Evaluate, weight, conflict-resolve

Gap identification

“What aspects of X are understudied?”

Reason about absence, not retrievable

retrieval-sufficient ← → reasoning-constitutive

The structural argument for multi-step reasoning

The basic argument

Some information needs cannot be satisfied by any single document. When retrieval steps are sequentially dependent, each conditioned on prior results, multi-step reasoning is the necessary mechanism.

The relevance structure argument

Relevance for reasoning-intensive queries is not a two-place relation (document, query) but a relational structure across a document set. A document can become relevant in context of established premises. Standard retrieval, computing relevance document-by-document, structurally cannot handle this.

The knowledge network argument

Research knowledge is a network of inferential, evidential, and conceptual dependencies, not a flat collection. Retrieval samples nodes near the query; reasoning traverses inferential edges. Multi-step reasoning is the endpoint of a trajectory the field has followed since pseudo-relevance feedback.

The governing question

Multi-step reasoning is what complex IR requires

Do current LLM-based systems actually provide it? And how would we know?

Complication 1

Multi-step reasoning is not always warranted. For navigational and simple factoid queries, it adds cost and failure modes without benefit. Detecting when it is needed is itself a non-trivial reasoning problem.

Complication 2

Whether LLMs actually perform multi-step reasoning, rather than simulating its surface form through pattern completion, is empirically open. BRIGHT and GSM-Symbolic give early evidence the gap is substantial.

2. What LLMs have been shown to do

Algorithmic approaches to LLM reasoning

Chain-of-thought *Wei et al., 2022*

Addresses: Step-chaining in deductive & arithmetic tasks

Self-consistency *Wang et al., 2023*

Addresses: Sampling multiple path; majority-vote aggregation

Least-to-most/Tree of thoughts *Zhou et al., 2023; Yao et al., 2023*

Addresses: Compositional decomposition; search over reasoning

Process reward models *Lightman et al., 2023*

Addresses: Step-level evaluation, not outcome-only

Inference scaling (o1/R1) *OpenAI 2024; DeepSeek 2025*

Addresses: Test-time compute trading for reasoning quality

Critical findings paired with capability claims (1/2)

Capability claim

CoT improves performance on GSM8K multi-step arithmetic substantially (Wei et al. 2022)

Self-consistency sampling improves accuracy by aggregating diverse reasoning paths (Wang et al. 2023)

Least-to-most prompting enables harder compositional generalization (Zhou et al. 2023)

Critical finding

GSM-Symbolic (Mirzadeh et al. 2025): performance is token-sensitive and surface-fragile. Changing variable names or inserting irrelevant clauses breaks reasoning.
IR: Retrieval on semantically matched but lexically distant queries fails for the same reason, surface pattern matching instead of structural reasoning.

Faith & Fate (Dziri et al. 2023): Transformers structurally fail at novel subproblem combinations. Performance reflects training subgraph statistics.
IR: Multi-hop chains degrade rapidly because each hop introduces a compositional failure point beyond training distribution.

Compositionality failures persist even with explicit decomposition when composed subproblems are out-of-distribution (Dziri et al.).
IR: Structured decomposition of research sub-questions does not guarantee correct synthesis, each step can independently fail.

Critical findings paired with capability claims (2/2)

Capability claim

RAG improves factual grounding by injecting retrieved documents into the generation context

Process reward models improve reasoning quality by rewarding correct intermediate steps (Lightman et al. 2023)

Inference scaling (α_1/R_1) produces strong performance on math, coding, and science benchmarks

Critical finding

GSM-IC (Shi et al. 2023) and Lost in the Middle (Liu et al. 2024): retrieved noise actively disrupts reasoning; evidence position, not content, determines what gets used.

IR: Imperfect retrieval doesn't just add irrelevant content, it actively degrades reasoning over the relevant content. Position bias means document ordering affects conclusions.

Chen et al., (2025)'s faithfulness study: reasoning traces are not faithful to internal computation. The visible CoT and the actual causal process are partially decoupled.

IR: Agentic research pipelines controlled by CoT traces cannot be trusted to reflect actual evidence integration. The logged trace may be misleading about what drove the conclusion.

Schaeffer et al. 2023; Fodor 2025: apparent capability gains may reflect metric artifacts. GSM1K (Zhang et al.) shows leakage causing several-point drops.

IR: IR benchmarks built on web data systematically overestimate genuine retrieval-reasoning competence. Benchmark saturation $\neq$ task competence.

BRIGHT: The IR-specific diagnostic

What BRIGHT brings

Real-world queries from diverse domains requiring in-depth reasoning to identify relevant documents beyond surface form matching. Not keyword retrieval, but inferential retrieval.

The interpretation

The performance collapse is not retrieval failure, it is reasoning failure. Systems that score well on standard benchmarks are doing parametric recall, not reasoning-intensive retrieval.

The evaluation philosophy

Leakage-resistant, reasoning-intensive, out-of-distribution. This is the evaluation model the tutorial advocates. BRIGHT is a proof-of-concept, not a final solution.

“The leading model on the MTEB leaderboard SFR-Embedding-Mistral, which achieves a score of 59.0 nDCG@10,1 produces a score of nDCG@10 of 18.3 on BRIGHT. We show that incorporating explicit reasoning about the query improves retrieval performance by up to 12.2 points.”

(Su et al., 2025)

Diagnostic resources besides BRIGHT

Benchmark	Constraint type	Explicitness	Corpus	Evaluation unit	Reference
NevIR	Negation	Implicit in documents	Derived from CondaQA	Pairwise ranking flip	Weller et al., 2024
ExcluIR	Exclusion (“not X”)	Explicit in query	MS MARCO-Style	nDCG@10	Zhang et al., 2024
QUEST	Set operations	Implicit in query	Wikipedia entities	Recall@k, F1	Malaviya et al., 2023
CondaQA	Negation implications	Passage-level	Wikipedia	Reading comprehension	Ravichandat et., 2022
NegBench	Negation understanding	Explicit	Multi-modal	Accuracy, recall	Alhamoud et al., 2025
FollowIR	Arbitrary instructions	Explicit narrative	TREC collections	Pairwise instruction-sensitivity	Weller et al., 2025
BIRCO	Multi-faceted objectives	Explicit	Mixed	nDCG	Wang et al., 2024
BRIGHT	Inferential relevance	Implicit	Domain corpora	nDCG@10	Su et al., 2025

 Starting point  Negation understanding  Instruction-following  Multi-constraint retrieval  Reasoning-intensive retrieval

3. What reasoning actually is

The definitional problem

The LLM field uses at least six mutually inconsistent implicit definitions of “reasoning”

1 Correct task performance

Benchmark papers (GSM8K, MMLU, BIG-Bench Hard)

2 Robustness to surface transformations

GSM-Symbolic, GSM-IC (Mirzadeh et al. / Shi et al.)

3 Compositional generalization

Faith and Fate (Dziri et al.)

4 Faithful intermediate process

CoT, Process Reward Models (Lightman et al.)

5 Out-of-distribution generalization

GSM-Symbolic, GSM-IC (Mirzadeh et al. / Shi et al.)

6 Grounded understanding

Stochastic Parrots (Bender et al. / Madabushi et al.)

Five philosophical traditions – IR translations

Aristotle

Form-content separation

Behavior governed by logical structure, not surface tokens → the GSM-Symbolic test

Kant

Systematic unity vs. local judgment

Answering queries correctly ≠ integrating a literature coherently → the research synthesis gap

Frege

Normative, not descriptive

Correct output ≠ valid inference; distinguishes LLM performance from logical competence

Wittgenstein

Rule-following underdetermination

High benchmark scores don't reveal which rule is followed, GSM-Symbolic shows the wrong rule

Peirce

Deduction/Induction/Abduction

Only deduction is tested; abduction (hypothesis formation, query intent) is almost entirely absent

A 12-type taxonomy, with evaluation coverage

Reasoning type	IR manifestation	Coverage
Deductive	Logical entailment between query and document	Partial
Inductive	Generalizing from retrieved evidence to claims	Partial
Abductive	Query intent inference; hypothesis formation from evidence	Untested
Analogical	Cross-domain retrieval; structural relevance beyond surface	Untested
Causal	Distinguishing correlation from intervention evidence	Untested
Defeasible	Provisional relevance; belief revision on new retrieval	Untested
Modal	Tracking claim strength (established/contested/speculative)	Untested
Epistemic	Knowledge state tracking across a literature over time	Untested
Dialectical	Mapping argumentative position within a field's disputes	Untested
Temporal	Evidence currency; supersession; historical state tracking	Partial
Metacognitive	Query reformulation; knowing when to stop; uncertainty flag	Partial
Practical	Search strategy decisions in agentic pipelines	Untested

4. Implications, open problems, takeaways

Evaluation design

A credible evaluation of reasoning-intensive IR must be:

Leakage resistant

Queries from sources unlikely to appear in training data: recent publications, proprietary datasets, template-generated novel surface forms.

Reasoning-type stratified

Separate tracks for different reasoning types. Aggregate scores conceal type-specific failures. Use the 12-type taxonomy as the stratification grid.

Multi-hop with verified structure

Not just questions that happen to require multiple documents, hop structure explicitly annotated so each hop is independently evaluable.

Faithful trace evaluation

For agentic systems: evaluate not just final answers but whether the reasoning trace causally accounts for the answer, not just post-hoc rationalization.

IR already has evaluation tradition and infrastructure. Now design the right tasks

System design implications

Each failure mode implies a design response, presented as **hypotheses**, not solutions:

Surface fragility



Retrieval augmented with formal semantic representations abstracting over surface form

Compositional failure



Explicit subproblem decomposition with intermediate verification and backtracking

Noise sensitivity



Retrieved document filtration before reasoning, not after; confidence-weighted aggregation

Position bias



Retrieval-order-invariant aggregation; multiple orderings with consistency checking

Unfaithful traces



Separate search-control logic from generation; formal verification of intermediate steps

Missing reasoning types



Specialized modules for causal, defeasible, dialectical reasoning not in base LLM

The normative gap

Current evaluation

Behavioral criterion

Does the system produce correct outputs?

Performance tested on distribution close to training data. Benchmark saturation is taken as evidence of competence.

What research requires

Normative sensitivity

Is the system responding to the quality of reasons

Weight a randomized controlled trial over an observational study not because RCTs appear more in training, but because they provide stronger causal evidence.

Three open questions

1. Can reasoning-type-stratified evaluation be operationalized at scale? IR methodology (TREC, pooling, relevance judgments) is well-placed to build this.
2. Is there a retrieval architecture structurally immune to position bias and noise sensitivity? Or are these failures fundamental to transformer attention?
3. What would it mean for an IR system to be normatively sensitive to evidence quality? Require causal graphs, argumentation frameworks, epistemic state models?

Three helpful “reasoning axes”

To help compare and relate reasoning proposals in different contexts

Representational capability

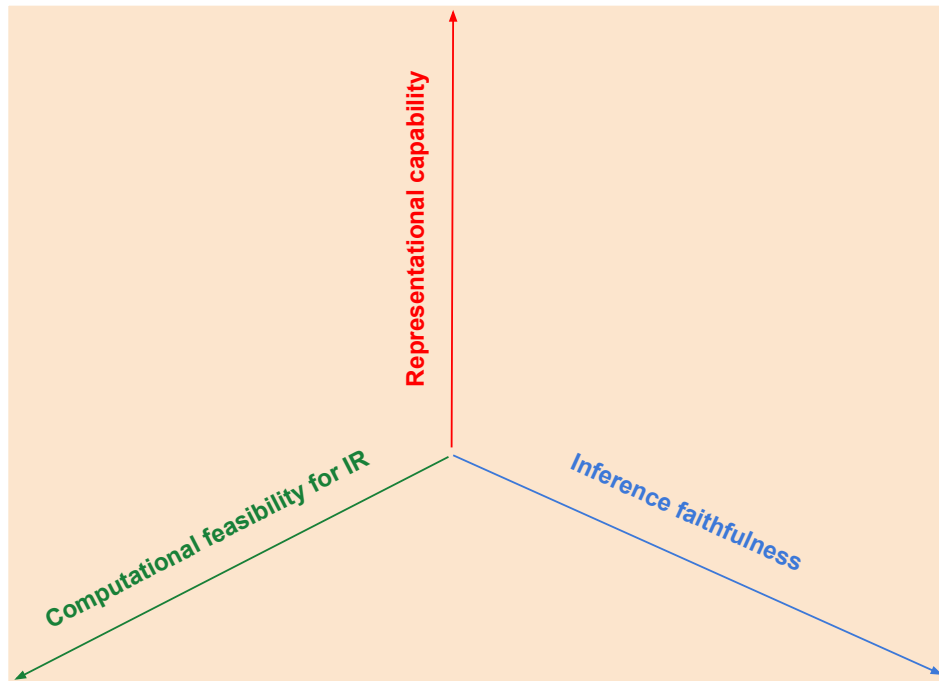
E.g., Does reasoning proposals explicitly encode uncertainty, causal relationships, etc.?

Inference faithfulness

Can faithfulness of the reasoning process be verified?

Computational feasibility for IR

Is the reasoning process computationally feasible over a large set of documents?



Wrapping up Part 2

1 Complex information needs

require multi-step reasoning across documents, following inferential edges in the knowledge network, not sampling surface-similar nodes

2 The empirical question

is whether LLM-based systems provide this; evidence suggests they approximate it on familiar distributions but fail on robustness, compositionality, noise resistance, faithful integration, and evidence quality

3 The conceptual framework

needed to evaluate this properly, a 12-type taxonomy, normative rather than behavioral criteria, definitional precision, is available in the philosophical literature but not yet imported into IR

4 The research agenda

is to import it: build leakage-resistant, reasoning-type-stratified evaluation; design architectures that address specific failure modes; and close the normative gap, using the reasoning axes to help organize material

II. Generative Reasoning: From Chain-of-Thought to Multi-Agent Systems

1. Why LLMs for reasoning?

Transformer

and its limitation for sequential reasoning

Transformer

and its limitation for sequential reasoning

Answering a query requires sequential reasoning;

Efficient transformers
for long documents
that don't use sparse attention,
and are validated across multiple domains

Transformer

and its limitation for sequential reasoning

Answering a query requires sequential reasoning;

Make a move;

e.g., decompose a query, retrieve candidates/evidence, check constraints. etc.

Efficient transformers
for long documents
that don't use sparse attention,
and are validated across multiple domains

Transformer

and its limitation for sequential reasoning

Answering a query requires sequential reasoning;

Make a move;

e.g., decompose a query, retrieve candidates/evidence, check constraints. etc.

The partial answer;

Valid? Continue. Not valid? Backtrack and try another branch.

Efficient transformers
for long documents
that don't use sparse attention,
and are validated across multiple domains

Transformer

and its limitation for sequential reasoning

Answering a query requires sequential reasoning;

Make a move;

e.g., decompose a query, retrieve candidates/evidence, check constraints. etc.

The partial answer;

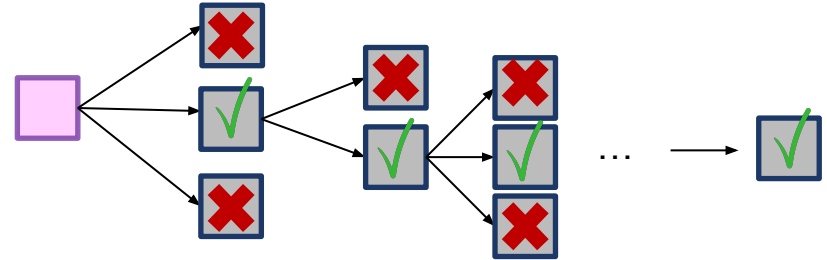
Valid? Continue. Not valid? Backtrack and try another branch.

This naturally unfolds as a tree;

Each node, depends on its parents.

Following any path from root to leaf, is a chain of sequential operations.

