

# CS 546 – Advanced Topics in NLP

Dilek Hakkani-Tür



UNIVERSITY OF  
**ILLINOIS**  
URBANA-CHAMPAIGN



Siebel School of  
Computing  
and Data Science



# Topics for Today

## Reasoning

- Introduction and facets of reasoning
- Before LLM Reasoning and CoT
- Prompting for Reasoning
- Single Agent
- Multi-Agent
- Reasoning Evaluation
- Training of reasoning models
- Reasoning Efficiency

# Readings



- Huang and Chang. Towards Reasoning in Large Language Models: A Survey. ACL Findings, 2023.
- Compilation of related work here:  
<https://github.com/atfortes/Awesome-LLM-Reasoning?tab=readme-ov-file>

# Introduction



- Some user requests are simple (e.g., *set a timer for 3 minutes*) and some are more complex, requiring analytical thinking (e.g., Math problems, schedule a reading group meeting).
- Wikipedia Definition: Reasoning is the capacity of consciously applying logic by drawing valid conclusions from new or existing information, with the aim of seeking truth
- Can LLMs think and reason?
- Can LLMs generate better responses if they can think and reason? (meta-generation, accd. to <https://arxiv.org/pdf/2406.16838.pdf>).
  - $x$ : input,  $y$ : output,  $z$ : reasoning steps
  - $x \rightarrow y$  versus  $x \rightarrow z \rightarrow y$
  - $P(y|x)$  versus  $P(z|x)$  and then  $P(y|z,x)$
- What could  $z$  be?

# Facets of Reasoning



**Types of mental operations** used when making sense of information, drawing conclusions, and solving problems.

**Deductive Reasoning:** Drawing **conclusions** from general rules or **premises**.

- Example: All animals are mortal. All cats are animals. All cats are mortal.

**Inductive Reasoning:** Inferring **conclusions** from specific **observations**.

- Example: The sun sets every day. The sun will set today.

**Abductive Reasoning:** Deriving an explanation given a set of observations.

- Example: The roads are wet in the afternoon. It must have rained earlier.

# Facets of Reasoning (cont.)

**Analogical Reasoning:** Solving new problems using their similarity to previously known problems with solutions.

- Example: Learning to ride a bike is like learning to walk. (i.e., it is rarely forgotten)

**Commonsense/Intuitive Reasoning:** Making logical inferences and judgments based on the everyday knowledge that most people share.

- Example: The stove is on. My hand will burn if I touch it.

...



# Topics for Today

## Reasoning

- Introduction and facets of reasoning

- Before LLM Reasoning and CoT

- Prompting for Reasoning

- Single Agent

- Multi-Agent

- Reasoning Evaluation

- Training of reasoning models

- Reasoning Efficiency

# Symbolic Logical Reasoning

- Relies on rules, first-order logic, and semantic representations.
- E.g., An extended model of natural logic for natural language inference ([MacCartney and Manning, ICCS 2009](#))
  - Premise,  $p$ : Every firm polled saw costs grow more than expected, even after adjusting for inflation.
  - Hypothesis,  $h$ : Every big company in the poll reported cost increases.
- Can  $h$  be inferred from the given  $p$ ?
  - Decomposes the inference problem into a sequence of atomic edits, linking  $p$  to  $h$
  - Predicts a lexical semantic relation for each edit;
  - Propagates these relations upward through semantic composition and
  - Joins the resulting semantic relations across the edit sequence to determine inference.
- Examples of semantic relations:

symbol <sup>5</sup>	name	example	set theoretic definition <sup>6</sup>
$x \equiv y$	equivalence	<i>couch</i> $\equiv$ <i>sofa</i>	$x = y$
$x \sqsubset y$	forward entailment	<i>crow</i> $\sqsubset$ <i>bird</i>	$x \subset y$
$x \sqsupset y$	reverse entailment	<i>European</i> $\sqsupset$ <i>French</i>	$x \supset y$
$x \wedge y$	negation	<i>human</i> $\wedge$ <i>nonhuman</i>	$x \cap y = \emptyset \wedge x \cup y = U$

# Symbolic Logical Reasoning (cont.)



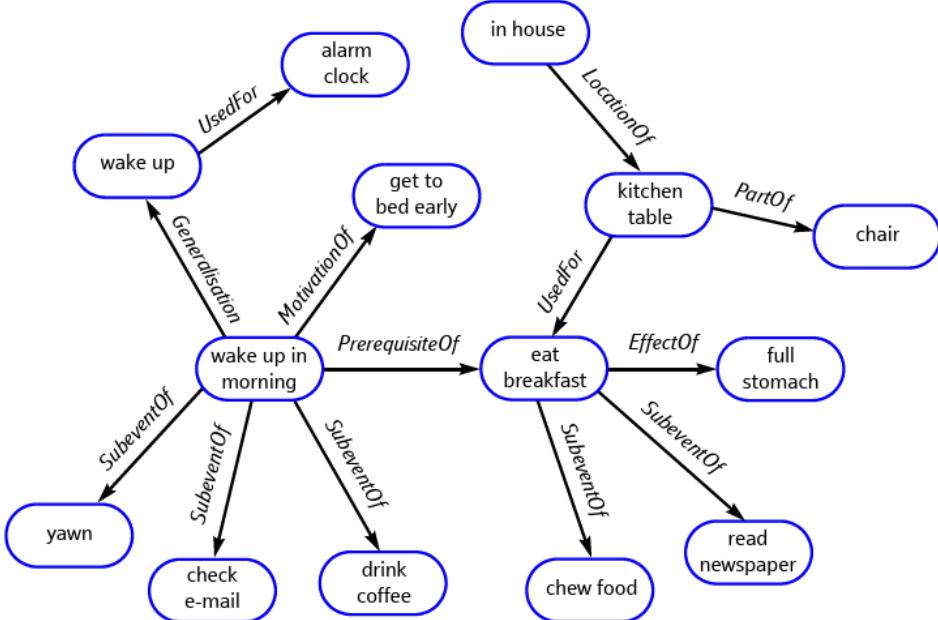
- CYC ([Lenat et al, Communications of ACM, 1995](#))
- Encode and represent **knowledge** about the world so AI systems can **reason logically** rather than just pattern-match.
- A knowledge base of millions of assertions about concepts and relations organized into an ontology.
- Example: All animals are mortal. All cats are animals. All cats are mortal.  
 $\forall X: (\text{isa}(X, \text{Animal}) \Rightarrow \text{isa}(X, \text{MortalBeing}))$   
 $\forall X: (\text{isa}(X, \text{Cat}) \Rightarrow \text{isa}(X, \text{Animal}))$
- Transitivity rule

```
;; Premises
(#$genls #$Animal #$MortalBeing)
(#$genls #$Cat #$Animal)

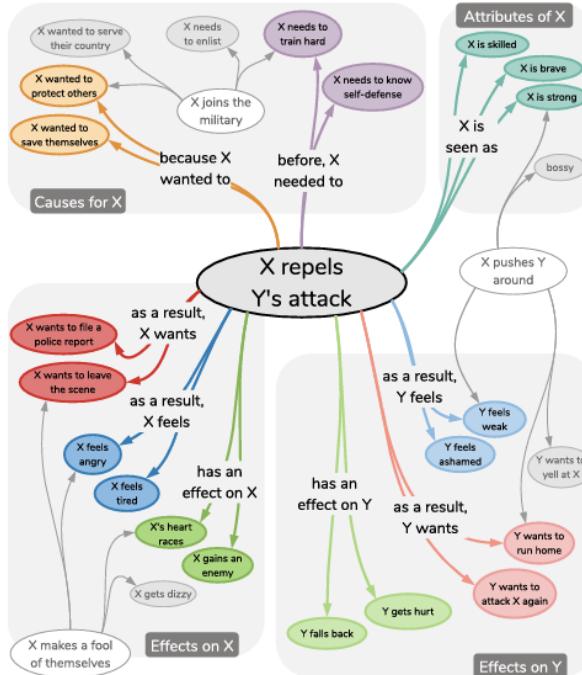
;; Inference rule (built into Cyc ontology):
(forAll (?A ?B ?C)
  (implies
    (and (#$genls ?A ?B)
          (#$genls ?B ?C))
    (#$genls ?A ?C)))

;; Derived conclusion
(#$genls #$Cat #$MortalBeing)
```