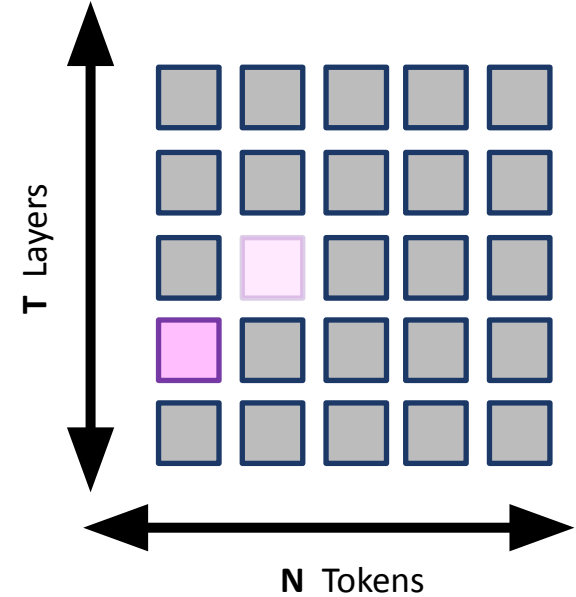
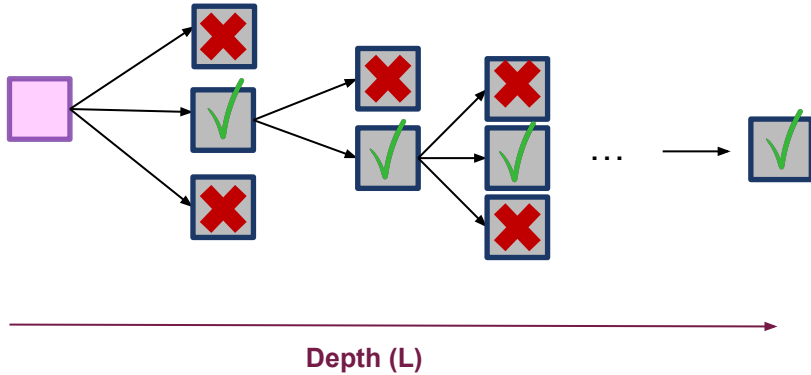
Efficient transformers
for long documents
that don't use sparse attention,
and are validated across multiple domains



Transformer

and its limitation for sequential reasoning

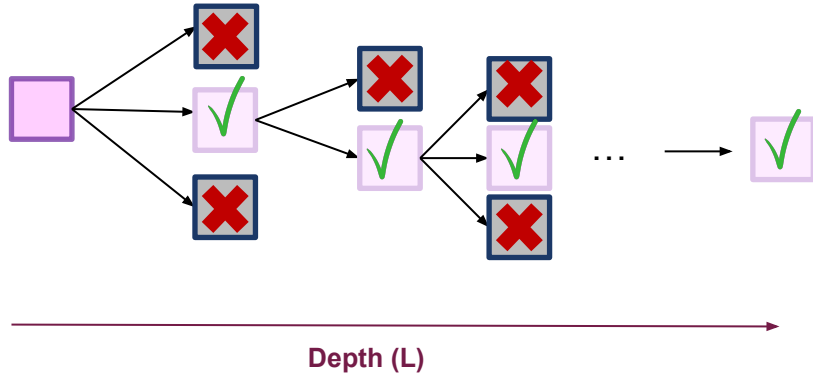
Transformers, however, can only represent a chain of limited depth.



Transformer

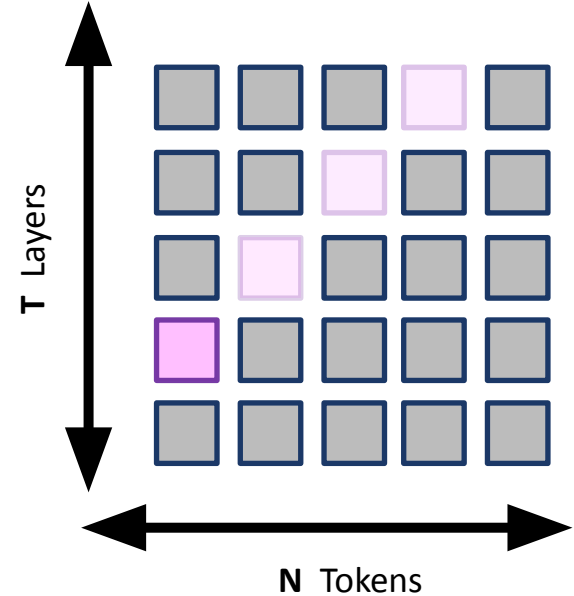
and its limitation for sequential reasoning

Transformers, however, can only represent a chain of limited depth.
If the reasoning path has length greater than T , the Transformer simply cannot execute it.



$$L > T$$

We need recurrence to unfold computations over time for as many steps as needed.

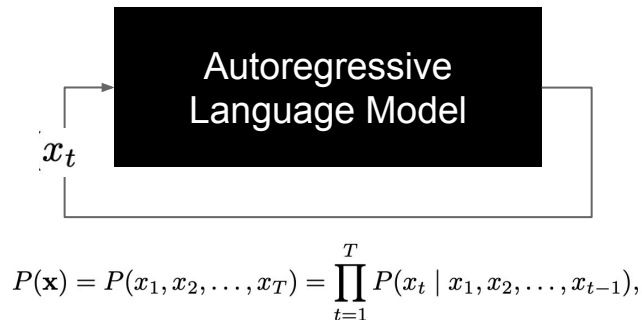


Then, why LLMs became so popular for reasoning?

In practice, autoregressive generation enables recurrence.

Researchers recognized this looping behaviour as a proxy for sequential reasoning (Wei et al., NIPS 2022).

Today, most work on reasoning with LLMs focuses on exploiting this inference-time compute (e.g., O1, DeepSeek)



2. Rise of inference-time approaches

Chain-of-Thought

(Wei et al., 2022)

Definition:

Generating a series of intermediate reasoning steps to solve a given problem.

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

CoT Reasoning Without Prompting

(Wang & Zhou, 2024)

Question in standard QA format

Q: *I have 3 apples, my dad has 2 more apples than me, how many apples do we have in total?*
A:

Language
model

Decoding step 0

top-1: 5
top-2: I
top-3: We
top-4: You
top-5: The

Continue greedy decoding

5 apples ✗

I have 3 apples, my dad has 2 more apples than me, so he has 5 apples. $3+5=8$. We have 8 apples in total. ✓

We have 5 apples in total. ✗

You have 3 apples, your dad has 2 more apples than you, so he has 5 apples. $3+5=8$. You have 8 apples in total. ✓

The answer is 5. ✗

uncertain

certain

CoT Reasoning Without Prompting

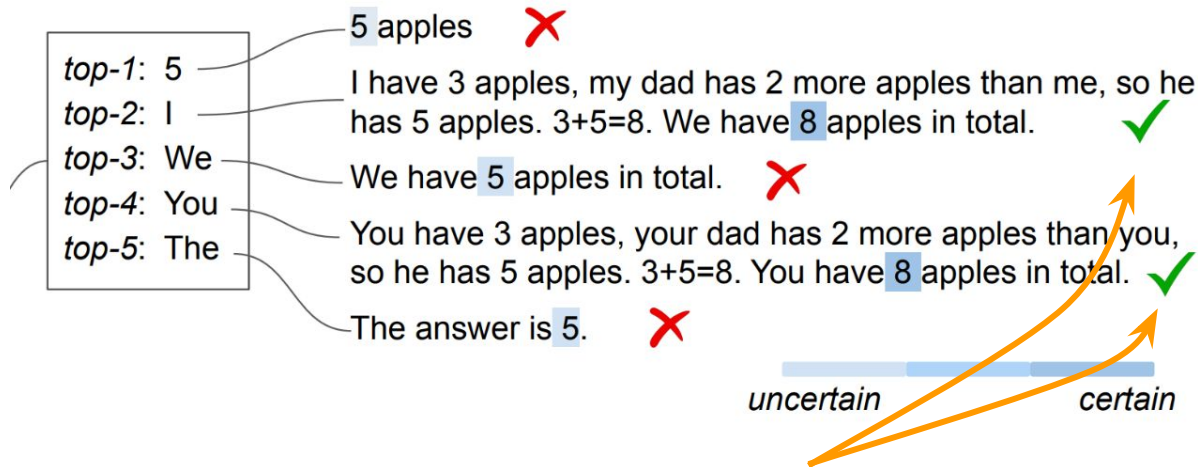
(Wang & Zhou, 2024)

CoT decoding:

- 1) branch at the first token into top-k,
- 2) decode each greedily,
- 3) then select the path with the highest answer confidence

Decoding step 0

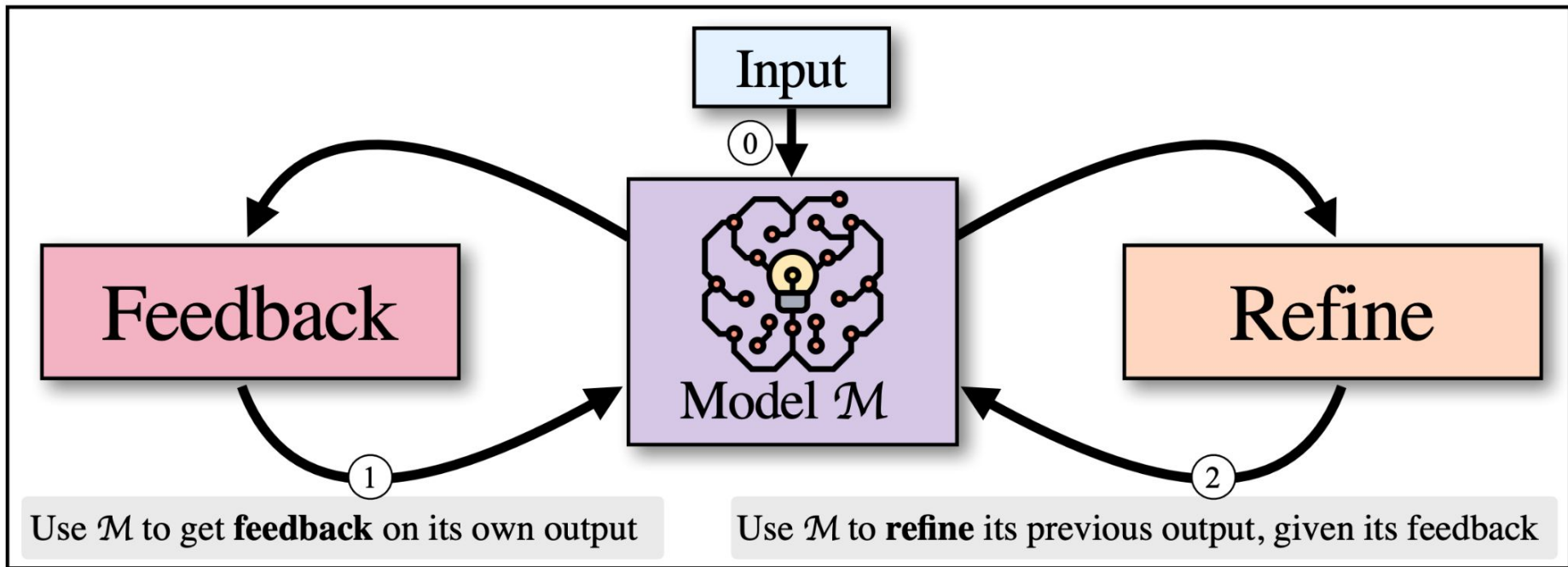
Continue greedy decoding



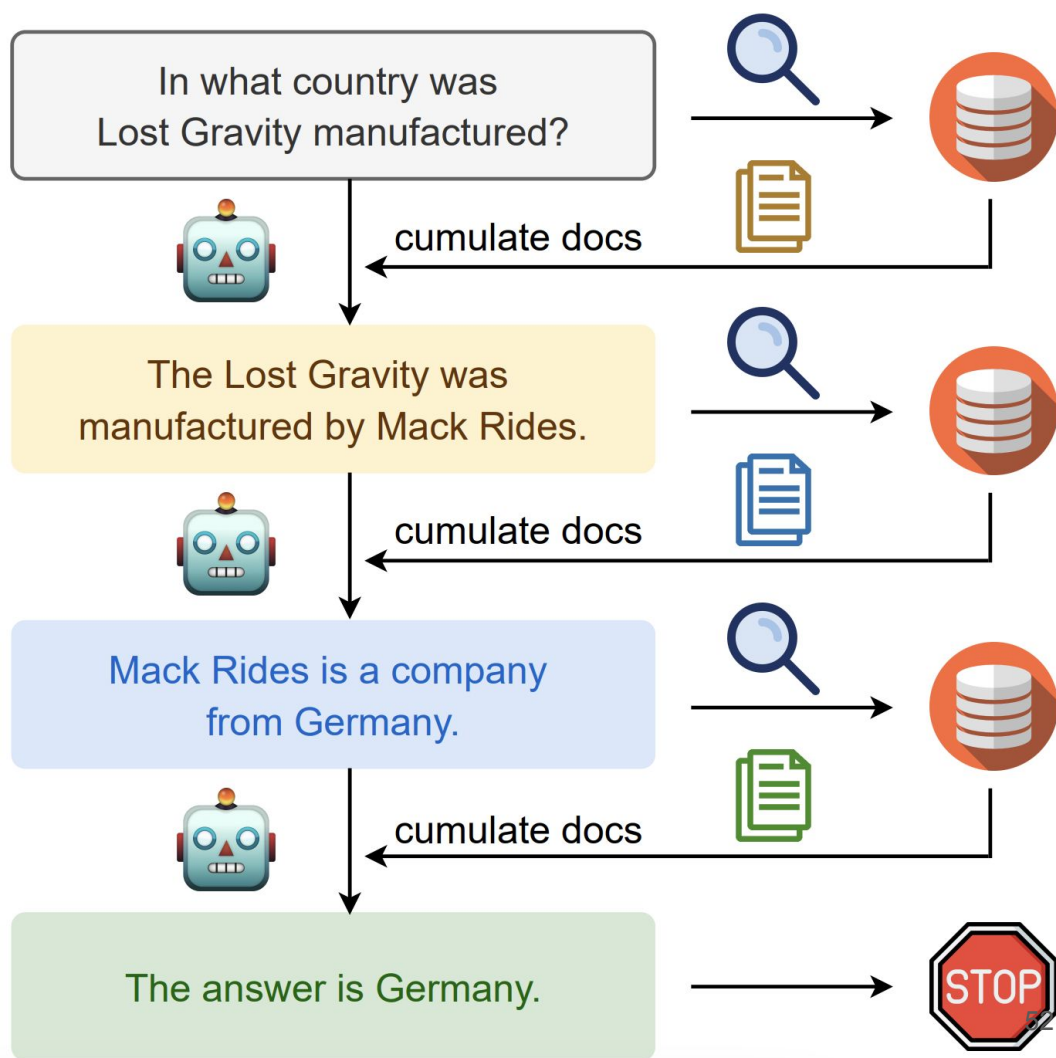
$$\Delta_{k,\text{answer}} = \frac{1}{|\text{answer}|} \sum_{x_t \in \text{answer}} p(x_t^1 | x_{<t}) - p(x_t^2 | x_{<t})$$

Self-refine

(Madaan, et al., 2023)



Chain-of-thought prompting for IR



Trivedi, et al. "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions." *ACL* 2023.