# Commonsense Reasoning KGs

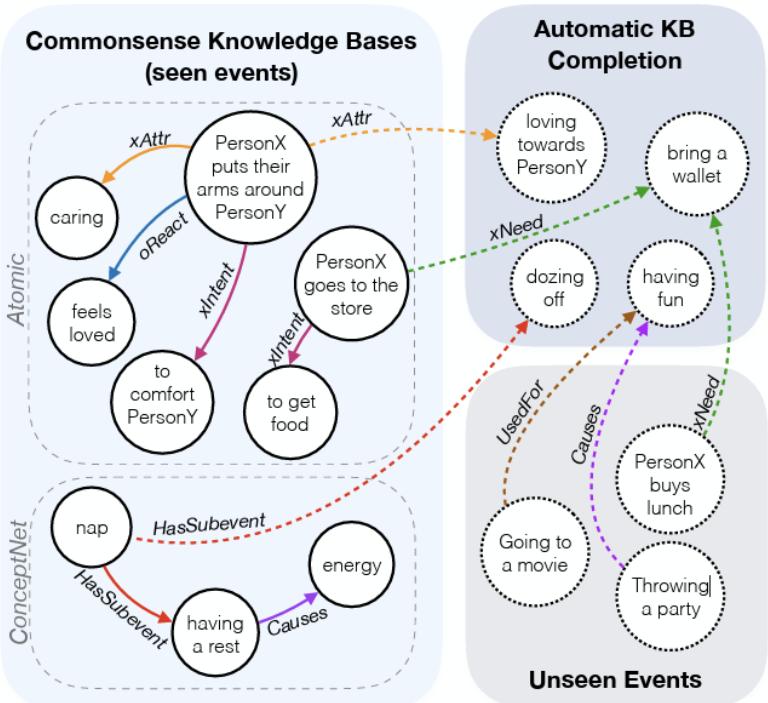


**ConceptNet**, static facts and relations between concepts ([Liu and Singh, BT Technology Journal, 2004](#))



**Atomic**, encodes event-centered knowledge, social and inferential commonsense, focusing on what typically happens before or after events, people's intents, and reactions. ([Sap et al., AAAI 2019](#))

# Commonsense Reasoning KGs (cont.)



- COMET, a neural commonsense reasoning model, trained on knowledge graphs like ATOMIC and ConceptNet to **generate new commonsense inferences** in natural language ([Bosselut et al., ACL 2019](#)).
- Based on GPT, trained to learn to produce the phrase object  $o$  of a knowledge tuple given the tuple's phrase subject  $s$  and relation  $r$ .

ATOMIC Input Template and ConceptNet Relation-only Input Template

$s$ tokens	mask tokens	$r$ token	$o$ tokens
------------	-------------	-----------	------------

PersonX goes to the mall [MASK] < $x$ Intent> to buy clothes

ConceptNet Relation to Language Input Template

$s$ tokens	mask tokens	$r$ tokens	mask tokens	$o$ tokens
------------	-------------	------------	-------------	------------

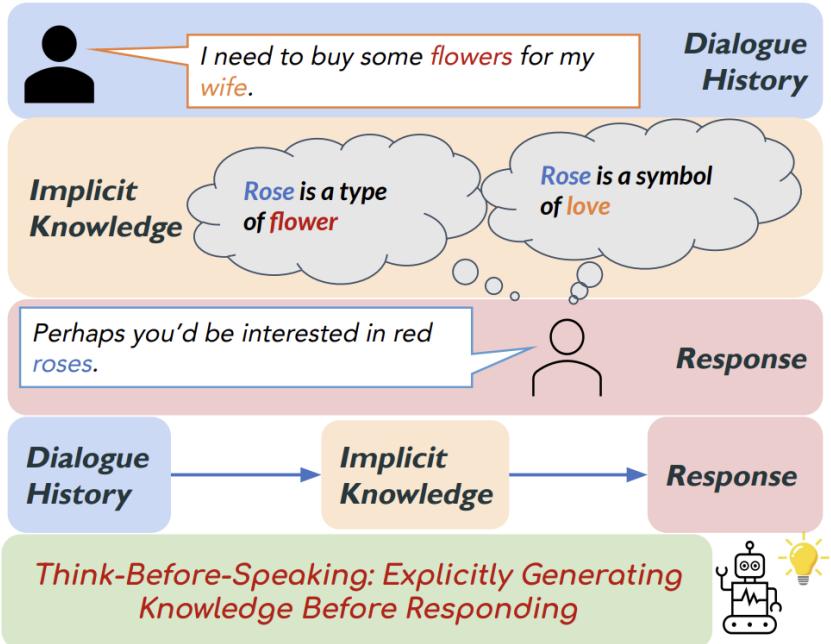
go to mall [MASK] [MASK] has prerequisite [MASK] have money

Figure 1: COMET  learns from an existing knowledge base (solid lines) to be able to generate novel nodes and edges (dashed lines).

# Think Before You Speak



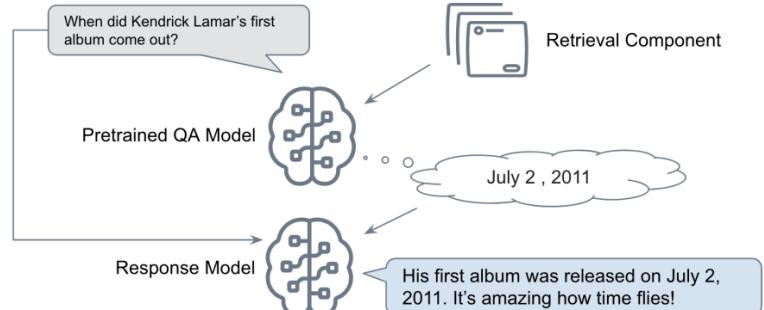
- ([Zhou et al., ACL 2022](#))
- Can LLMs learn to generate explicit commonsense reasoning steps before generating responses?
  - Extended existing conversational datasets with implicit commonsense reasoning knowledge (<https://github.com/alexa/Commonsense-Dialogues>)
  - Trained a model to generate these steps given a conversation.
- Does such reasoning improve model responses?
  - Performed such reasoning before generating responses.



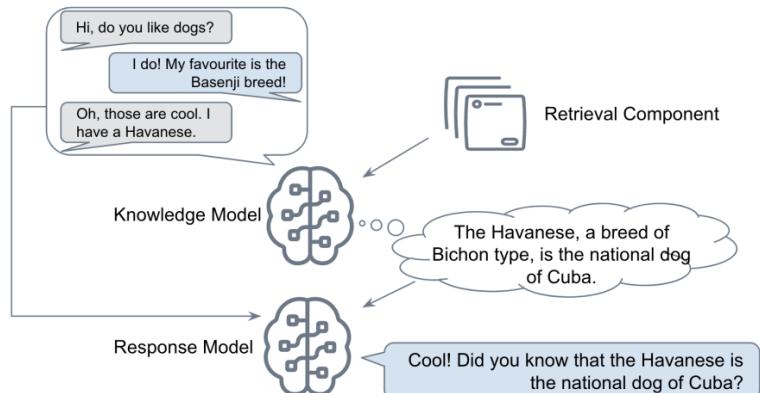
# Reason First, Then Respond



- ([Adolphs et al., EMNLP Findings, 2022](#))
- Similar idea, but with general knowledge.
- A seq2seq knowledge model that maps from context to knowledge.
- A seq2seq response model that generates the final response given the predicted knowledge and the context.



(a)



(b)



# Topics for Today

## Reasoning

- Introduction and facets of reasoning
  - Before LLM Reasoning and CoT
  - Prompting for Reasoning
  - Single Agent
  - Multi-Agent
- 
- Reasoning Evaluation
  - Training of reasoning models
  - Reasoning Efficiency

# Chain-of-Thought (CoT) Prompting

Slide from  
Prompting  
Lecture



In-context examples

## 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. X

## 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. ✓

# Reasoning with Prompting – Single Agent

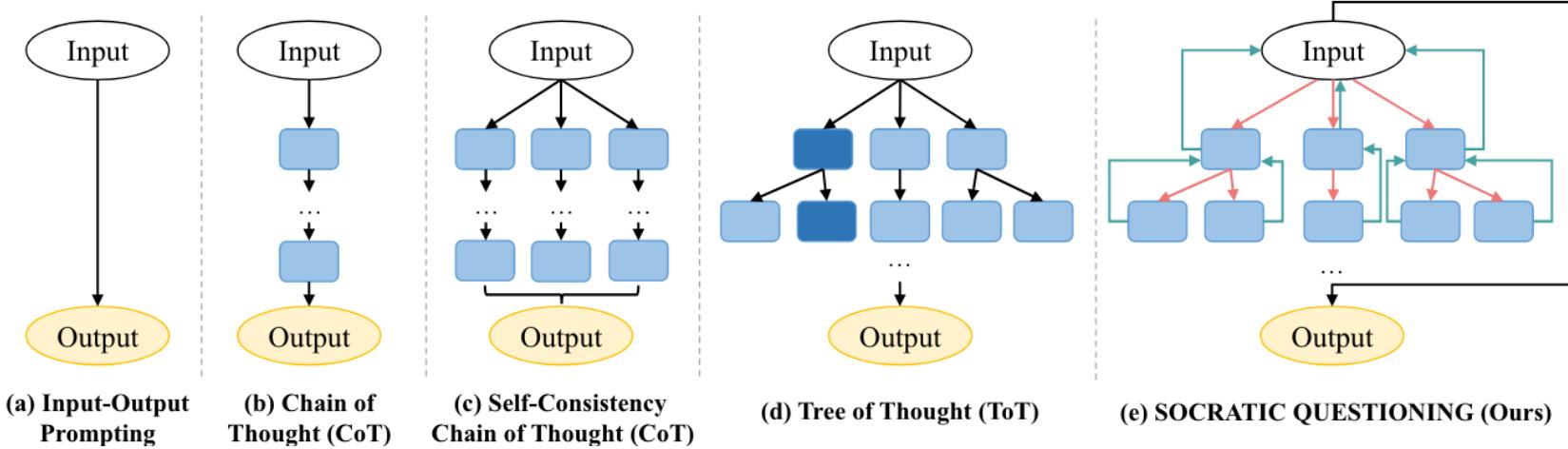
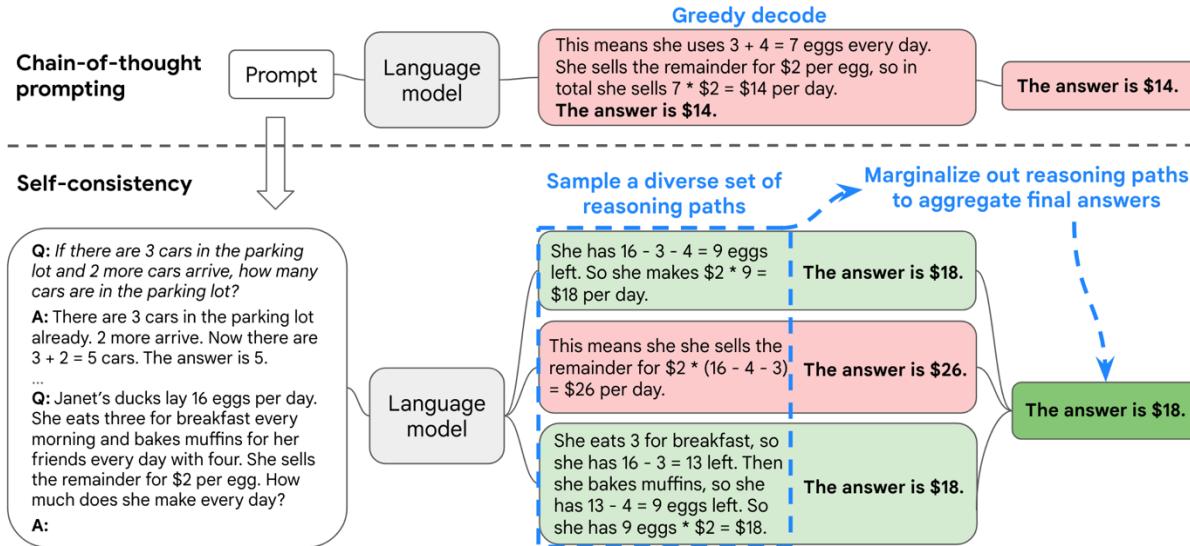


Figure from: ([Qi et al., EMNLP 2023](#))

# Self Consistency CoT



- ([Wang et al., ICLR 2023](#))
- Samples a diverse set of reasoning paths
- From the options, selects the most consistent answer by marginalizing out the sampled paths.



# Self Consistency CoT (cont.)

- $a_i \rightarrow i^{\text{th}}$  model answer,  $i=1,\dots,m$ .
- $r_i \rightarrow$  reasoning steps corresponding to the  $i^{\text{th}}$  model answer.
- Sample multiple  $(r_i, a_i)$  from the model's decoder
- Marginalize over  $r_i$  by taking a majority vote over  $a_i$ :

$$\arg \max_a \sum_{i=1}^m \mathbb{1}(\mathbf{a}_i = a)$$

- Other answer aggregation strategies:
  - Weight each  $(r_i, a_i)$  by  $P(r_i, a_i | \text{prompt, question})$  and sum

$$P(\mathbf{r}_i, \mathbf{a}_i | \text{prompt, question}) = \exp^{\frac{1}{K} \sum_{k=1}^K \log P(t_k | \text{prompt, question}, t_1, \dots, t_{k-1})}$$

- Normalize probability by output length
- Weighted average

# Self-Refine: Iterative Refinement with Self-Feedback

- ([Madaan et al., NeurIPS 2023](#))
- An LLM generates an initial output.
- Then, the same LLM provides feedback for its output.
- Then, the same LLM uses the feedback to refine itself, iteratively.

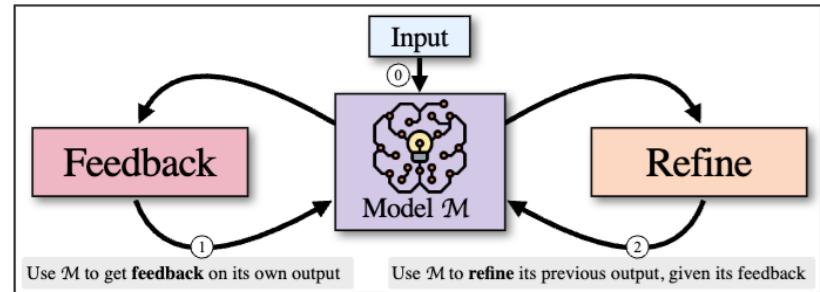


Figure 1: Given an input (0), SELF-REFINE starts by generating an output and passing it back to the same model  $\mathcal{M}$  to get feedback (1). The feedback is passed back to  $\mathcal{M}$ , which refines the previously generated output (2). Steps (1) and (2) iterate until a stopping condition is met. SELF-REFINE is instantiated with a language model such as GPT-3.5 and does not involve human assistance.

# Generating Sequence by Learning to Self-Correct

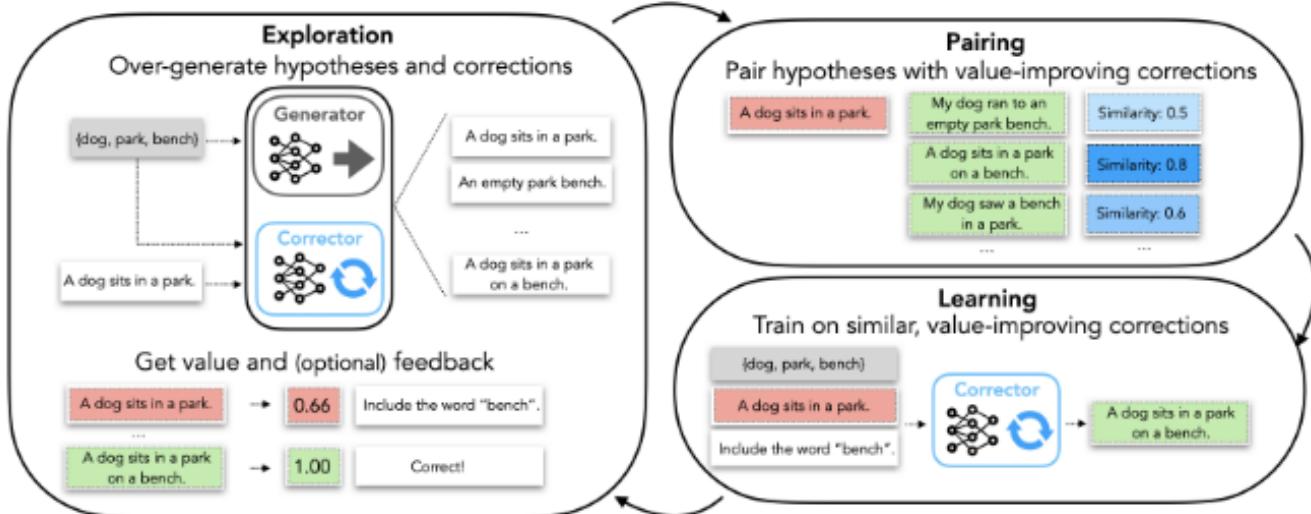


- ([Welleck et al., ICLR 2023](#))
- LLMs often meet most task requirements, but can miss a few, and need to start again from scratch.
- A more natural, intuitive approach could be leveraging the generation as a starting point and refining it into higher quality output:
  - o *Generator* produces a reasonable initial hypothesis
  - o *Corrector* is trained to make up the difference between the initial hypothesis and the optimal solution.
  - o The corrector can be applied multiple times.



# Self-Correct (cont.)

- Learning a Corrector:
  - Exploration
  - Pairing
  - Learning
  - Re-Exploration



# Tree of Thoughts (ToT)

- ([Yao et al., NeurIPS 2023](#))
- Autoregressive LLMs are still restricted to token-level, left to right decision-making processes.
  - Falls short in tasks that require exploration, lookahead etc.
  - Problem of “Hallucination Snowballing” ([Zhang et al., ICML 2024](#))
- Existing methods do not explore different continuations of a thought process
- We may benefit from a more deliberate "System 2" planning process that:
  - Maintains and explores diverse alternatives
  - Evaluates its current status and actively looks ahead or backtracks to make more global decisions.