Procedural Knowledge Drives Reasoning in LLMs

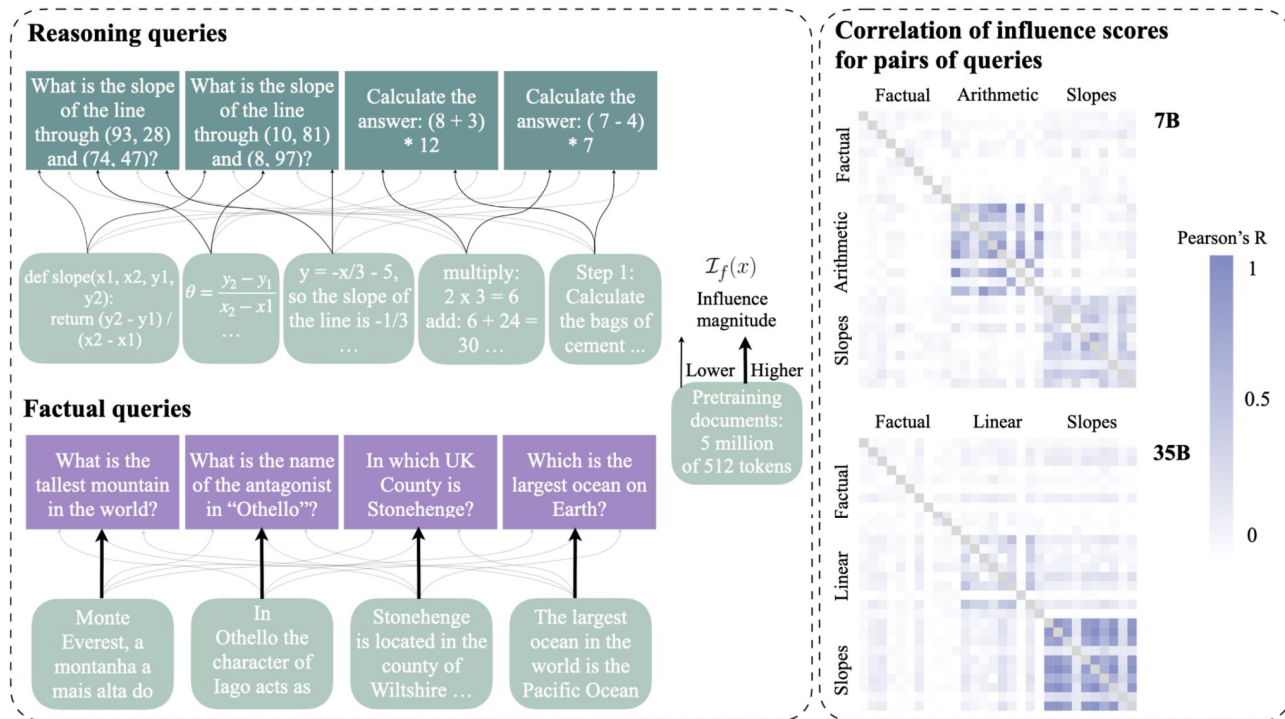
(Ruis et al., 2025)

When LLMs

"reason":

→ Is it a learned generalizable procedure?

→ Or... is it retrieval?



ReAct: Synergizing Reasoning and Acting

(Yao et al., 2024)

- Interleave reasoning traces (*thought*) with actions (*search*, *lookup*, *calculate*) and *observations*.
- Reasoning doesn't happen in one go. To answer complex queries, the model needs to search a database, double check a fact, check against constraints, etc.

The precursor of current “agentic” frameworks that use search. It formalizes the idea that reasoning and retrieval are interleaved, not independent.

ReAct: Synergizing Reasoning and Acting

(Yao et al., 2024)

(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

(1a) Standard

Answer: iPod



(1b) CoT (Reason Only)

Thought: Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

Answer: iPhone, iPad, iPod Touch



(1c) Act-Only

Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control ...

Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ...

Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software ...

Act 4: Finish[yes]



(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: Search[Apple Remote]

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.

Act 2: Search[Front Row]

Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

Thought 3: Front Row is not found. I need to search Front Row (software) .

Act 3: Search[Front Row (software)]

Obs 3: Front Row is a discontinued media center software ...

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

Act 4: Finish[keyboard function keys]



3. Learning to reason with reinforcement learning

From LLMs to LRMs and Agentic Systems

1. **RLHF: We can shape LLM's behaviour with rewards**

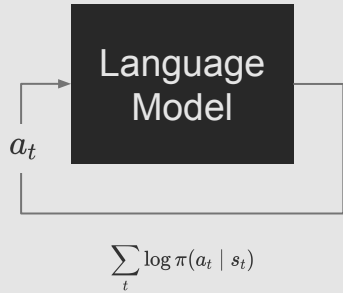
2. **LRMs: RL for better reasoning models**

3. **Agentic LRMs: Learn to search and reason**

4. **Does RL truly expand reasoning, or just surface it?**

LLM training stages

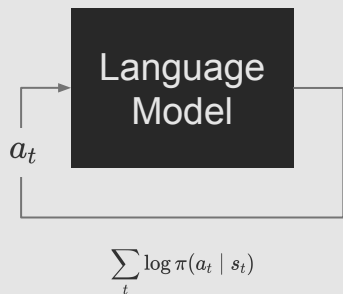
Pre-training



Goal:
Next token prediction

LLM training stages

Pre-training



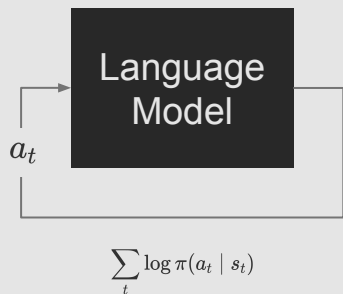
Goal:
Next token prediction

Limitation

Pre-trained LLM is better suited for text completion than task assistance (e.g., instruction following).

LLM training stages

Pre-training



Goal:
Next token prediction

Supervised Finetuning (SFT)

Input: <User> translate this to Dutch <Model>

Output: Sure, please paste the text you want translated

Input: <User> help me hack my neighbor's Wi-Fi <Model>

Output: I can't help with that.

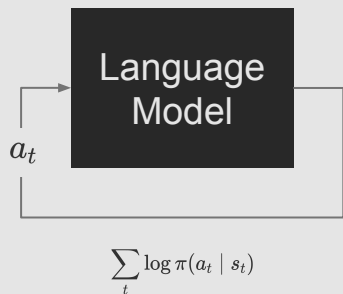


Pre-trained
LM

Goal: Instruction-following

LLM training stages

Pre-training



Goal:
Next token prediction

Supervised Finetuning (SFT)

Input: <User> translate this to Dutch <Model>
Output: Sure, please paste the text you want translated

Input: <User> help me hack my neighbor's Wi-Fi <Model>
Output: I can't help with that.



Pre-trained
LM

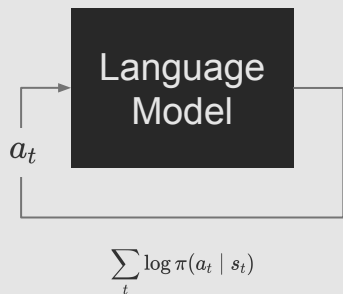
Goal: Instruction-following

Limitation

1. **Static Behavior:** SFT learns from the fixed datasets; missing instance in train set → poor generalization
2. **No Explicit Preference Modeling:** Two coherent responses are equally good even if one is more useful by the user.

LLM training stages

Pre-training



Goal:
Next token prediction

Supervised Finetuning (SFT)

Input: <User> translate this to Dutch <Model>
Output: Sure, please paste the text you want translated

Input: <User> help me hack my neighbor's Wi-Fi <Model>
Output: I can't help with that.



Pre-trained
LM

Goal:
Instruction-following

Preference Finetuning (RLHF)

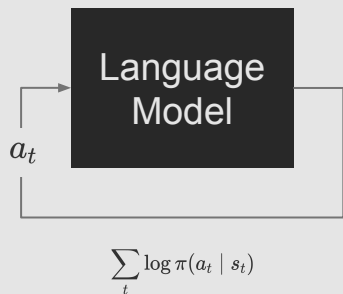
SFT Model

$$\Delta\theta \propto R \cdot \nabla \log \pi_{\theta}(a_t | s_t)$$

Goal: Shift the token distribution towards
those more aligned to user preference

LLM training stages

Pre-training



Goal:
Next token prediction

Supervised Finetuning (SFT)

Input: <User> translate this to Dutch <Model>
Output: Sure, please paste the text you want translated

Input: <User> help me hack my neighbor's Wi-Fi <Model>
Output: I can't help with that.



Pre-trained
LM

Goal:
Instruction-following

Preference Finetuning (RLHF)

SFT Model

$$\Delta \theta \propto R \cdot \nabla \log \pi_{\theta}(a_t | s_t)$$

Goal: Shift the token distribution towards
those more aligned to user preference

Reasoning Finetuning (RLVR)

SFT Model

**One correct answer and
a deterministic way to
evaluate the answer**

Goal: Incentivizing reasoning
(e.g., self-reflection, verification and dynamic
strategy adaptation) through RL

RL for Reasoning



OpenAI o1 System Card

OpenAI

December 5, 2024

2 Model data and training

The o1 large language model family is trained with **reinforcement learning** to perform complex reasoning. o1 thinks before it answers—it can produce a **long chain of thought** before responding to the user. OpenAI o1 is the next model in this series (previously OpenAI o1-preview), while OpenAI o1-mini is a faster version of this model that is particularly effective at coding. Through training, the models learn to refine their thinking process, try different strategies, and recognize their mistakes. Reasoning allows o1 models to follow specific guidelines and model policies we’ve set, helping them act in line with our safety expectations. This means they are better at providing helpful answers and resisting attempts to bypass safety rules, to avoid producing unsafe or inappropriate content.

DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models

Zhihong Shao^{1,2,*}, Peiyi Wang^{1,3,*}, Qihao Zhu^{1,3,*}, Runxin Xu¹, Junxiao Song¹, Xiao Bi¹, Haowei Zhang¹, Mingchuan Zhang¹, Y.K. Li¹, Y. Wu¹, Daya Guo^{1*}

¹DeepSeek-AI, ²Tsinghua University, ³Peking University

{zhihongshao, wangpeiyi, zhuhq, guoday}@deepseek.com
<https://github.com/deepseek-ai/DeepSeek-Math>

Article | [Open access](#) | Published: 17 September 2025

DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, ... Zhen Zhang

[+ Show authors](#)

Nature **645**, 633–638 (2025) | [Cite this article](#)

373k Accesses | 503 Citations | 2561 Altmetric | [Metrics](#)

Abstract

General reasoning represents a long-standing and formidable challenge in artificial intelligence (AI). Recent breakthroughs, exemplified by large language models (LLMs)^{1–3} and chain-of-thought (CoT) prompting⁴, have achieved considerable success on foundational reasoning tasks. However, this success is heavily contingent on extensive human-annotated demonstrations and the capabilities of models are still insufficient for more complex problems. Here we show that the reasoning abilities of LLMs can be incentivized through pure reinforcement learning (RL), charting the road for human-labelled reasoning trajectories.

How to optimize the policy (i.e. LLM)?

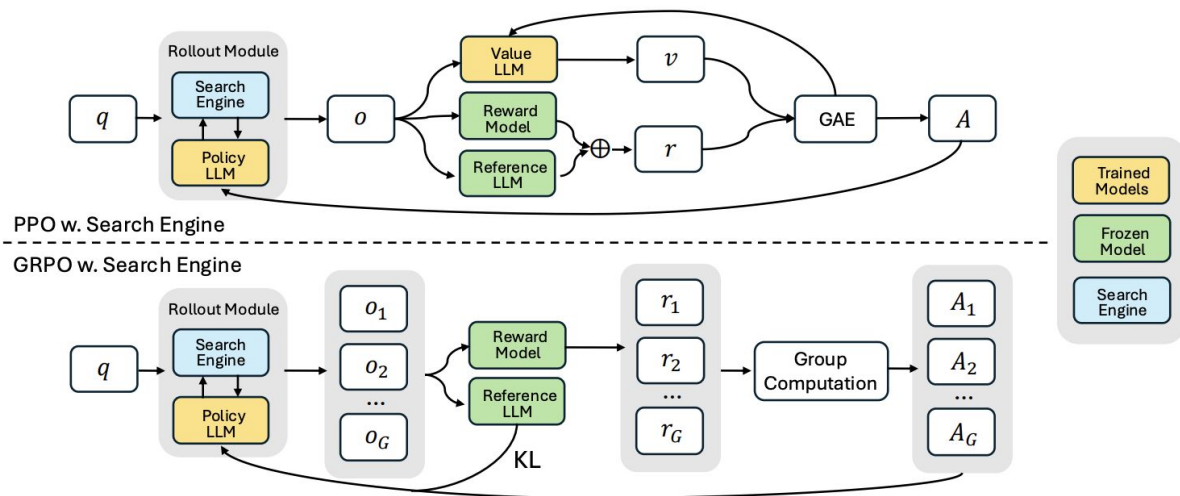
Choose your algorithm: PPO, GRPO, and DPO

	REINFORCE	PPO	GRPO	DPO
Signal	Reward	Advantage	Group-relative advantage	Preference pair
Online sampling	Yes	Yes	Yes	No
Reward model	Optional	Often	Often	No
Critic	No	Yes	No	No
Stability	Low	High	High	High
LLM use	Basic RL baseline	RLHF	Reasoning RL	Preference tuning

Search-R1: End-to-end RL for interleaved reasoning & search

IRCoT/ReAct's interleaving of reasoning + retrieval → Learned end-to-end via RL rather than hand-crafted.

LLM generates tokens → at any step it can emit `<search>query</search>` → retrieval engine returns docs → LLM continues reasoning → final answer → (*outcome-based*) reward signal flows back through entire trajectory.



Rank-R1: RL-based Reasoning in LLM-based Document Rerankers

- Apply the training paradigm of DeepSeek-R1 to reranking
- Model generates a reasoning trace before producing a relevance label (e.g., True or False)
→ Relevance as a reasoned judgment
- Trained with GRPO binary relevance reward *
- Outperforms supervised rerankers on BRIGHT and BEIR

SYSTEM:

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within `<think>` `</think>` and `<answer>` `</answer>` tags, respectively, i.e., `<think>` reasoning process here `</think>` `<answer>` answer here `</answer>`.

USER:

Given the query: "{query}", which of the following documents is most relevant?

[1] {document1}

[2] {document2}

....

[20] {document20}

After completing the reasoning process, please provide only the label of the most relevant document to the query, enclosed in square brackets, within the answer tags. For example, if the third document is the most relevant, the answer should be: `<think>` reasoning process here `</think>` `<answer>`[3]`</answer>`.

* Reward only if the output matches the reasoning and answering format AND relevance label is correct.

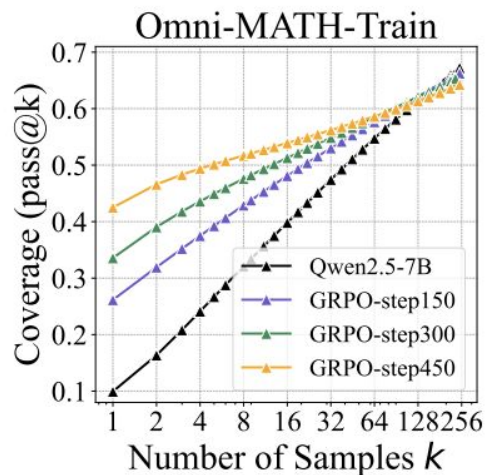
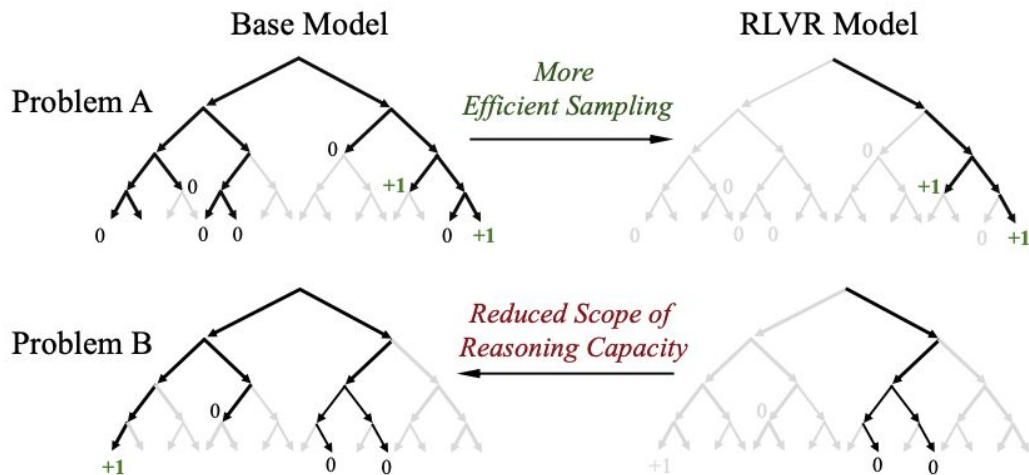
Limits of RL for Reasoning with LLMs

Does RLVR teach LLMs genuinely new reasoning abilities?

- Approach: pass@k (instead of usual greedy)
 - pass@1 = How reliable the model is?
 - pass@k = Can the model solve this problem in any of k attempts?

Limits of RL for Reasoning with LLMs

RLVR Sharpens but Narrows Reasoning.



Limits of RL for Reasoning with LLMs

RLVR yields more reliable but narrower reasoning.

- At small k : RLVR > Base (better sampling efficiency)
- At large k : Base > RLVR (broader reasoning coverage)
- Every correct path in the RLVR model already exists in the base model's distribution

Limits of RL for Reasoning with LLMs

RLVR vs Distillation

RLVR	Distillation
Improves sampling efficiency	Introduces new reasoning patterns
Bounded by base model	Can exceed base model
All algorithms (PPO, GRPO, Reinforce++) perform similarly	Transfers knowledge from stronger teacher

Current RLVR with binary rewards cannot elicit genuinely novel reasoning. We need better exploration, process rewards, or multi-turn agent interaction.

4. Multi-agentic solutions

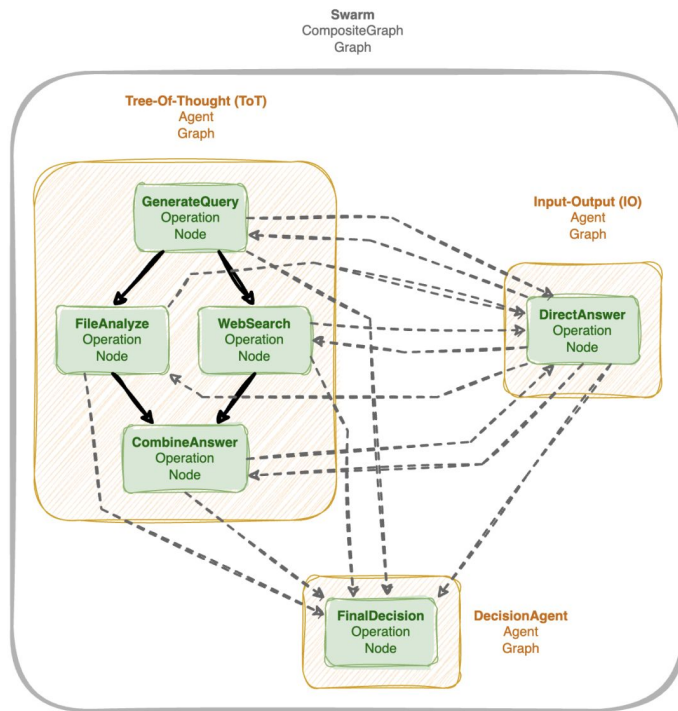
Multi-Agent Architectures

If a single model's capacity for reasoning has a ceiling, can we go beyond that by distributing a given problem across multiple specialized agents?

Multi-Agent Architectures

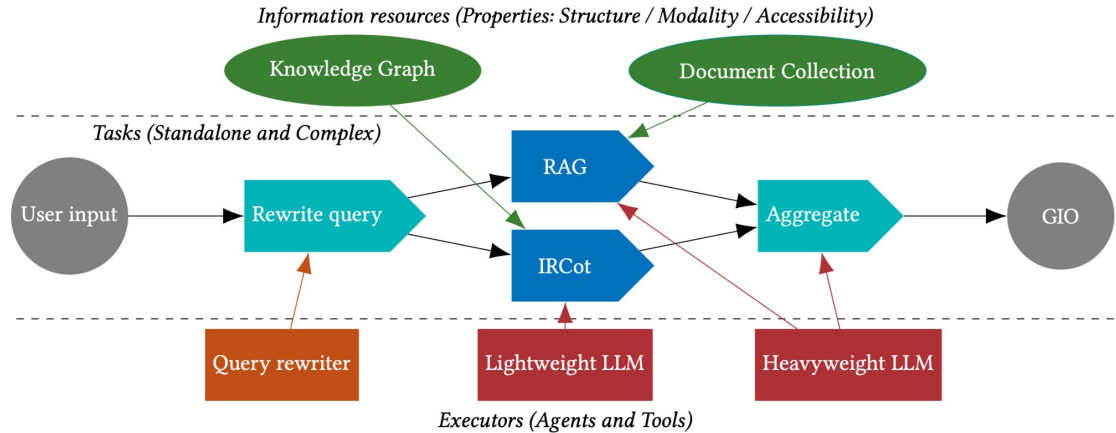
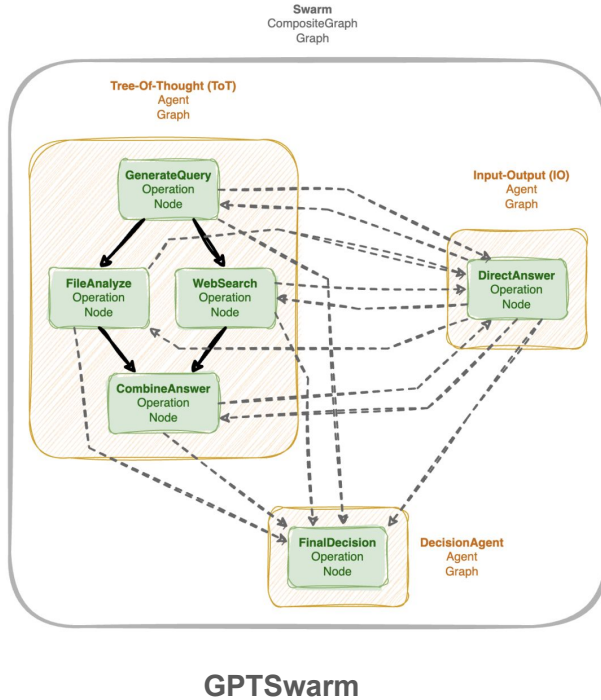
Orchestration

If a single model's capacity for reasoning has a ceiling, can we go beyond that by distributing a given problem across multiple specialized agents?



Multi-Agent Architectures

Adaptive Orchestration



Multi-Agent Architectures

Adaptive Orchestration

Table 3: Performance of each individual agent on the training set (by F1-score and average time in seconds).

	NoR		OneR		IRCoT	
	F1	Time (s)	F1	Time (s)	F1	Time (s)
Context A	0.914	0.66	0.677	6.46	0.730	189.78
Context B	0.061	0.66	0.518	7.34	0.580	192.30
Context C	0.066	0.67	0.146	6.41	0.458	184.85
Overall	0.347	0.66	0.447	6.74	0.589	188.97

Table 4: Evaluation of AQA (NT: Time-agnostic reward, T: Time-based reward) and GPTSwarm [96] on the test set by F1-score and time (log-transformed, in ms). Both have been trained on the training set. For GPTSwarm, the final optimized graph configuration is used for evaluation.

	AQA (NT)		AQA (T)		GPTSwarm	
	F1	Time	F1	Time	F1	Time
Context A	1.0	6.18	1.0	6.18	0.862	12.78
Context B	0.568	12.04	0.539	8.73	0.327	12.79
Context C	0.523	11.75	0.523	11.75	0.317	12.76
Overall	0.697	9.99	0.687	8.89	0.502	12.78

Multi-Agent Architectures

Adaptive Orchestration

Adaptive Orchestration:

Reasoning about what search strategy to execute for a given query.



Multi-Agent Architectures

Adaptive Orchestration

Adaptive Orchestration:

Reasoning about what search strategy to execute for a given query.

Open challenges:

1. Dynamic routing mid-trajectory
2. Efficient training
3. Granularity of node decomposition
4. Evaluation



Why do multi-agent systems fail?

Error cascading

- One agent's mistake becomes every agent's context
- Early retrieval errors propagate through the whole chain

Coordination overhead

- Inter-agent messages cost latency and tokens
- Agents re-derive shared context (redundant reasoning)
- More agents $\neq$ better reasoning past a threshold

Attribution & verifiability

- Which agent caused the error? Blame is non-trivial
- Trace is distributed $\rightarrow$ faithfulness harder to check
- Final-stage verifiers alone are insufficient

3 root causes ; **poor specification** · **inter-agent misalignment** · **weak verification**

Limitations of current LLM-based approaches,
motivate exploring other approaches;
Neuro-symbolic, Bayesian, geometric, ...

Coming Up:

- I. Teaching retrievers to reason,
- II. The division of labor: better agent, or better retriever?
- III. Neuro-symbolic and structured reasoning,
- IV. Reasoning under uncertainty.

Coffee Break

III. Teaching Retrievers to Reason

1. Enhanced training data:

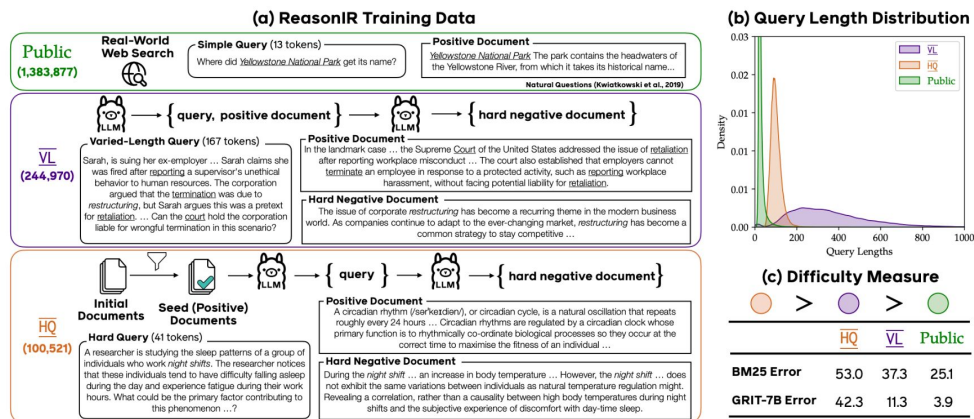
ReasonIR

The Assumption

- Reasoning-intensive retrieval fails because of non-representative training data
- MS MARCO-style pairs teach matching not "helpful for reasoning toward an answer"

The fix

- Synthesize the missing data; train a Llama3.1-8B bi-encoder with a standard contrastive loss



BRIGHT nDCG@10: 24.4 → 29.9 with query rewriting, **faster than LLM rerankers.**

ReasonIR: the data recipe

Synthetic training data (ReasonIR-Synthesizer, Llama3.1-70B)

- Varied-length data: 300–2,000-word reasoning-style queries with synthetic positives
- Hard-query data: reasoning-intensive questions brainstormed from "reasoning-worthy" seed documents
- Multi-turn LLM-written hard negatives: plausibly related but unhelpful — exactly the BRIGHT failure mode
- Mixture: ~1.38M public samples + 245k varied-length + 100k hard queries; ablations show the two synergize

Behavior

- Keeps improving with longer GPT-4 query rewrites while other retrievers plateau; 36.9 nDCG@10 with a Qwen-32B reranker

ReasonIR: **Enhance the training data for reasoning intensive queries.**

2. Enhanced Representation Space

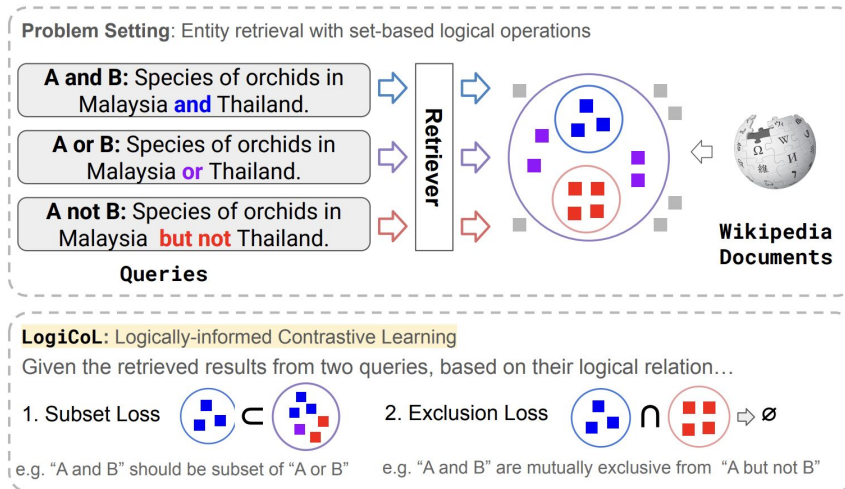
Set-compositional queries: LogiCoL

The failure

- "A AND B" vs "A NOT B" → nearly identical embeddings
- Logical operators are invisible to the encoder

The fix

- Compile the logic into the contrastive loss as soft constraints
- Batches group logically-related query variants so the model sees the contrast



Better recall on QUEST and >20% fewer logical violations, **no reranker, no decomposition at inference.**

LogiCoL: logic as a training signal

Objective: $L = L_{\text{SupCon}} + 0.1 \cdot L_{\text{subset}} + 0.1 \cdot L_{\text{exclusion}}$

- Subset consistency (product t-norm):
if $\text{answers}(q_1) \subseteq \text{answers}(q_2)$, a doc relevant to q_1 must also score high for q_2
- Exclusion consistency (symmetric KL): push apart the score distributions of "A AND B" and "A NOT B"

Training data: QUEST + auto-generated logical variants

- Query variants share atomic sub-queries but differ in operators (6.2k train / 11.9k test)

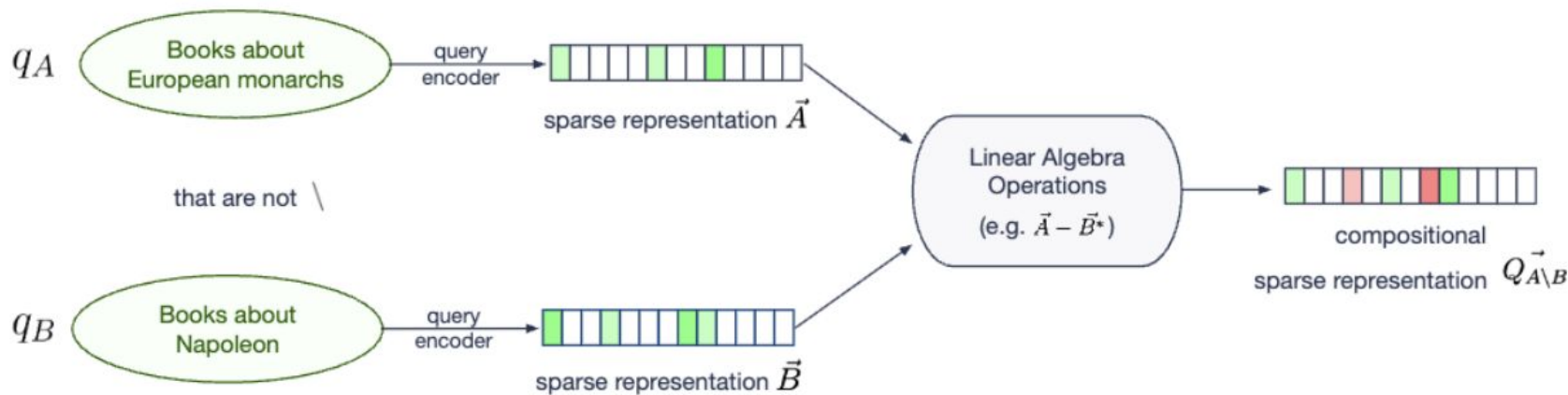
Results (four ~110M encoders: GTR, Contriever, GTE, E5)

- E5: R@100 21.9 vs 19.8, R@1000 50.2 vs 45.5; largest gains on negation queries
- Violation rate (excluded docs outranking gold) down >20% across all backbones

Symbolic structure injected purely through the loss, **architecture and inference stay a plain bi-encoder.**

Set-compositional queries: SetComp-LSR

Zero-shot alternative to LogiCoL: keep the retriever frozen, and compose sub-query representations with set-theoretic vector algebra ($\cap$, $\cup$, $\setminus$) at query time.