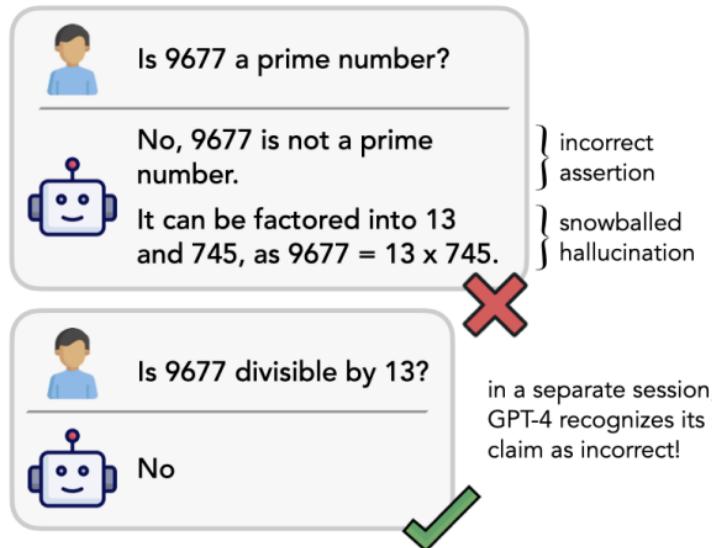
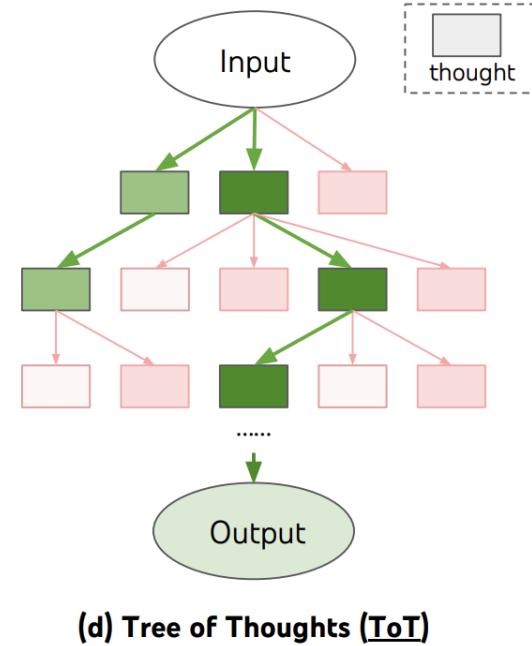


Figure from ([Zhang et al., ICML 2024](#))

# ToT (cont.)

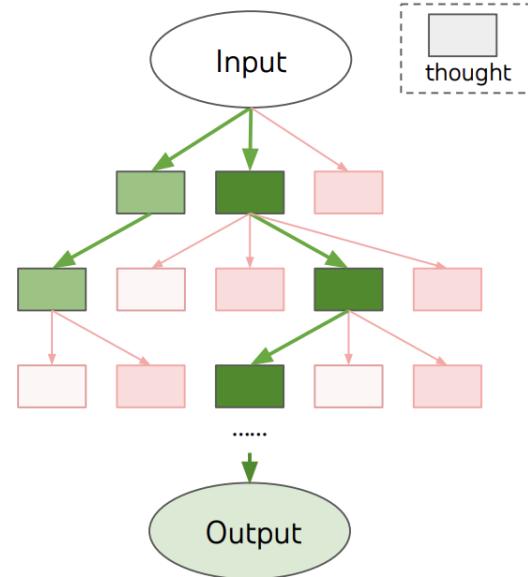


- Allows LMs to explore multiple reasoning paths over thoughts
- Frames problems as search over a tree
- Each node is a state  $s = [x, z_{1..i}]$  representing a partial solution
- 4 Questions:
  1. How to decompose intermediate process into thought steps?
  2. How to generate potential thoughts from each stage?
  3. How to heuristically evaluate states?
  4. What search algorithm to use?



# ToT Steps

- **Thought Decomposition:** Splits problem into intermediate steps or thoughts depending on the task.
- **Thought Generator:** Given a state, sample or propose thoughts depending on the task (two different prompts).
- **State Evaluator:** Evaluates progress a state makes towards solving a problem.
  - Which states do we keep exploring?
  - What order to explore in?
- **Search:** Breadth-first versus depth-first



(d) Tree of Thoughts (ToT)

# Socratic Questioning: Recursive Thinking with LLMs



- ([Qi et al., EMNLP 2023](#))
- CoT is confined by its single-pass and sequential generation process, and errors accumulate (again, the snowballing effect).
- CoT also lacks the ability to refine an already generated reasoning path.
- Humans adopt recursive thinking when tackling complex reasoning problems:
  - o Iteratively break the original problem into approachable subproblems and
  - o Aggregating their answers to resolve the original one.
- SOCRATIC QUESTIONING, a divide-and-conquer algorithm, mimicking the humans' **recursive** thinking process.

# Socratic Questioning Example



**Question:** A ball slides down a landslide of height  $h$ , the length of the landslide is  $l$ , ball mass is  $m$ , and the angle between the landslide and the ground is  $\theta$ . Find the velocity  $v$  of the ball at the landslide bottom. Ignore friction force and air resistance.

## Standard Prompting

X  $v = \sqrt{2 * g * h * \sin(\theta) / (1 + (l * \sin(\theta)) / (2 * h))}$



## Chain-of-Thought (CoT)

The kinetic energy theorem states that the net work done on an object equals its change in kinetic energy.



In this case, the net work done is done by the frictional force. The work done by friction can be calculated as  $W = f * l$ , where  $l$  is the landslide length and  $f = \mu * N$ . Write the equation as:  $\mu * m * g * \cos(\theta) * l = (1/2) * m * v^2$ . Solve that equation,  $v = \sqrt{(2 * \mu * g * \cos(\theta) * l) / 1}$ .



## Socratic Questioning

I do not know.



1. Does this problem obey the energy conservation law?
2. What is the mechanical energy of the initial state?
3. What is the mechanical energy in the final state?

1. Because there is no external force doing work, so this question follows energy conservation law.
2. At beginning, ball's velocity is 0, so its kinetic energy is 0. And the potential energy is  $mg * h$ .
3. At the end, ball's height is 0, so the potential energy is 0. And kinetic energy is  $1/2 * m * v^2$ .



Again, find the velocity  $V$  of the ball at the bottom of the landslide. Ignore friction force.

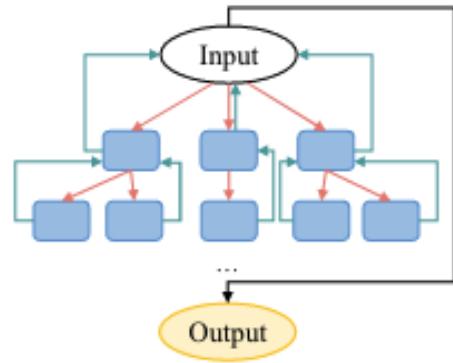
Solve  $0 + mg * h = 1/2 * m * v^2 + 0$ , we get  
✓  $v = \sqrt{2 * g * h}$ .



# Socratic Questioning Approach



- **Top-down exploration process (in red):** the original complex problem is decomposed into simpler or related sub-problems until the sub-problems can be solved
- **Bottom-up backtracking process (in green):** the solutions to the sub-problems are returned and selectively used to solve the original problem.
- SELF-QUESTIONING proactively raises and answers questions that are essential to the target question.
- SOCRATIC QUESTIONING recursively backtracks and tailors the intermediate thoughts acquired from SELF-QUESTIONING until reaching an answer to the original input question.



(e) SOCRATIC QUESTIONING (Ours)

# Reasoning with Prompting – Multi-Agent

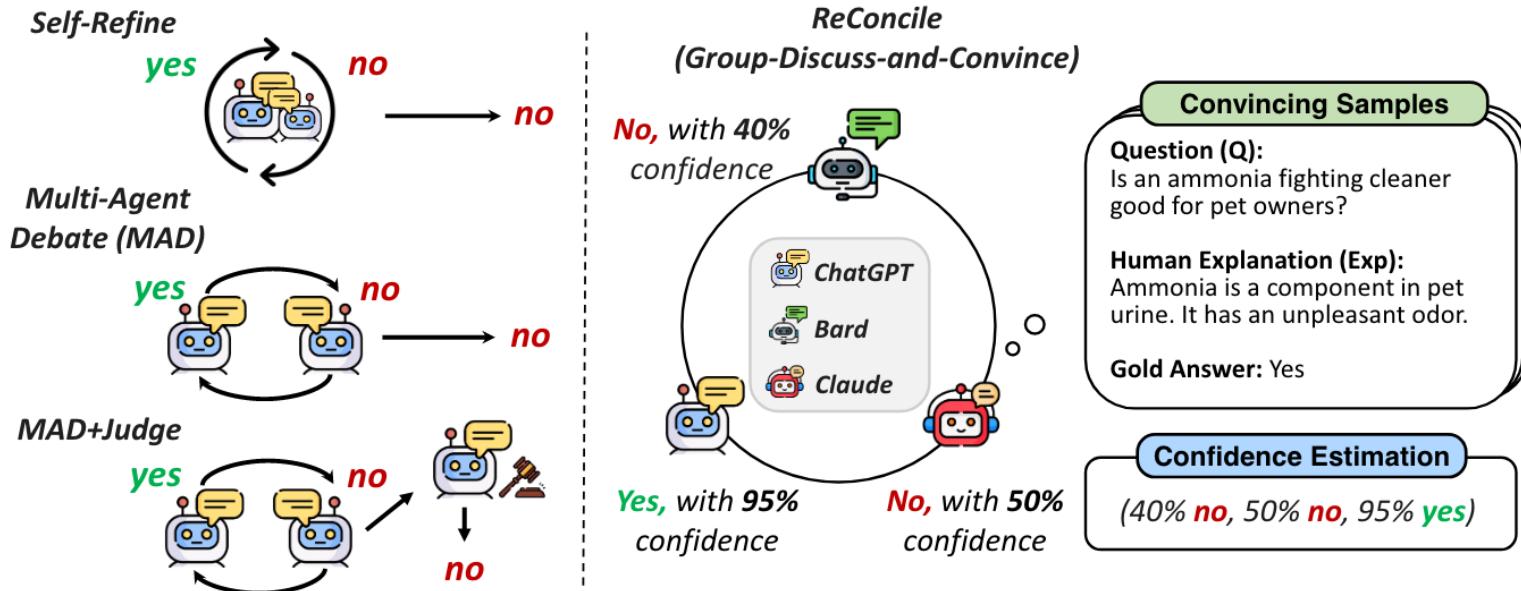


Figure from: ([Qi et al., EMNLP 2023](#))  
Right figure from:

# Multi-agent Debate (MAD)



- ([Du et al., Preprint, 2023](#))
- Inspiration from "Society of Mind" (Minsky, 1988): A conceptual framework for how complex intelligent behavior can emerge from the interactions of many simple components (Minsky called these *agents*), rather than a central controller.
- Given a query, multiple instances of a language model generate individual candidate answers to a query.
- Then each individual model instance reads and critiques the responses of all other models and uses this content to update its own answer.  
→ repeated over several rounds.