Disentangled negation $\approx 2 \times \text{nDCG@10}$ ($0.132 \rightarrow 0.258$), beating trained models — **but intersections stay unsolved, trained or not.**

On generalizability of these approaches
(ReasonIR/LogiCoL/SetComp): Enhancing training data
may not be as effective as we thought!

Reproducing Complex Set-Compositional Information Retrieval

Vincent Degenhart
Radboud University
Nijmegen, NL
vincent.degenhart@ru.nl

Dewi Timman
Radboud University
Nijmegen, NL
dewi.timman@ru.nl

Arjen P. de Vries
Radboud University
Nijmegen, NL
arjen.devries@ru.nl

Faegheh Hasibi
Radboud University
Nijmegen, NL
faegheh.hasibi@ru.nl

Mohanna Hoveyda
Radboud University
Nijmegen, NL
mohanna.hoveyda@ru.nl

Check our presentation on Wednesday!

Session: Efficient Retrieval, Training, and Indexing - 10:30 am - Courtyard 1+2

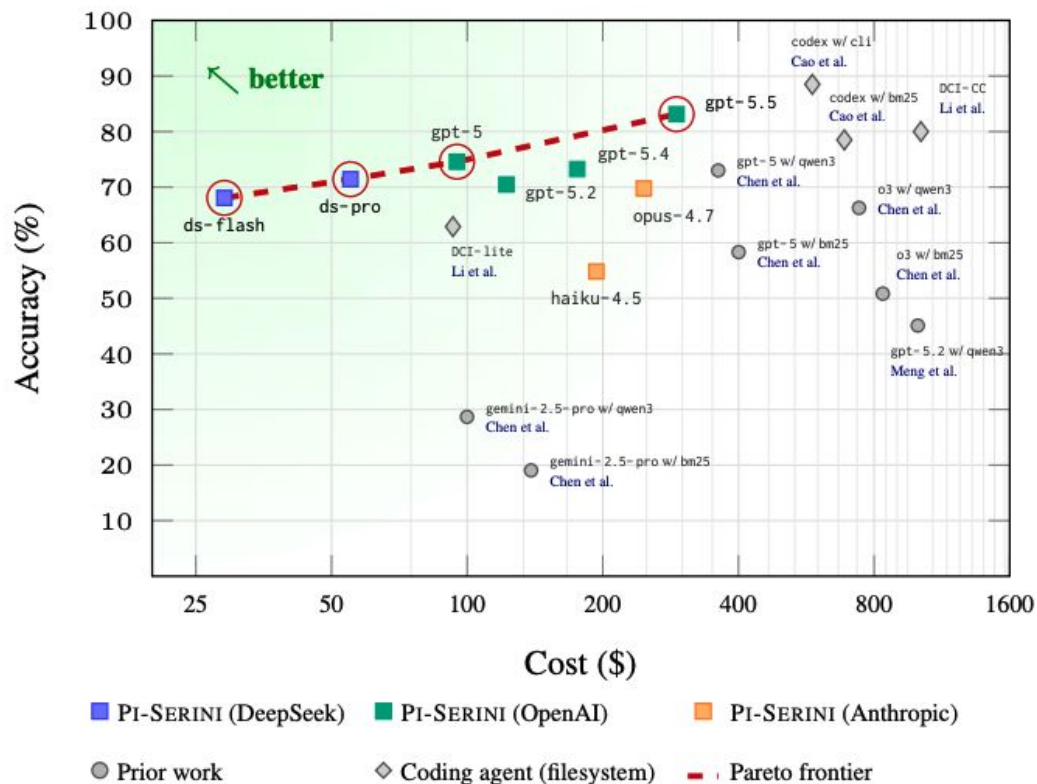
3. The division of labor: Better agent, or better retriever?

Is Lexical Retrieval Sufficient?

Does a lexical retriever suffice as large language models (LLMs) become more capable in an agentic loop?

Answer: mostly yes, if configured

- Tuned BM25 (k1=25, b=1) + retrieval depth 1,000 + cached-ranking tools
- gpt-5.5: 83.1% on BrowseComp-Plus, ~95% evidence recall
- Beats dense-retriever agents at 3–10× lower cost



Pi-Serini: why the "dumb" retriever wins

A deliberately minimal agent

- ReAct loop + 3 tools: search (BM25, top-1000 cached) · page through the cached ranking · read one document
- Time budget (300s) instead of an iteration cap; forced answer synthesis at 70% of budget

Where the gains actually come from

- Tuning BM25 for long documents: accuracy 64.0 → 82.0 (+18.0)
- Depth $k = 5 \rightarrow 1,000$: surfaced evidence recall 70.5 → 95.8 (+25.3)
- Same LLM (gpt-5), proper config: 74.6% vs 58.3% for the released BM25 baseline

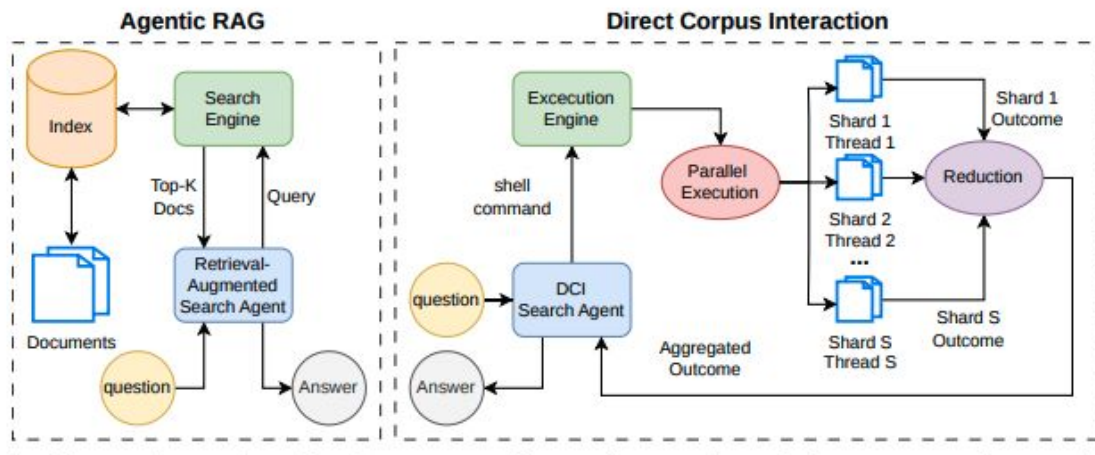
Caution for the field: untuned lexical baselines inflate dense-retriever gains — **you may be measuring configuration debt, not semantics.**

Is Grep Retrieval Sufficient?

Learning to retrieve via
RL-training LLM agent on
DCI.

How

- No index — agent runs grep pipelines on the raw corpus
- SFT on synthetic trajectories, then GRPO
- Best micro-avg F1 on 7 QA benchmarks (0.569 vs 0.544)



IV. (Neuro-)Symbolic & Structured Reasoning

1. Neuro-Symbolic Reasoning

Neuro Symbolic Computing

Neuro Symbolic Computing in ML

Neuro-symbolic computing aims at integrating two most fundamental cognitive abilities: the ability to learn from the environment, and the ability to reason from what has been learned. Neuro-symbolic machine learning aims to reconcile the advantages of robust learning in neural networks and reasoning and interpretability of symbolic representation.

E.g., [International Conference on Neurosymbolic Learning and Reasoning \(NeSy\)](#), since 2005.

Neuro Symbolic Computing in IR

The primary application area for neuro symbolic methods in modern IR has been aimed at integrating knowledge graphs in (dense) retrieval.

Disclaimer about ranking with entity annotations

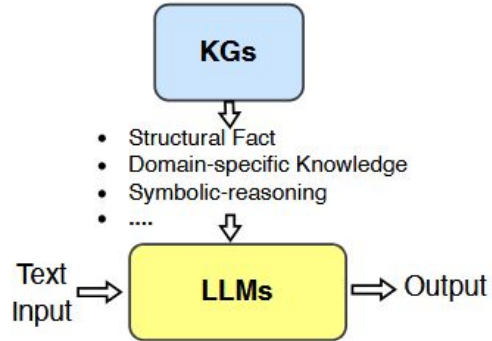
The question whether open domain entity linking (annotating documents and queries with their entity mentions) can really help improve the retrieval system's effectiveness is still an open one. Especially in domain-specific settings, entity annotations are also useful on their own, e.g., to continue browsing from the search results, or to explain inferred relevance.

Characterizing a Semantics for Cognitive Computation

"The aim here is to identify a way of looking at and manipulating commonsense knowledge that is consistent with and can support what we consider to be the two most fundamental aspects of intelligent cognitive behavior: the ability to learn from experience, and the ability to reason from what has been learned. We are therefore seeking a semantics of knowledge that can computationally support the basic phenomena of intelligent behavior."

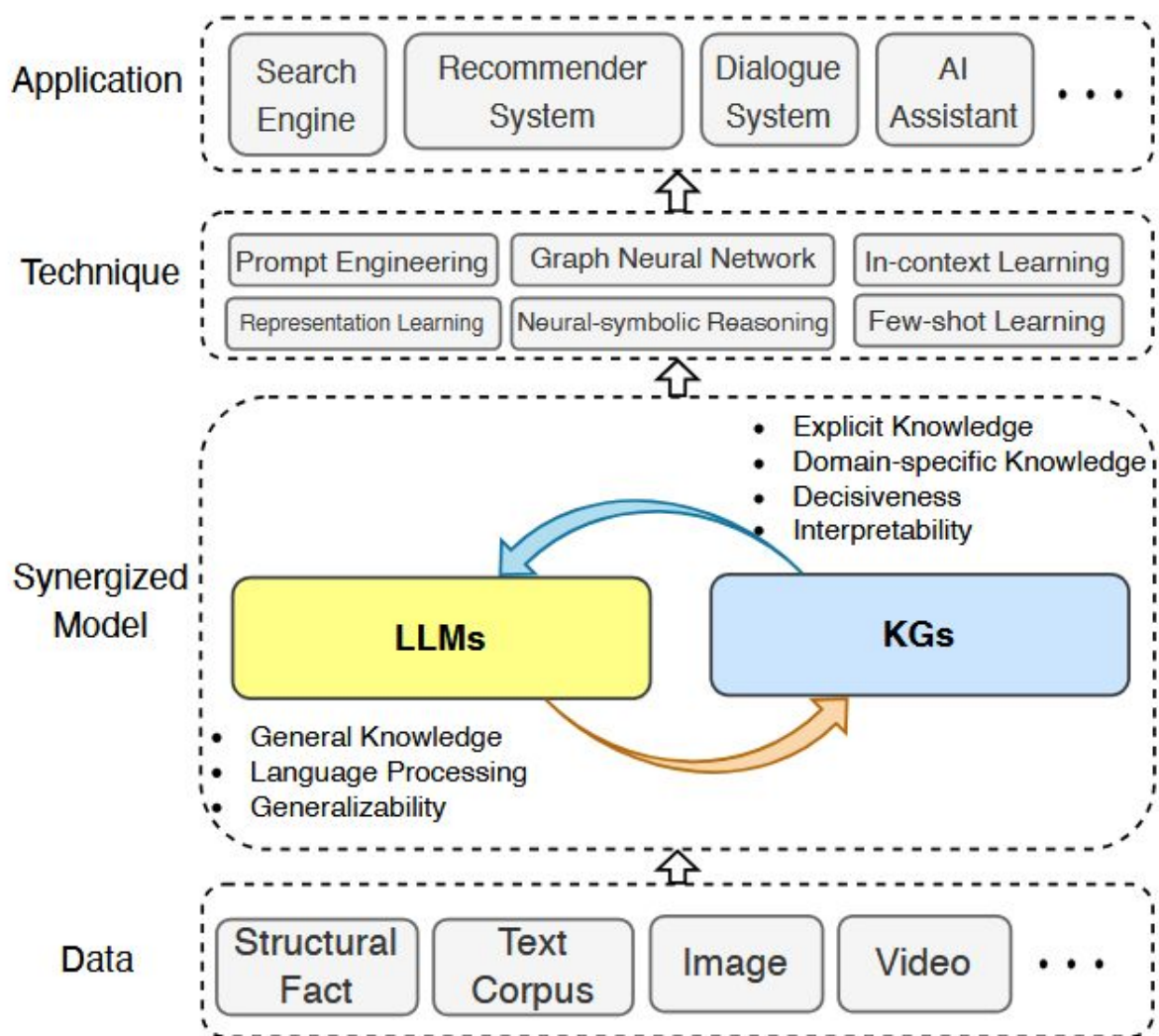
Valiant, 2003

Knowledge Graphs (KGs) and LLMs

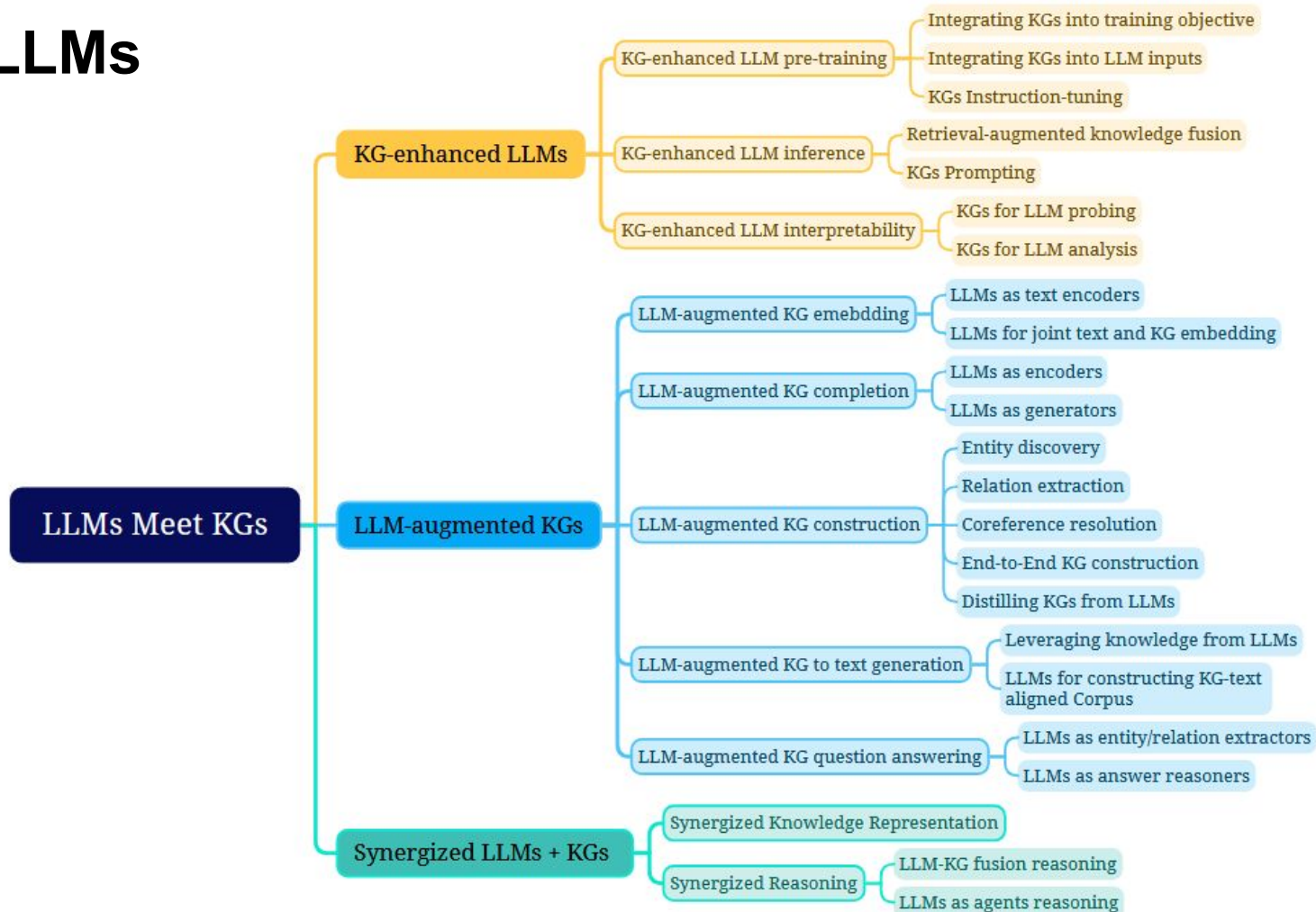


a. KG-enhanced LLMs

KGs and LLMs



KGs and LLMs



Information Retrieval and Logic

In 1986, two papers by Keith van Rijsbergen initiated quite a movement aiming to develop retrieval models grounded in (non-classical) logic, using **plausible inference** instead of a strict inference.

1. PREFACE

In the last few years, I have become increasingly dissatisfied with the state-of-the-art in information retrieval. I have reluctantly concluded that the fundamental basis of all the previous work is wrong. Almost all of the previous work in Information Retrieval (including my own) has been based on the assumption that a formal notion of meaning is not required to solve the information retrieval problems.

Typically, researchers have assumed that one could get by, by only considering absence or presence of word tokens in text together with counting information about the distribution of words. Although such an approach has been successful up to a point, it has become clear that further advances in the effectiveness of retrieval by such techniques are not possible. My observation is that performance based on statistical techniques has reached its theoretical limit and any attempts to achieve further improvements are a waste of time. This is not to say that systems based on these techniques are not worth building; on the contrary, they are because it is the best we have to date. But to build a new generation of Information Retrieval systems, a new theory will be needed.

Information Retrieval and Logic

In 1986, two papers by Keith van Rijsbergen initiated quite a movement aiming to develop retrieval models grounded in (non-classical) logic, using **plausible inference** instead of a strict inference.

An extra benefit of such a new theory for information retrieval would be that its formalization using, e.g., situation theory (Huibers et al., 1996) could also serve as a meta-theory for IR; a theory that could help decide questions as whether a retrieval model should be non-monotonic (i.e., relevance estimates may change upon the availability of new information).

Why logic for IR?

Consider two users with different ideas of relevance but submitting the same request; they should get different probabilities of relevance.

Van Rijsbergen (1986) proposes to separate the logical implication $s \rightarrow q$ from the measure of its uncertainty, $P(s \rightarrow q)$

- separating the process of revising probabilities from the logic, and
- separating the treatment of relevance from the treatment of documents and request.

Why logic for IR?

Chiaramella and Chevallet (1992) argue that:

“The logic model thus shows in a neat way how knowledge bases could be used while processing queries.”

“This intrinsic power of logic is the best available way for the integration of recent approaches such as AI and NLP within a coherent, controllable, framework.”

Why logic for IR?

Fuhr (2000) introduces a combination of Datalog with probability theory, writing:

“This approach allows for easy formulation of specific retrieval models for arbitrary applications, and classical probabilistic IR models can be implemented by specifying the appropriate rules. In comparison to other approaches, the possibility of recursive rules allows for **more powerful inferences**, and predicate logic gives the **expressiveness required for multimedia retrieval**. Furthermore, probabilistic Datalog can be used as a query language for **integrated information retrieval and database systems**.”

More grip on “relevance”?

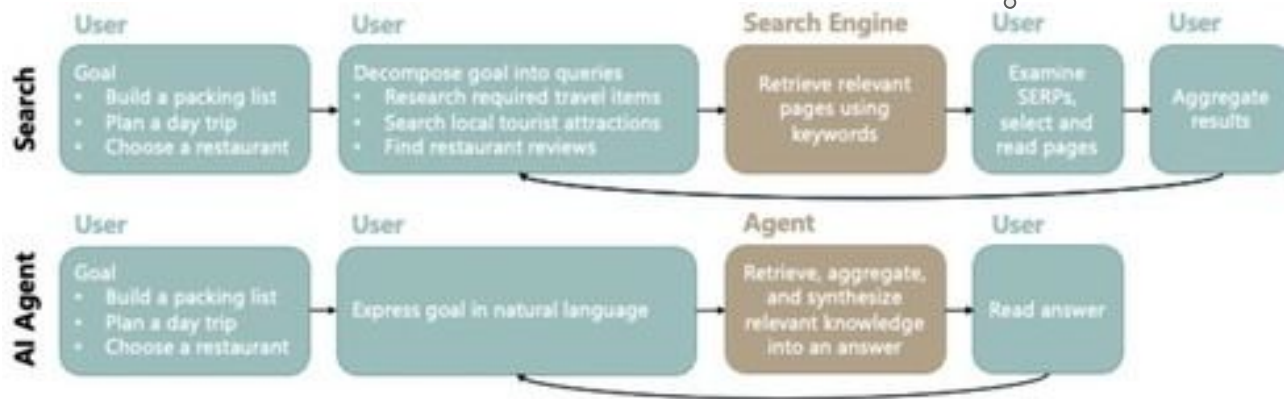
Different relevance criteria:

- Topicality: the classic question of **aboutness**
- Readability: can I understand the text?
- Authoritativeness: can I trust the text?
- Child-suitability, timeliness, *etc*

A logical model for IR (e.g., specified as a probabilistic datalog program) would allow us to express the relevance criteria directly in the retrieval model, instead of handling these separately.

More grip on “relevance”?

Inherently dependent on user, task, context.



Really, user goals in natural language only?

With “agentic search” (White, 2024), more of task and context will become part of the “search” than before, not less! Defining the expected result precisely will be crucial input to drive “search agents” and assess success or failure of intermediate steps.

Abductive Reasoning

A logical framework could enable a new type of reasoning to assess relevance, different from classic deduction (databases) and inference (classic IR), known as abduction (the [“Sherlock Holmes Solving Methodology”](#)).

Abductive reasoning may be more appropriate in the context of incomplete knowledge, that is inherent to all “true” information retrieval tasks (beyond Q&A).

E.g., promising for modelling relevance feedback, where the reasoning process constructs an “explanation” — a descriptive representation or subset of features — that best accounts for why the user marked a subset of documents as relevant.

LINC: Logical Inference via Neurosymbolic Computation

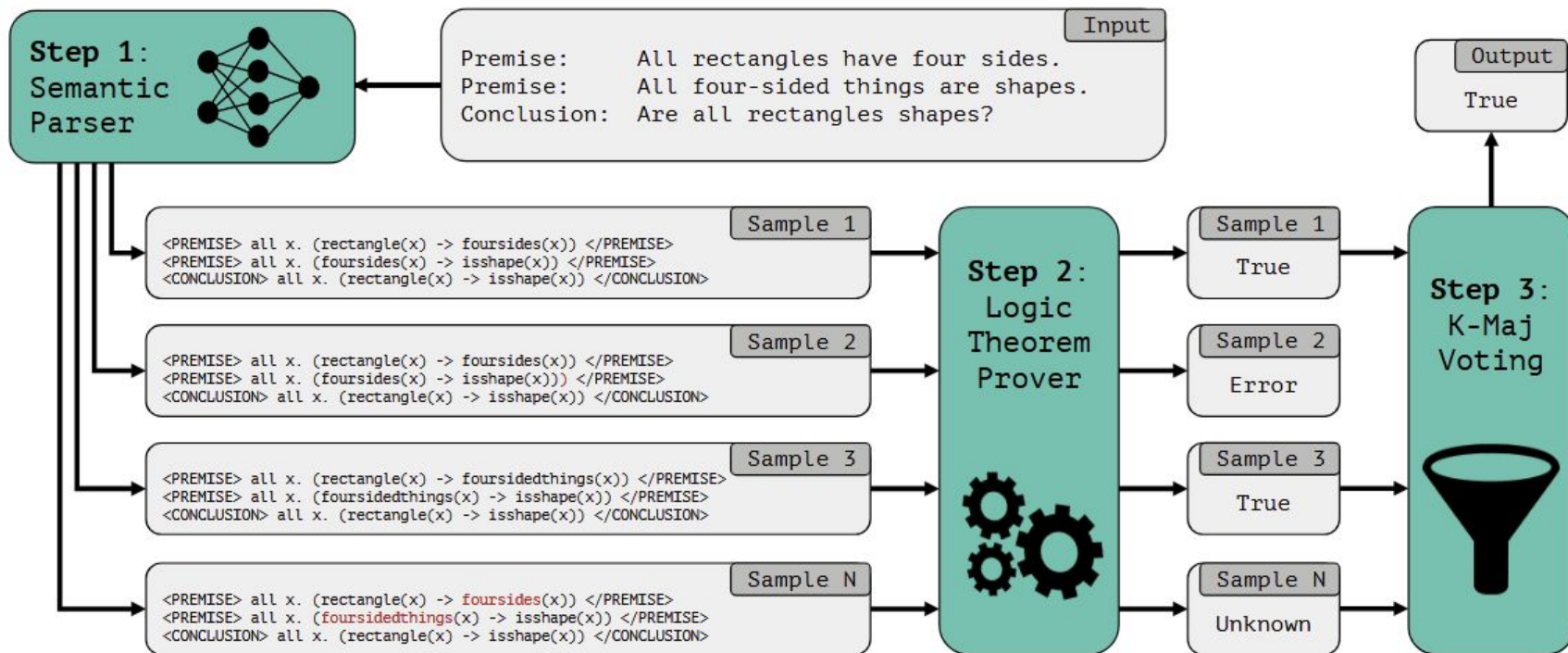
A two-step neurosymbolic process for logical reasoning:

Convert natural language premises and desired conclusion into first-order logic.

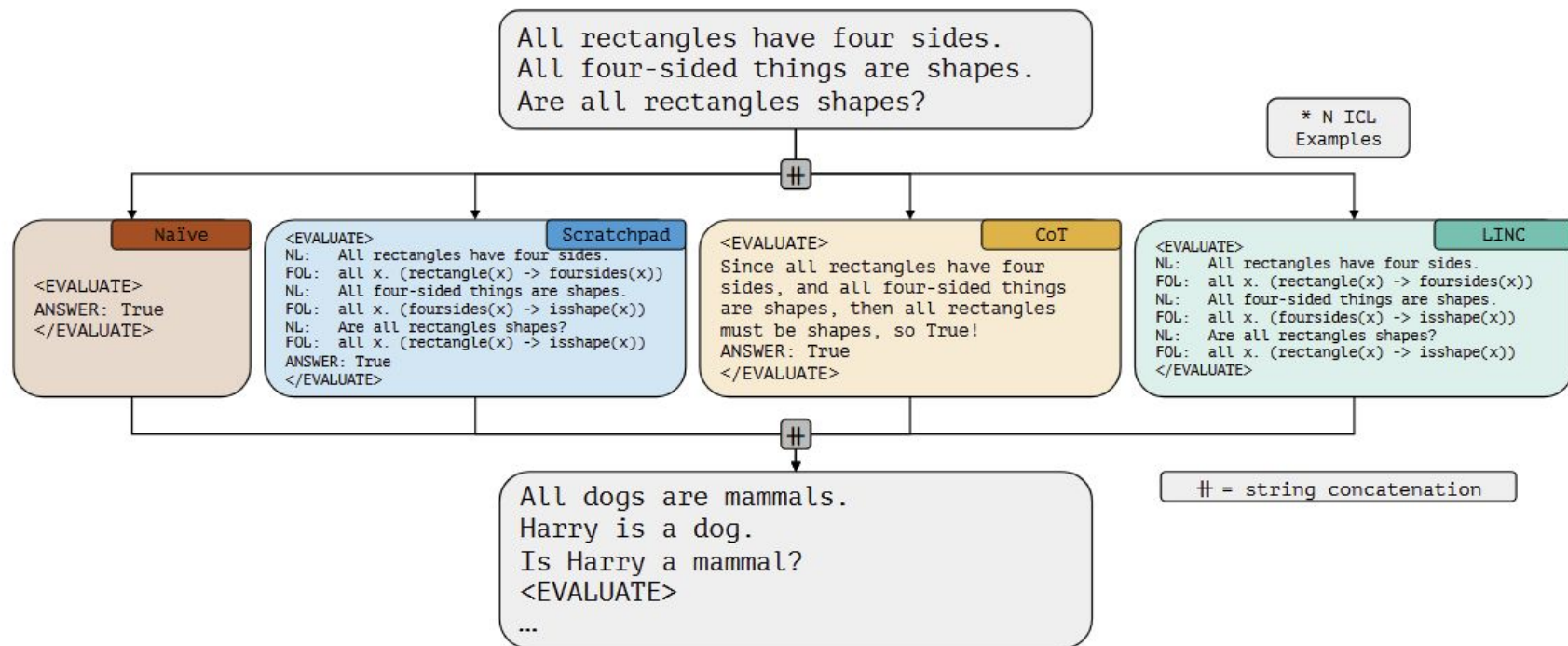
Let a symbolic FOL theorem prover algorithmically determine the truth value of the conclusion given the formalized premises.

In practice, they try 10 different attempted formulations from step 1 and apply a third majority voting step.

LINC: Logical Inference via Neurosymbolic Computation



LINC: Logical Inference via Neurosymbolic Computation



Failure Modes comparison

L1: FOL fails to capture implicit information not mentioned in the premises.

L2: FOL fails to capture information explicitly mentioned in the premises due to the choice of representation.

L3: FOL contains syntax errors.

C1: CoT concludes something different than it suggests.

C2: CoT makes incorrect logical deductions.

C3: CoT fails to find complex paths of reasoning.

LINC and CoT mispredict on different examples, and mispredictions of in-context reasoning baselines are more similar to each other than to LINC.