# MAD Example

**User Input:** A treasure hunter found a buried treasure chest filled with gems. There were 175 diamonds, 35 fewer rubies than diamonds, and twice the number of emeralds than the rubies. How many of the gems were there in the chest?

## Round 1

**Agent 1:** Let  $x$  be the number of rubies. Then the number of diamonds is 175. And the number of emeralds is  $2(x-35)$ , since there are twice as many emeralds as rubies and there are 35 fewer rubies than diamonds.

To find the total number of gems, we add the number of diamonds, rubies, and emeralds: Total gems =  $175 + x + 2(x-35) = 3x + 105$ .

We don't know the exact value of  $x$ , but we do know that it represents the number of rubies, which is less than 175 (since there are 35 fewer rubies than diamonds). Let's try some values of  $x$ : If  $x = 50$ , then the total gems =  $3(50) + 105 = 255$ . If  $x = 100$ , then the total gems =  $3(100) + 105 = 405$ .

Since we know there were fewer rubies than diamonds, the answer must be closer to 255 than to 405. After some trial and error, we can find that  $x = 40$  gives us: Total gems =  $3(40) + 105 = 225$ .



**Agent 2:** If there were 175 diamonds, then the number of rubies is 35 fewer, or  $175 - 35 = 140$  rubies.

And since there were twice as many emeralds as rubies, we can represent the number of emeralds as  $2 * 140 = 280$  emeralds.

To find the total number of gems in the chest, we can add up the number of diamonds, rubies, and emeralds:  
 $175 + 140 + 280 = 595$ . Therefore, there were 595 gems in the chest.



**User Input:** Using the solutions from other agents as additional information, can you give an updated response....

(Copy Agent 2's reply)

(Copy Agent 1's reply)

# MAD Example (cont.)

## Round 2

**Agent 1:** Given the information provided in other agents' solutions, we have two answers: One agent found that the number of rubies is 40 and the total number of gems is 225. Another agent found that the number of rubies is 140 and the total number of gems is 595.

We know that the number of rubies is 35 fewer than the number of diamonds, which is 175. So the number of rubies should be  $175 - 35 = 140$ , as the second agent found. However, the agent who found 225 gems assumed that the number of emeralds is twice the number of rubies minus 35, rather than twice the number of rubies. If we use this assumption, we get: Number of rubies = 140. Number of emeralds =  $2 * 140 - 35 = 245$ . Total number of gems =  $175 + 140 + 245 = 560$ . Our final answer is 560. 

**User Input:** Using the solutions from other agents as additional information, can you give an updated response....

[\(Copy Agent 2's reply\)](#)



[\(Copy Agent 1's reply\)](#)

## Round 3

**Agent 1:** After reviewing the solutions provided by other agents, I agree with the second agent who found that the number of rubies is 140 and the total number of gems in the chest is 595. Therefore, my answer to the math problem is 595.



**Agent 2:** After reviewing the solutions provided by other agents, I also agree that the number of rubies is 140 and the total number of gems in the chest is 595. Therefore, my answer to the math problem is also 595.



# Multi-agent Debate with Judge



- ([Liang et al., EMNLP, 2024](#))
- Multiple agents express their arguments, and a judge manages the debate process to obtain a final solution.
  - The distorted thinking of one LLM can be corrected by the others
  - The resistance to change of one LLM will be complemented by the others, and
  - Each agent can obtain external feedback from the others.

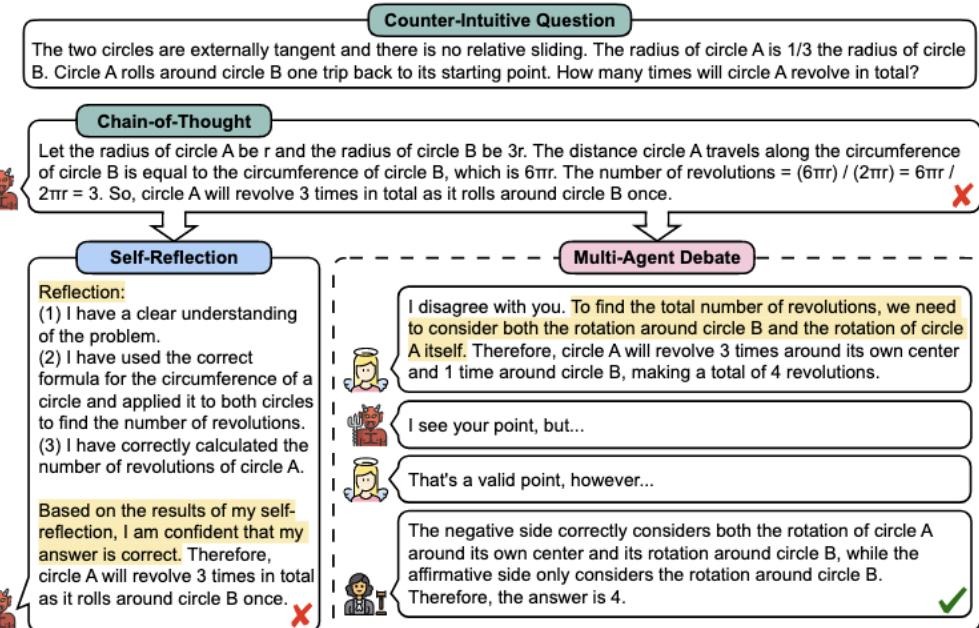


Figure 2: Framework of Multi-Agent Debate. Here we designate the devil (devil icon) as the affirmative side while the angel (angel icon) as the negative side. We want the angel to correct the devil's mistakes.

# MAD with Judge (cont.)

---

**Algorithm 1** MAD: Multi-Agents Debate
 

---

**Require:** Debate topic  $t$ , maximum number of

rounds  $M$  and number of debaters  $N$

**Ensure:** Final answer  $a$

```

1: procedure MAD( $t, M, N$ )
2:    $J$             $\triangleright$  Initialize the judge
3:    $D \leftarrow [D_1, \dots, D_N]$   $\triangleright$  Initialize debaters
4:    $H \leftarrow [t]$        $\triangleright$  Initialize debate history
5:    $m \leftarrow 0$          $\triangleright$  Current round
6:   while  $m \leq M$  do
7:      $m \leftarrow m + 1$ 
8:     for each  $D_i$  in  $D$  do
9:        $h \leftarrow D_i(H)$   $\triangleright$  Generate argument
10:       $H \leftarrow H + [h]$   $\triangleright$  Append  $h$  to  $H$ 
11:      if  $J_d(H)$  then
12:         $\text{break}$            $\triangleright$  Debate is over
13:       $a \leftarrow J_e(H)$      $\triangleright$  Extract the final answer
14:    return  $a$ 

```

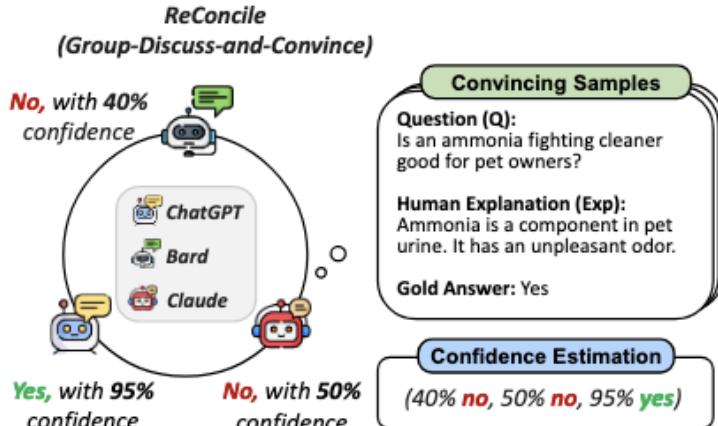
---

- Debaters generate turns in a round-robin fashion.
- Judge decides whether the debate should be over (or max number of rounds is reached)
- Judge makes the final decision.

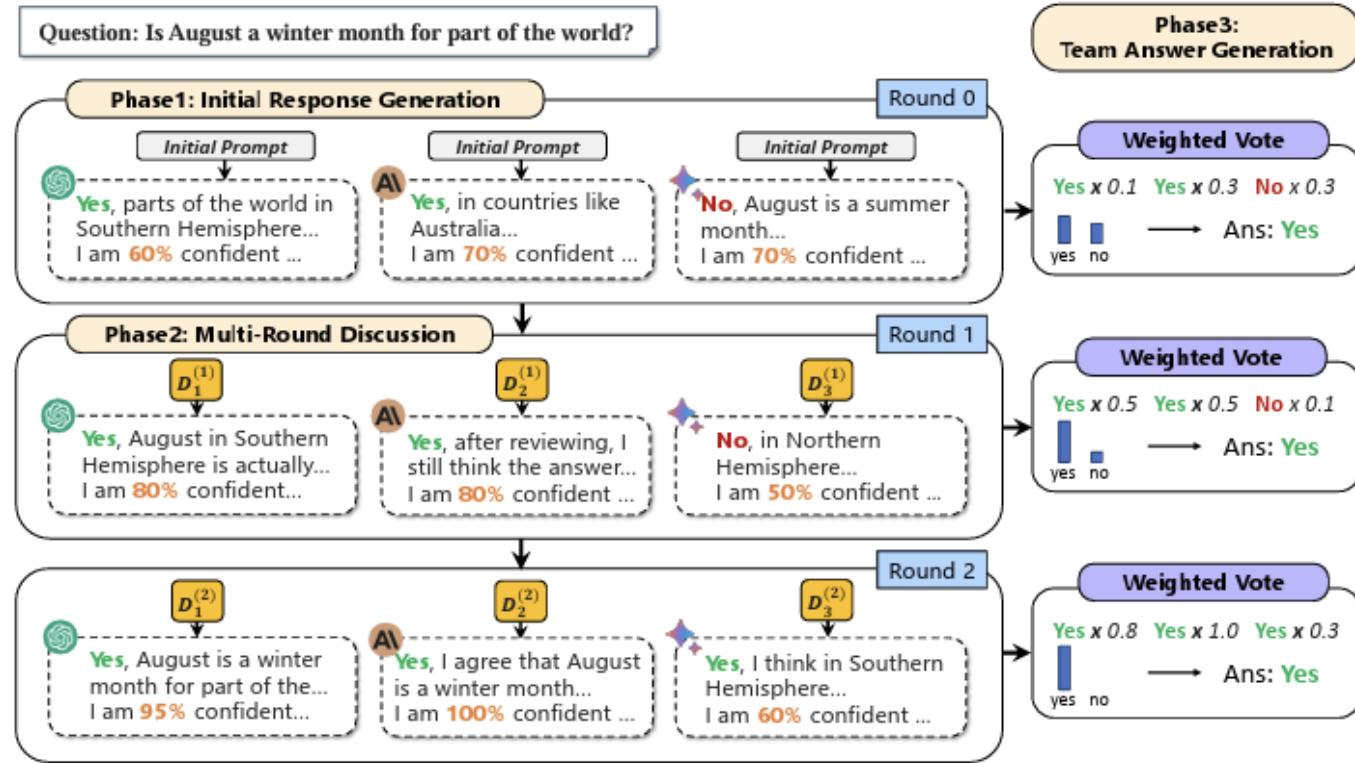
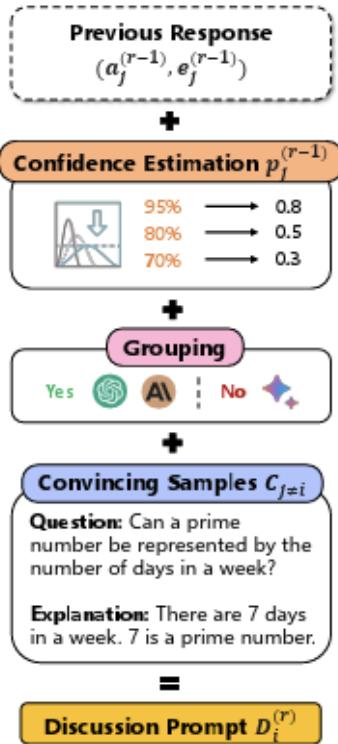
# ReConcile: Group-Discuss-and-Convince



- ([Qi et al., EMNLP 2023](#))
- Uses multiple diverse LLM agents (such as different model families)
  - E.g., ChatGPT, Bard, Claude2.
- Proceeds through multiple rounds of discussion, where each agent provides an answer and explanation based on other agents' prior round outputs.
- Each agent also provides a confidence score.
- A weighted voting / consensus mechanism produces the final answer.
- Aims to converge to a better answer via collaboration, persuasion among agents, rather than strictly adversarial debating.



# ReConcile Example





# Topics for Today

## Reasoning

- Introduction and facets of reasoning
  - Before LLM Reasoning and CoT
  - Prompting for Reasoning
  - Single Agent
  - Multi-Agent
- Reasoning Evaluation
- Training of reasoning models
  - Reasoning Efficiency

# Reasoning Evaluation

- Existing work on reasoning evaluation mainly focused on end-task performance
- Are these the same?



- Finer-grained metrics can help us assess which reasoning steps support the accurate final answers.
- Outcome reward models (ORMs) that determine reward based on the final outcome vs.
- Process reward models (PRMs) that evaluate individual reasoning steps to determine a reward.

- ([Golovneva et al., ICLR 2023](#))
- Aimed at assessing the quality of reasoning chains produced by language models.
- Reference-free metrics: do not require manual ground-truth annotations.
- Unsupervised metrics: do not require training data annotations.

Table 1: Step-by-step reasoning for logical inference task generated from 2-shot GPT-3 (text-davinci-002) ([Brown et al., 2020](#)). Error types are highlighted: logical or factual issues, partially correct, somewhat-relevant. Full 2-shot exemplars are in Appendix B.

**Context:** Shelley is from Virginia, but is visiting that city with that famous market where they throw the fish. Going home next Tuesday!

**Question:** Is it likely that Shelly will be near the Pacific Ocean this weekend?

**GPT-3 Step-by-step reasoning:**

- No, it is not likely that Shelley will be near the Pacific Ocean this weekend.

- Shelley is from Virginia, which is on the East Coast of the United States.

- The Pacific Ocean is on the West Coast.

- Shelley is visiting a city with a famous market, which could be anywhere in the world.

- Tuesday is her travel day, so she will likely not be traveling this weekend.

# ROSCOE (cont.)



- Grouped fine-grained metrics under four perspectives:
  - **semantic alignment:** how well each step in a model's reasoning chain is semantically aligned with what has come before, i.e., the original problem and the preceding reasoning steps.
  - **semantic similarity:** how internally consistent and semantically coherent the steps are with each other, without necessarily conditioning on the task or input.
  - **logical inference:** whether each step in the reasoning chain follows from the previous steps in a logically valid way, i.e., does the inference structure make sense?
  - **language coherence:** the form and fluency of the reasoning chain, i.e., is it composed of well-formed natural language that a human can follow smoothly.
- A set of metrics based on token embedding similarities, perplexity, etc.

- ([Prasad et al., EMNLP 2023](#))
- Measures two key properties (also reference-free):
  - correctness, i.e., each step makes a valid inference based on information contained within the step, preceding steps, and input context
  - Uses NLI models
- informativeness, i.e., each step provides new information that is helpful towards deriving the generated answer
- Uses information theoretic metrics, such as pointwise V-information.

**Context:** The moon is a kind of moon. Earth is a kind of planet. Moons orbit planets. Gravity causes orbits.

**Question:** What keeps the Moon orbiting Earth?

**Model-generated Step-by-Step Rationales:**

- **Step 1:** [Moon is a kind of moon] and [earth is a kind of planet], so [the moon and earth are planets].
- **Step 2:** [Gravity causes orbits], so [gravity causes moon to orbit earth].

**Answer:** Earth's gravity.

Figure 1: Model-generated step-by-step reasoning from Entailment Bank ([Dalvi et al., 2021](#)). Reasoning errors include: incorrect step inference (requires inferring '*moon orbits earth*'), and incorrect inference and uninformative ('*moon is a planet*' does not help answer the question). Reasoning Content Units (RCUs) are shown as '[.]'.

# Premise-Augmented Reasoning Chains (PARC)



- ([Mukherjee et al., ICML 2025](#))
- Prompts an LLM to identify premises and their relationships.