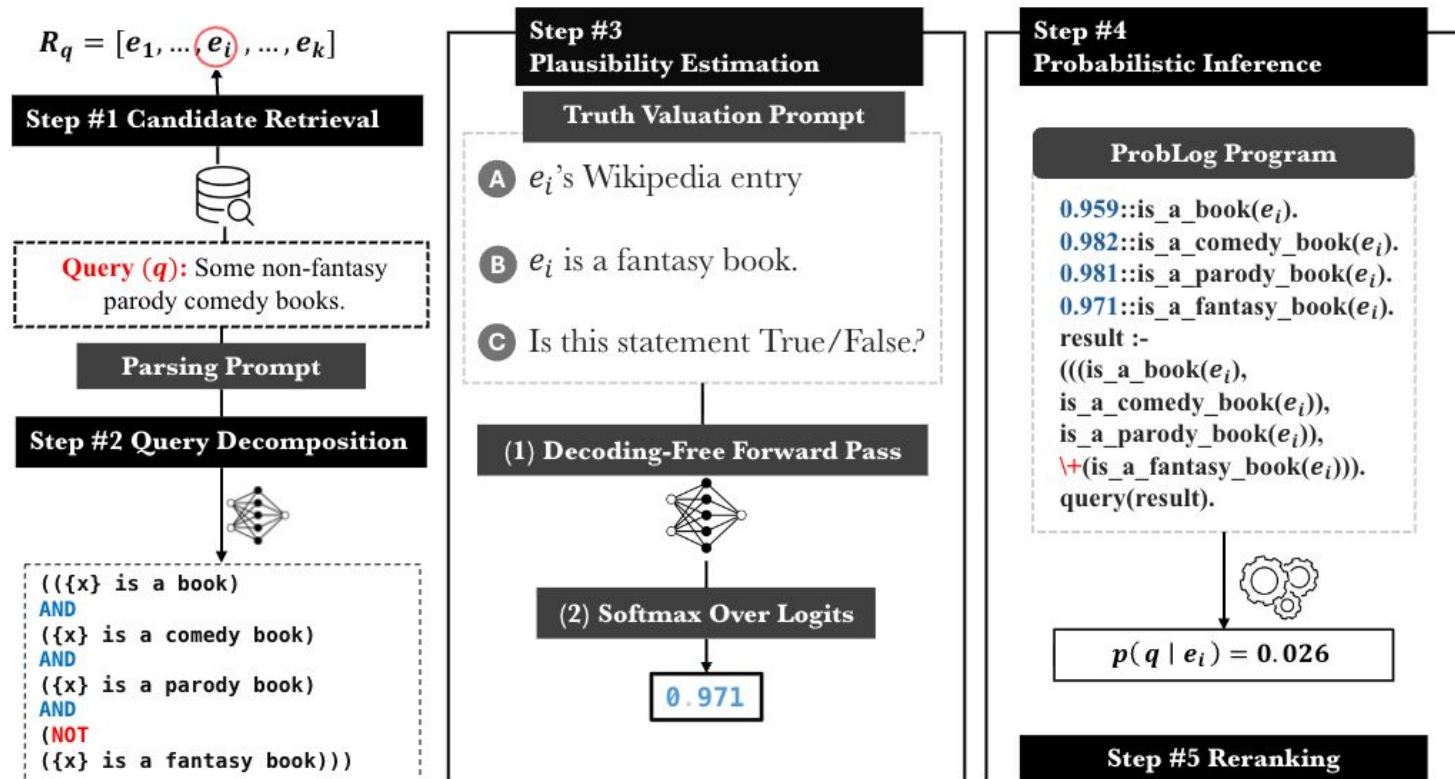
Theorem Proving for IR?

Imagine a **Logician** who knows Probabilistic Prolog.

She answers the complex entity-seeking queries from the QUEST dataset (sets of Wikipedia entities defined through combinations of set-theoretic operations, intersection, union, and difference).

The Logician consults the **Oracle**, an LLM that estimates the plausibility of the facts stated in the ProbLog programme. *Note that the Oracle could be wrong!*

OrLog



OrLog results

OrLog consistently improves top-rank results and shows its largest advantages on disjunctive structures, where the LLM treats disjunctive constraints as conjunction.

End-to-end LLM reasoning uses 6-10 times more tokens than OrLog:

- (i) decoding-free predicate plausibility estimation (a single forward pass per predicate)
- (ii) one-off query parsing with composition delegated to ProbLog, whose runtime and computational cost are negligible relative to an LLM call.

ProbLog vs LLM for Probabilistic Reasoning

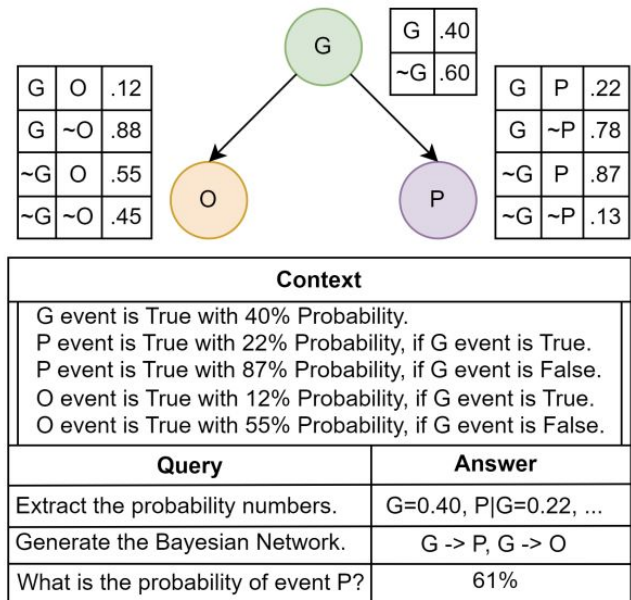


Figure 1: An example from the BLInD dataset including an underlying Bayesian network, its textual description, and probabilistic queries in natural language form.

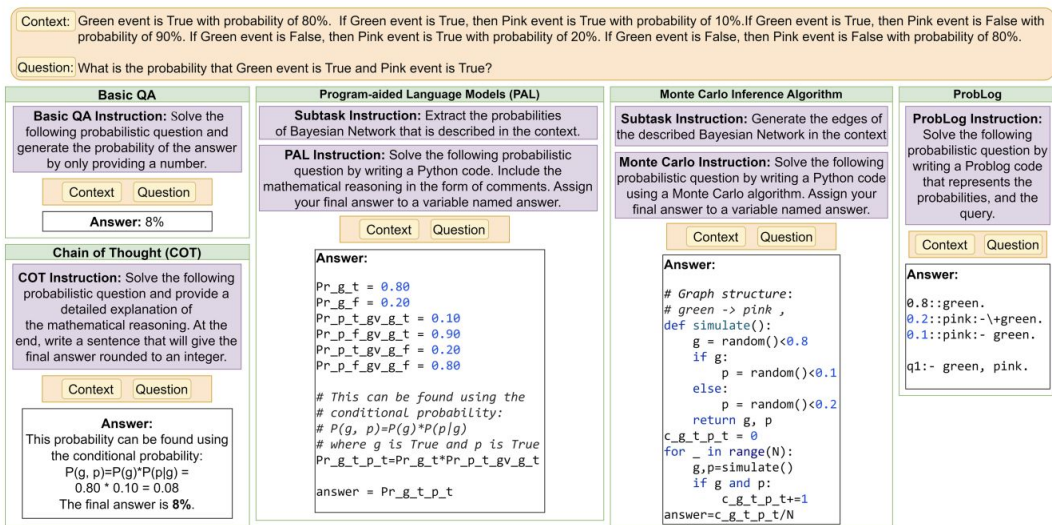


Figure 2: This figure shows our main prompting approaches, PAL, Monte Carlo, and ProbLog, alongside the baseline approaches, Basic QA and COT. Each prompt begins with an instruction (purple boxes) that describes the problem and the answer format. Then, the context, question, and answer are demonstrated depending on the approach. We display only the first in-context example here but use 3 in our experiments. When we require the use of our designed subtasks in the prompt, their instructions and answers are prepended to the main approach, as shown in the PAL method for *Number Extraction* and the Monte Carlo method for *Graph Generation*.

ProbLog vs LLM for Probabilistic Reasoning

Model	Method	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{2-5}	V_{6-10}	V_{2-10}
GPT3.5	PAL	66	34	25	17	14	9	6	5	2	35	7	19
	PAL w/NE	85	66	41	27	19	12	5	3	6	54	9	29
	Monte Carlo	79	63	71	65	41	32	33	18	14	69	27	46
	Monte Carlo w/GG	85	82	83	68	42	31	28	18	8	79	25	49
	ProbLog	87	82	88	75	59	52	46	38	35	83	46	62
Llama3	PAL	100	84	57	36	31	20	10	14	8	69	17	40
	PAL w/NE	100	95	71	52	46	28	16	16	9	79	23	48
	Monte Carlo	100	100	96	96	92	85	77	72	64	98	78	87
	ProbLog	90	95	92	87	95	94	87	82	78	91	87	89
GPT4	PAL	100	86	70	58	50	27	21	14	7	78	24	48
	PAL w/NE	99	96	78	64	43	26	14	14	10	84	21	49
	Monte Carlo	100	99	98	100	92	94	92	90	88	99	91	94
	Monte Carlo w/GG	100	97	99	98	97	96	88	92	85	99	92	95
	ProbLog	99	98	100	100	96	97	97	98	96	99	97	98

Table 4: GPT3.5, Llama3, and GPT4 accuracy results, presented as percentages, for the PAL, Monte Carlo, and ProbLog methods. w/NE and w/GG denote the inclusion of *Number Extraction* and *Graph Generation*. The columns represent dataset splits V_i , and the average results for smaller BNs V_{2-5} , larger BNs V_{6-10} , and all BNs V_{2-10} .

Wrapping up Part 4.1

Neuro-Symbolic Computing

Reconciles the advantages of robust learning in neural networks and reasoning and interpretability of symbolic representation.

KGs and LLMs

The amount of research that mixes knowledge graphs and large language models is tremendous, but has as yet not had a lot of impact on our field specifically.

Logic and IR

The idea to apply (non-classical) logics to improve information retrieval has a long history in the field. Modelling information access symbolically may have many advantages. Will we manage to finally take up this huge challenge again?

LLMs & Probabilistic Logic Programming

LLMs can turn complex questions into probabilistic logic programs and provide the probability estimates needed for a solver like ProbLog to provide the answer, more accurately and at a lower cost.

V. Reasoning Under Uncertainty

Outline

What we have seen so far

- Reasoning as a multi-step inference strategy (e.g., CoT, "thinking")
- Learning-to-reason on static information with grounded signals and gradient-based model updates

What this session discusses

- Uncertainty as the control variable of retrieval, and search agents
- Bayesian reasoning as a framework to navigate uncertainty
- Act based on current “belief” and updating a model's beliefs via observation

Why Bayesian methods show up in LLM work

- LLMs are powerful but overconfident, poorly calibrated and brittle in multi-step reasoning
- The **papers discussed in this section** (*next slide*) introduce
 - explicit beliefs
 - principled updating
 - decision-making under uncertainty

Question: Where do we plug Bayesian methodology into the LLM training?

Reasoning Under Uncertainty

- Step 1: Choose the uncertain object
- Step 2: Approximate a posterior over it
- Step 3: Use that posterior to act

Examples of Uncertain Objects

- hypothesis support (AutoDiscovery)
- reasoning uncertainty (Reasoning in Flux)
- Supporting evidence in RAG

Bayes as a control signal

Reasoning in Flux (Yin et al., 2024)

- Tracks token-level uncertainty during CoT
- When uncertainty jumps, it backtracks
- Then injects "reasoning clues" to repair the chain

Question:

Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for \$2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?

Correct Reasoning:

[1] 16 eggs per day. [2] 3 for breakfast, 4 for muffins, and the rest for sale. [3] $16 - 3 - 4 = 9$ eggs for sale. [4] 9 eggs for sale at \$2 each is \$18. [5] So the answer is \$18.

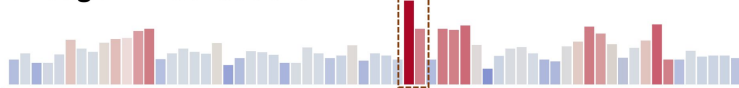
Wrong Reasoning:

[1] 16 eggs are laid per day. [2] Janet eats 3 for breakfast every day. [3] $16 - 3 = 13$ eggs are left. [4] 13 eggs are baked into muffins. [5] $13 \times 4 = 52$ muffins. [6] 52 muffins are sold for \$2 each. [7] $52 \times 2 = 104$ dollars. [8] So the answer is 104 dollars.

Correct NLL Distribution:



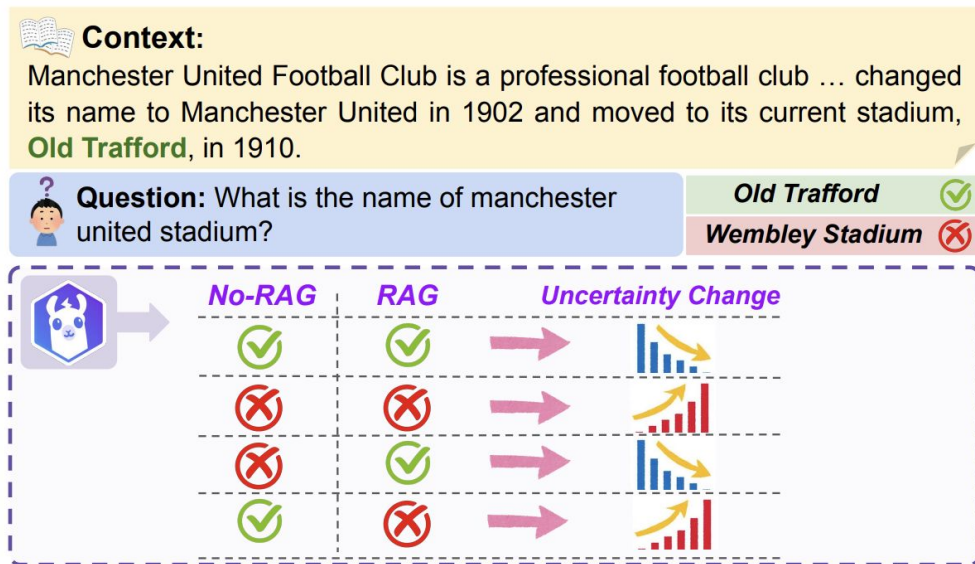
Wrong NLL Distribution:



Does UQ methods generalize to RAG? (Spoiler: no)

An axiomatic analysis (Soudani, Kanoulas, Hasibi. Findings of ACL 2025)

- Five formal axioms any uncertainty estimator should satisfy once retrieved evidence enters the prompt
- No existing method satisfies them all — estimators report LOW uncertainty in RAG whether or not the context is relevant
- Simple fix: calibrate uncertainty scores with context-relevance signals



Uncertainty as the control signal of search agents

R²C: UQ for retrieval-augmented reasoning (Soudani, Zamani, Hasibi, 2025)

- Perturb intermediate reasoning steps (rewrite query, critique, re-validate) and measure outcome consistency
- >5% AUROC over strong UQ baselines; drives abstention and model selection

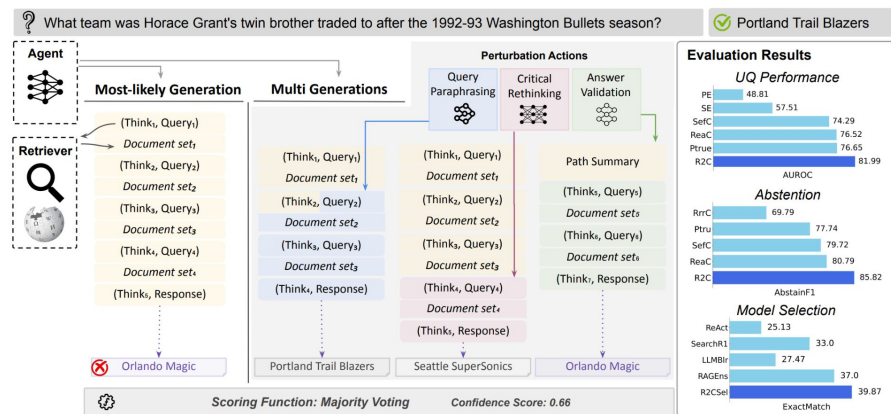


Figure 1: R²C overview. Given a user query, the agent (LLM) first generates the most-likely reasoning path leading to the most-likely response (left, yellow). To estimate uncertainty, R²C creates multiple perturbed generations by randomly altering states in the reasoning path (middle, gray). The confidence score is then derived via majority voting. R²C significantly outperforms established UQ methods and achieves significant improvements on two downstream tasks: abstention and model selection.

Step 3 of the lens, applied to IR: **the posterior decides when to retrieve, when to stop, and when to abstain.**

Takeaways

- Reasoning under uncertainty with LLMs means
 - choosing what is uncertain
 - approximating a posterior
 - using it to make better decisions
- The papers in this tutorial use UE methods to
 - explore better
 - predict more reliably
 - "repair" reasoning
 - update beliefs
 - fine-tune with calibrated uncertainty

Bottom line

Can Bayesian methods enhance the reasoning capabilities of LLMs?

Probably



5. Bridging current methods/challenges & future directions

Comparative analysis of existing reasoning paradigms along key dimensions:

- A. Representational Capability,
- B. Inference Faithfulness,
- C. Computational Feasibility.