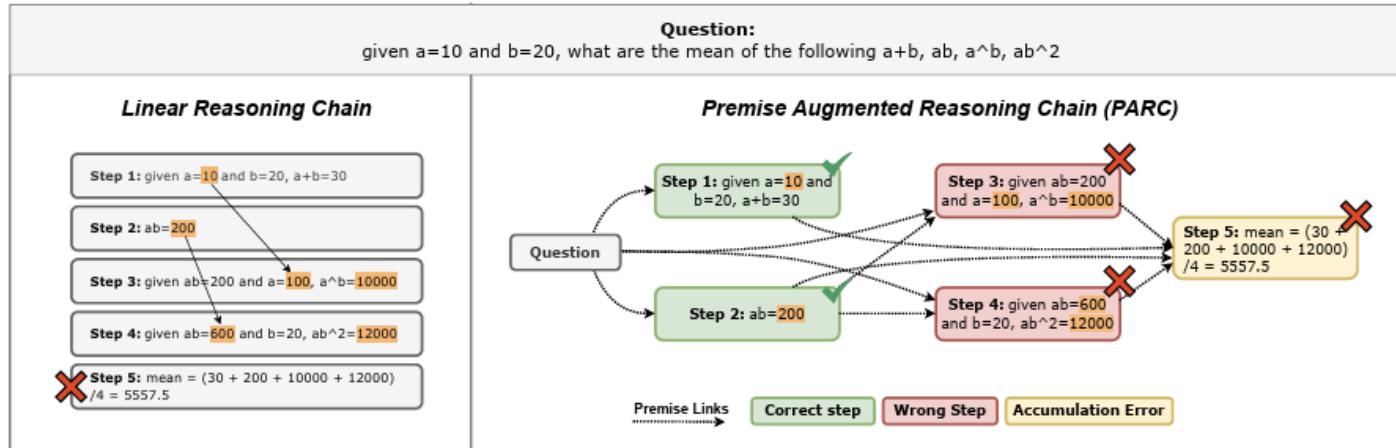


Figure 1. Comparison between a Linear Reasoning Chain (LRC) and our proposed PARC (Premise-Augmented Reasoning Chain). The LRC (left), is linear and there no explicit premise link between steps. In PARC (right), premise links are explicitly established, enabling better identification of correct and incorrect steps. Accumulation errors can be traced back to faulty premises. Establishing these premises helps improve error detection with LLMs.

# PARC (cont.)



- Verifies reasoning steps based on its premises.
- Types of errors:
  - A native error if it contains an inherent discrepancy (e.g., a miscalculation or logical inconsistency)
  - An accumulation error if a given step is logically valid, but at least one of its premises is incorrect.

At constant temperature, the pressure of a sample of gas is inversely proportional to its volume. I have some hydrogen in a 3.67 liter container with a pressure of 4 kPa. If I move all of it to a 1.835 liter container at the same temperature, what will the new pressure be in kPa?

Solution	Baseline Verifier	Our Verifier
<b>Step 1:</b> We can use the formula for inverse proportionality between pressure (P) and volume (V) of a gas at constant temperature, which is given by: $P \propto 1/V$	✓	✓
<b>Step 2:</b> We can rewrite this as an equation by introducing a constant of proportionality (k): $P = k / V$	✓	✓
<b>Step 3:</b> Given that the initial pressure ( $P_1$ ) is 4 kPa and the initial volume ( $V_1$ ) is 3.67 liters, we can write: $P_1 = k / V_1$ , $4 = k / 3.67$	✓	✓
<b>Step 4:</b> To find the value of k, we can multiply both sides by 3.67: $k = 4 * 3.67$ , $k = 14.68$	✓	✓
<b>Step 5:</b> Now that we have the value of k, we can use it to find the new pressure ( $P_2$ ) when the volume ( $V_2$ ) is 1.835 liters: $P_2 = k / V_2$ , $P_2 = 14.68 / 1.835$	✓	✓
<b>Step 6:</b> To find the new pressure, we can divide 14.68 by 1.835: $P_2 = 7.98$ kPa	✓	✗ Mathematical Error
<b>Step 7:</b> Therefore, the new pressure of the hydrogen in the 1.835 liter container will be approximately 7.98 kPa.	✓	✗ Accumulation Error

Figure 2. An example where the baseline method fails to detect errors, while our verification method with established premise links successfully identifies the mathematical error in step 6, and the accumulation error in step 7.



# Topics for Today

## Reasoning

- Introduction and facets of reasoning
- Before LLM Reasoning and CoT
- Prompting for Reasoning
  - Single Agent
  - Multi-Agent
- Reasoning Evaluation
- Training of reasoning models
- Reasoning Efficiency

# System 2 Thinking

- Slow, deliberate, effortful, and conscious mode of reasoning



“Systems 1 and 2 are both active whenever we are awake. System 1 runs automatically and System 2 is normally in a comfortable low-effort mode, in which only a fraction of its capacity is engaged. System 1 continuously generates suggestions for System 2: impressions, intuitions, intentions, and feelings. If endorsed by System 2, impressions and intuitions turn into beliefs, and impulses turn into voluntary actions. When all goes smoothly, which is most of the time, System 2 adopts the suggestions of System 1 with little or no modification. You generally believe your impressions and act on your desires, and that is fine—usually.”

# Training LLMs for effortful reasoning

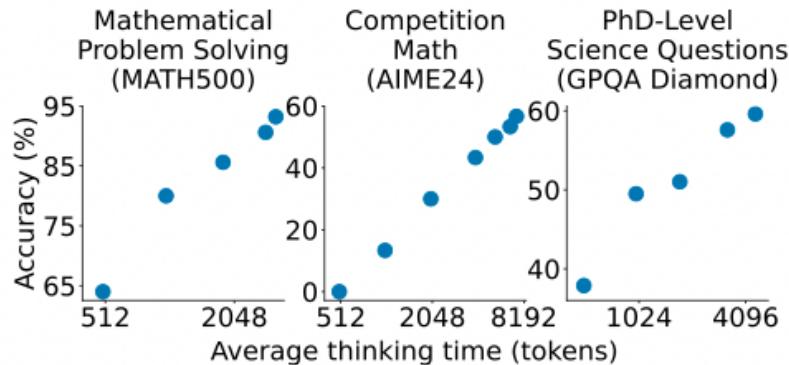


- ([Snell et al., ICLR 2025](#))
- Test-time scaling vs. scaling model parameters
- Can we enable an LM to make use of additional computation at test time to improve the accuracy of its response?
- Two mechanisms:
  - Fine-tune models to iteratively refine their answers in complex-reasoning based settings.
  - Process-reward model (PRM) which produces a prediction of correctness of each step in a solution (rather than only the final answer) and (tree) search over the space of solutions
- A “compute-optimal” scaling strategy, which acts to most effectively allocate test-time compute adaptively per prompt

# S1 - Simple test time scaling



- ([Muennighoff et al., Preprint, 2025](#))
- **S-1K** - 1,000 {question, reasoning trace} pairs relying on: difficulty, diversity, and quality
- **Budget forcing** - control test-time compute by forcefully terminating the model's thinking process or appending "Wait"
- **Recipe**
  - Train model on S1k with SFT
  - Budget force during inference



# S1 - Simple test time scaling



PS: It is an ongoing debate whether thinking more always helps 😊

**OptimalThinkingBench: Evaluating Over and Underthinking in LLMs**

You can go through the papers and form your opinion (it is fun)

---

**Does Thinking More *always* Help?  
Understanding Test-Time Scaling in Reasoning Models**

---

# DeepSeek-R1



- ([DeepSeek AI, Preprint 2025](#))
- DeepSeek-R1-Zero
  - Trains model with RL
  - Rule-based rewards for task completion, formatting.