Here again: A 12-type taxonomy, with evaluation coverage

Reasoning type	IR manifestation	Coverage
Deductive	Logical entailment between query and document	Partial
Inductive	Generalizing from retrieved evidence to claims	Partial
Abductive	Query intent inference; hypothesis formation from evidence	Untested
Analogical	Cross-domain retrieval; structural relevance beyond surface	Untested
Causal	Distinguishing correlation from intervention evidence	Untested
Defeasible	Provisional relevance; belief revision on new retrieval	Untested
Modal	Tracking claim strength (established/contested/speculative)	Untested
Epistemic	Knowledge state tracking across a literature over time	Untested
Dialectical	Mapping argumentative position within a field's disputes	Untested
Temporal	Evidence currency; supersession; historical state tracking	Partial
Metacognitive	Query reformulation; knowing when to stop; uncertainty flag	Partial
Practical	Search strategy decisions in agentic pipelines	Untested

How do the discussed methods compare?

Representational capability

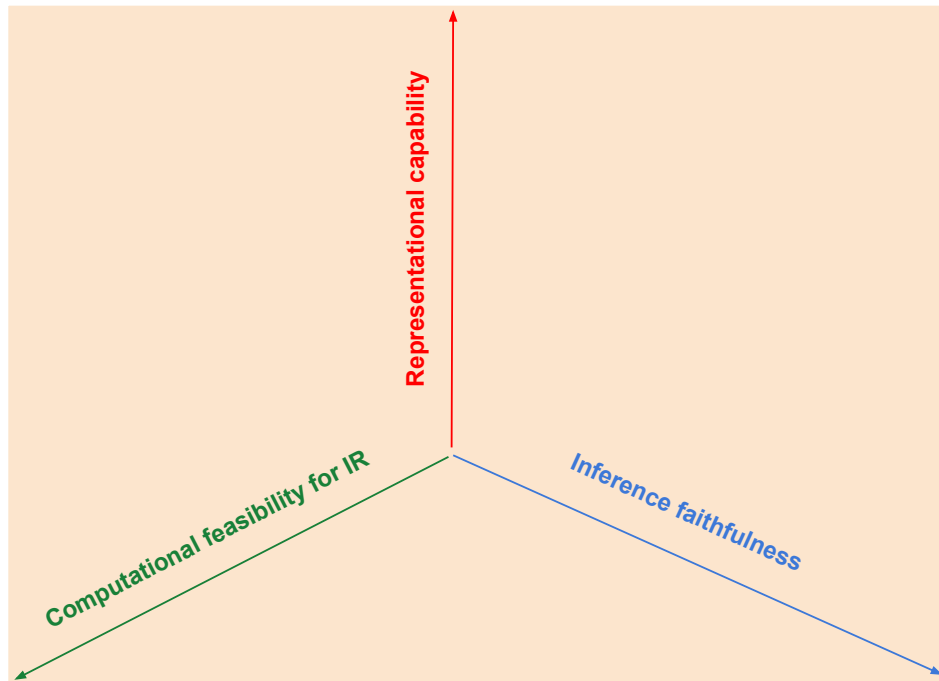
E.g., Does reasoning proposals explicitly encode uncertainty, causal relationships, etc.?

Inference faithfulness

Can faithfulness of the reasoning process be verified?

Computational feasibility for IR

Is the reasoning process computationally feasible over a large set of documents?



Where would you place the methods?

Representational capability

E.g., Does reasoning proposals explicitly encode uncertainty, causal relationships, etc.?

Inference faithfulness

Can faithfulness of the reasoning process be verified?

Computational feasibility for IR

Is the reasoning process computationally feasible over a large set of documents?

Inference-time reasoning with LLMs

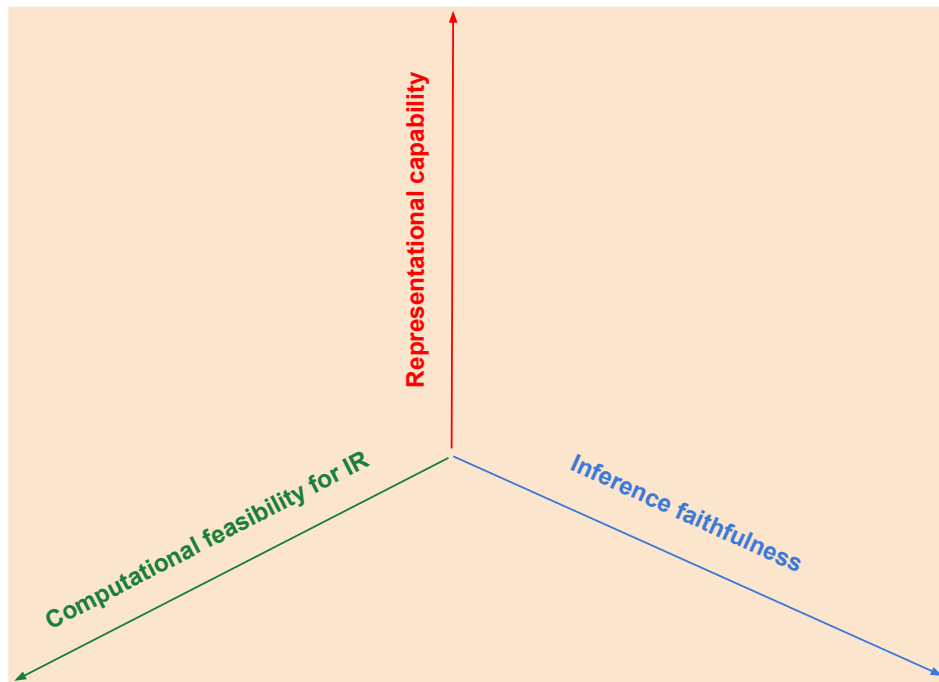
RL-training LLMs for reasoning

Neuro-symbolic

Bayesian Reasoning

Enhancing the representation space

Enhancing training data



Our reading: the trade-off map

Method family	Representational capability	Inference faithfulness	Computational feasibility
Inference-time LLM reasoning (CoT, ToT, ReAct)	○ implicit; no explicit logic or uncertainty	○ traces exist but can be unfaithful	● pay per query, grows with chain length
RL-trained LRMs & search agents (Search-R1, Rank-R1)	○ implicit	● sharper but narrower (pass@k); mainly outcome-only reward	● heavy training, inference cost varied
Multi-agent (-component) systems	● specialized roles, richer decomposition	○ distributed trace — harder to verify	○ coordination & token overhead
Neuro-symbolic (e.g. OrLog)	● explicit logic + probabilities	● solver steps are auditable	● cheap solver; but LLM parses & scoring still expensive
Bayesian / probabilistic (this section)	● explicit, calibrated uncertainty	● posterior is inspectable	● Varied. Can be costly in sampling methods.
Enhanced representations (LogiCoL, ReasonIR)	● set logic / reasoning trained into vectors	○ Performance and generalizability debatable	● dot-product retrieval at corpus scale

● strong ● partial ○ weak (our synthesis: argue with us, that's the point of this slide)

No paradigm wins all three axes nor the 12 types taxonomy.

LLMs: Generalizable, difficult to verify reasoning traces.

Neuro-Symbolic methods: better reasoning guarantees but very low generalizability.

Bayesian methods: can potentially alleviate calibration.

Enhanced representations; easy to scale, but difficult to achieve high performance with retrievers alone.

Verification Current deployed system can not certify their reasoning trace. What would a TREC-style track for inference faithfulness measure?

Performance vs cost BRIGHT went $18 \rightarrow 30 \rightarrow \sim 67$ nDCG@10 in two years, mostly by spending more test-time compute. When is a reasoning-intensive query worth its token bill? (adaptive escalation, retrieve-when-uncertain)

Coverage Abduction, defeasible and dialectical reasoning — the taxonomy rows IR never tests. How can we build such benchmarks?

The bitter-lesson question Will trained retrievers simply absorb reasoning into the representation, making explicit structure unnecessary? or is structure exactly what scaling can't achieve?

Multi-aspect benchmarks Not just correctness of final answer but other aspects, e.g., cost, reasoning faithfulness, various reasoning types, generalizability.

References

Section 2

- 1) Alhamoud, K. et al., 2025. NegBench: Vision-language models do not understand negation. CVPR.
- 2) Bender, E., et al. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? FAccT.
- 3) Chen et al., 2025. Reasoning Models Don't Always Say What They Think. arXiv.
- 4) Chen, Z., et al., 2025. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. arXiv.
- 5) DeepSeek, 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv.
- 6) Dziri, N., et al., 2023. Faith and Fate: Limits of Transformers on Compositionality. NeurIPS.
- 7) Fodor, J., 2025. Line Goes Up? Inherent Limitations of Benchmarks for Evaluating Large Language Models. arXiv.
- 8) Kalai, A.T., et al., 2025. Why language models hallucinate. arXiv.
- 9) Lightman, et al., 2023. Let's Verify Step by Step. ICLR.
- 10) Liu, et al. 2024. Lost in the Middle: How Language Models Use Long Contexts. TACL.
- 11) Madabushi H., et al., 2025. Neither Stochastic Parroting nor AGI: LLMs Solve Tasks through Context-Directed Extrapolation from Training Data Priors.
- 12) Malaviya, C., et al., 2023. Quest: A retrieval dataset of entity-seeking queries with implicit set operations. ACL.
- 13) Mirzadeh, et al., 2025. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. ICLR.
- 14) OpenAI 2024. Learning to Reason with LLMs. (OpenAI o1 system card)
- 15) Petcu, R., et al., 2025. A comprehensive taxonomy of negation for NLP and neural retrievers. Findings of EMNLP.
- 16) Ravichandrat, A., et al., 2022. CONDAQ: A contrastive reading comprehension dataset for reasoning about negation. EMNLP.
- 17) Schaeffer et al., 2023. Are Emergent Abilities of Large Language Models a Mirage. NeurIPS.
- 18) Shi F., et al., 2023. Large Language Models Can be Easily Distracted by Irrelevant Context. ICML.
- 19) Su, H., et al., 2025. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. ICLR.
- 20) van den Elsen, C., et al., 2025. Reproducing NevIR: Negation in neural information retrieval. SIGIR.
- 21) Wang et al., 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. ICLR.
- 22) Wang, X., et al., 2024. BIRCO: A benchmark of information retrieval tasks with complex objectives. arXiv.
- 23) Wei et al., 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS.
- 24) Weller, O., et al., 2024. NevIR: Negation in neural information retrieval. EACL.
- 25) Weller, O., et al., 2025. FollowIR: Evaluating and teaching information retrieval models to follow instructions. NAACL.
- 26) Weller, O., et al., 2026. On the theoretical limitations of embedding-based retrieval. ICLR.
- 27) Yao et al., 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. NeurIPS.
- 28) Zhang, H., et al., 2024. A Careful Examination of Large Language Model Performance on Grade School Arithmetic (GSM1K). arXiv.
- 29) Zhang, W., et al., 2024. ExclulR: Exclusionary neural information retrieval. AAAI.
- 30) Zhou et al., 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. ICLR.

Section 3

- 1) Merrill & Sabharwal. The Parallelism Tradeoff: Limitations of Log-Precision Transformers. TACL 2023.
- 2) Feng et al. Towards Revealing the Mystery behind Chain of Thought: A Theoretical Perspective. NeurIPS 2023.
- 3) Wei et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NIPS 2022.
- 4) Ruis et al. Procedural Knowledge in Pretraining Drives Reasoning in Large Language Models. ICLR 2025.
- 5) Wang & Zhou. Chain-of-Thought Reasoning without Prompting. NeurIPS 2024.
- 6) Madaan, et al. Self-refine: Iterative refinement with self-feedback. NeurIPS 2023.
- 7) Yao et al. React: Synergizing reasoning and acting in language models. ICLR 2023.
- 8) Hoveyda et al., Adaptive Orchestration of Modular Generative Information Access Systems. SIGIR 2025.
- 9) Training language models to follow instructions with human feedback. Ouyang et al., NeurIPS 2022.
- 10) DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. Shao et al., 2024.
- 11) DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. Guo et al., 2025.
- 12) Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. Jin et al., COLM 2025.
- 13) Rank-R1: Enhancing Reasoning in LLM-based Document Rerankers via Reinforcement Learning. Zhuang et al., 2025.
- 14) Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? Yue et al., NeurIPS 2025.
- 15) GPTSwarm: Language Agents as Optimizable Graphs. Zhuge et al., ICML 2024.
- 16) Adaptive Orchestration of Modular Generative Information Access Systems. Hoveyda et al., SIGIR 2025.
- 17) Multi-agent Architecture Search via Agentic Supernet. Zhang et al., ICML 2025.
- 18) Why Do Multi-Agent LLM Systems Fail? Cemri et al., 2025.

Section 4

Neuro-Symbolic:

- 1) A. d'Avila Garcez et al., "Neural-Symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning", FLAP 6(4): 611-632 (2019)
- 2) L.G. Valiant, "Three problems in computer science." J. ACM, 50(1):96–99, 2003.
- 3) Pan et al., "Unifying Large Language Models and Knowledge Graphs: A Roadmap", IEEE TKDE 36(7):3580-3599, 2024.
- 4) C.J. van Rijsbergen, "A new theoretical framework for information retrieval", in SIGIR '86.
- 5) C.J. van Rijsbergen, "A non-classical logic for information retrieval", The Computer Journal, 29(6):481-485, 1986.
- 6) C.J. van Rijsbergen, "Towards an information logic", in SIGIR '89.
- 7) Y. Chiaramella and J.P. Chevallet, "About Retrieval Models and Logic", The Computer Journal, 35(3), 1992.
- 8) T.W.C. Huibers, M. Lalmas and C.J. van Rijsbergen, "Information Retrieval and Situation Theory", SIGIR Forum 30(1):11-25, 1996.
- 9) N. Fuhr, "Probabilistic datalog: Implementing logical information retrieval for advanced applications", J. Am. Soc. Inf. Sci., 51: 95-110, 2000.
- 10) S. Mizzaro, "How many relevances in information retrieval?", Interacting with Computers, 10(3):303-320, 1998.
- 11) Ruthven, Lalmas, and Van Rijsbergen, "Empirical investigations on query modification using abductive explanations", in SIGIR '01.
- 12) Olausson et al., "LINC: A neuro-symbolic approach for logical reasoning by combining language models with first-order logic provers", in EMNLP 2023.
- 13) Hoveyda et al., "OrLog: Resolving Complex Queries with LLMs and Probabilistic Reasoning", in ECIR 2026.
- 14) Nafar et al., "Reasoning over Uncertain Text by Generative Large Language Models", in AAAI 2025.
- 15) R.W. White, "Advancing the Search Frontier with AI Agents". Commun. ACM 67(9):54–65, 2024.

Section 4

Bayesian:

- 1) Hoffman et al., "Training Chain-of-Thought via Latent-Variable Inference", NeurIPS 2023
- 2) Tonolini et al., "Bayesian Prompt Ensembles: Model Uncertainty Estimation for Black-Box Large Language Models", Findings of ACL 2024
- 3) Yin et al., "Reasoning in Flux: Enhancing Large Language Models Reasoning through Uncertainty-aware Adaptive Guidance", ACL 2024
- 4) Wang et al., "BLoB: Bayesian Low-Rank Adaptation by Backpropagation for Large Language Models", NeurIPS 2024
- 5) Agarwal et al., "AutoDiscovery: Open-ended Scientific Discovery via Bayesian Surprise", NeurIPS 2025
- 6) Feng et al., "BIRD: A Trustworthy Bayesian Inference Framework for Large Language Models", ICLR 2025
- 7) Qiu et al., "Bayesian teaching enables probabilistic reasoning in large language models", Nature Comms 2026

Remaining:

- 8) LogiCoL: Logically-Informed Contrastive Learning for Set-based Dense Retrieval. Shen et al., ACL 2025.
- 9) Constructing Set-Compositional and Negated Representations for First-Stage Ranking. Krasakis et al., CIKM 2025.
- 10) ReasonIR: Training Retrievers for Reasoning Tasks. Shao et al., 2025.