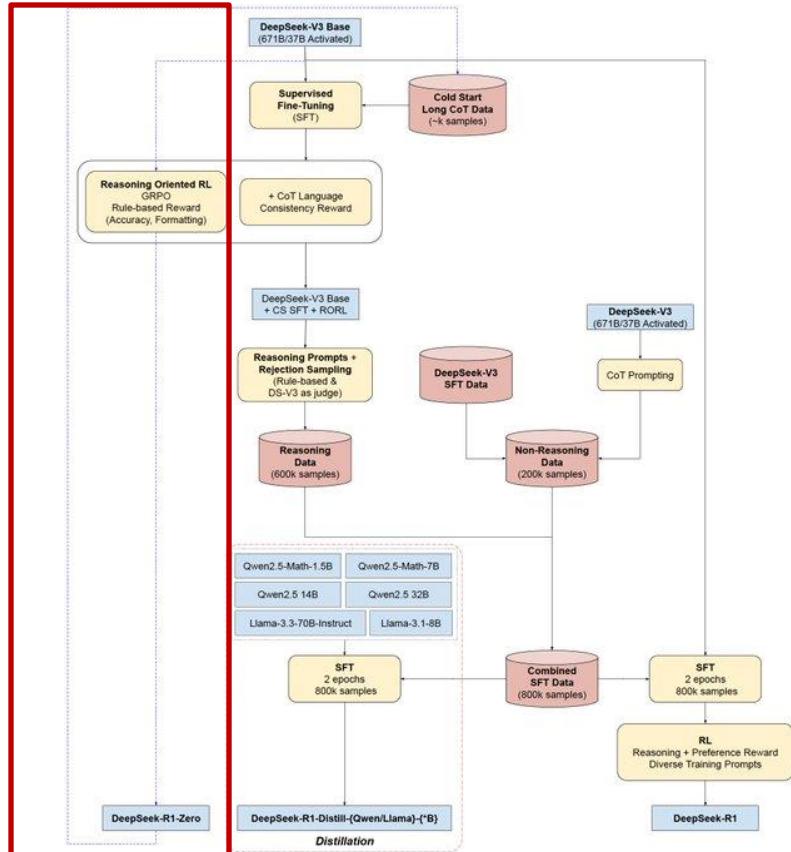


Figure from: <https://x.com/SirrahChan/status/1881540279783887036>

# DeepSeek-R1 (cont.)

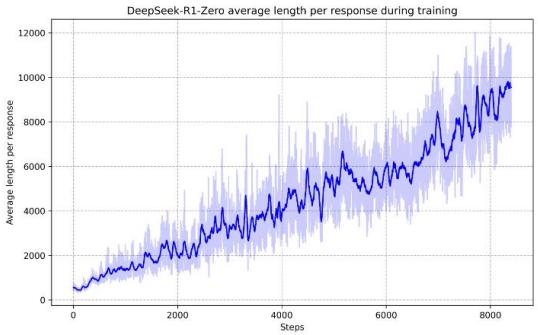


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

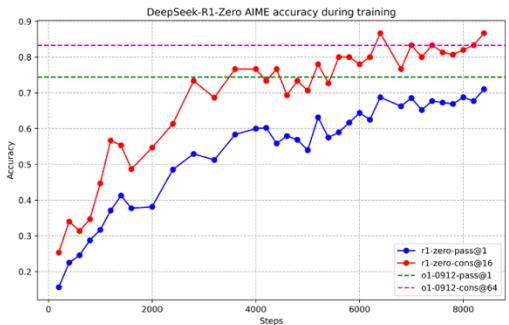


Figure 2 | AIME accuracy of DeepSeek-R1-Zero during training. For each question, we sample 16 responses and calculate the overall average accuracy to ensure a stable evaluation.

# DeepSeek-R1 (cont.)

- Model *discovers* long reasoning patterns they refer to as “*aha moments*”

---

Question: If  $a > 1$ , then the sum of the real solutions of  $\sqrt{a - \sqrt{a+x}} = x$  is equal to

---

Response: <think>

To solve the equation  $\sqrt{a - \sqrt{a+x}} = x$ , let's start by squaring both ...

$$(\sqrt{a - \sqrt{a+x}})^2 = x^2 \implies a - \sqrt{a+x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

...

**Wait, wait. Wait. That's an aha moment I can flag here.**

Let's reevaluate this step-by-step to identify if the correct sum can be ...

We started with the equation:

$$\sqrt{a - \sqrt{a+x}} = x$$

First, let's square both sides:

$$a - \sqrt{a+x} = x^2 \implies \sqrt{a+x} = a - x^2$$

Next, I could square both sides again, treating the equation: ...

...

---

Table 3 | An interesting “aha moment” of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.

# DeepSeek-R1



- Problem: the reasoning chains are not always human readable (e.g., weird tokens)
- Solution: Fine-tune the base model with long reasoning chains and SFT.
- Then apply RL

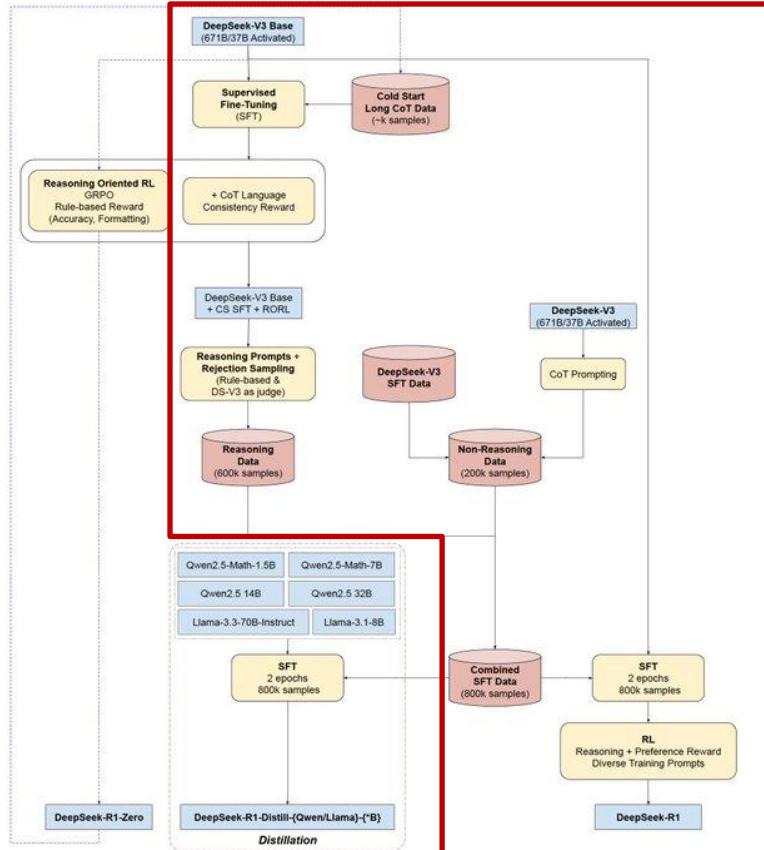


Figure from: <https://x.com/SirrahChan/status/1881540279783887036>

# DeepSeek-R1 (cont.)

- Then distills reasoning behaviors into smaller models through SFT
- Fun extra info: GRPO had a length bias which caused the model to generate longer reasoning chains ([Marjanović et al., Preprint 2025](#))

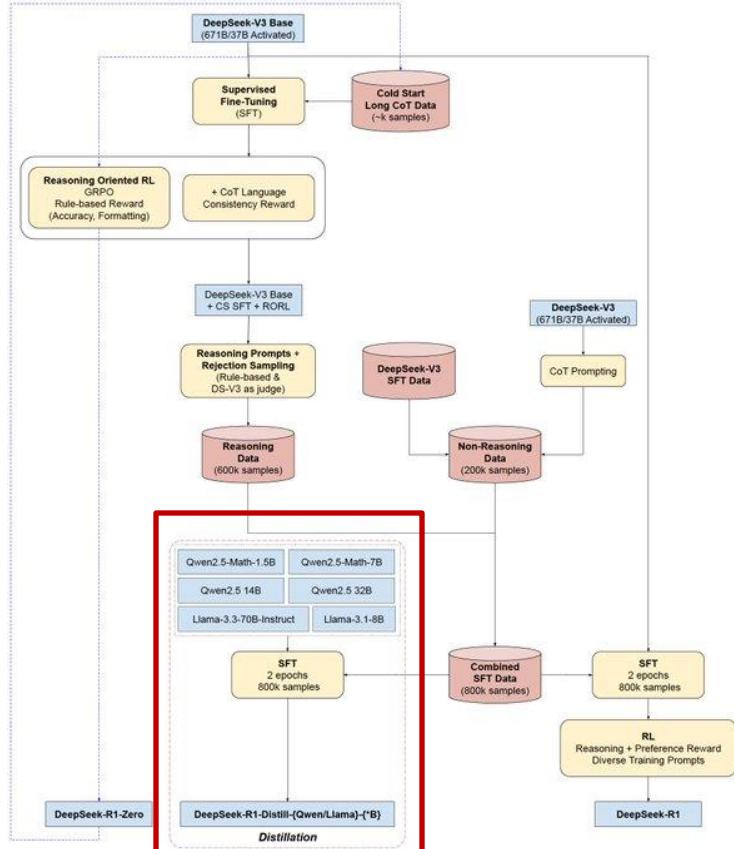


Figure from: <https://x.com/SirrahChan/status/1881540279783887036>



# Topics for Today

## Reasoning

- Introduction and facets of reasoning
- Before LLM Reasoning and CoT
- Prompting for Reasoning
  - Single Agent
  - Multi-Agent
- Reasoning Evaluation
- Training of reasoning models
- Reasoning Efficiency

# Reasoning Efficiency



- Big progress in accuracy with reasoning.
- The autoregressive paradigm imposes significant burden, limiting the application of these methods.
- LLMs often exhibit excessive reasoning, with several redundant wordings.
- Mechanisms to reduce these costs, such as using external mechanisms to orchestrate between reasoning and non-reasoning models.

# Reasoning Chain Compression



- TokenSkip ([Xia et al., EMNLP 2025](#))
- Not all the tokens in the chain-of-thought reasoning steps are important.
- Identifies more important tokens and fine-tunes the model to generate shorter reasoning chains.

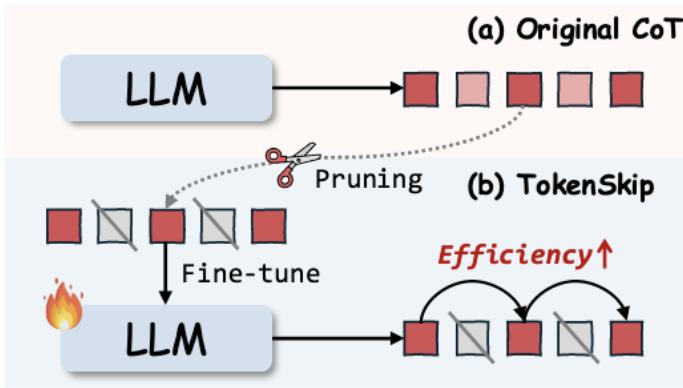


Figure 1: In contrast to vanilla CoT that generates all reasoning tokens sequentially, TokenSkip enables LLMs to *skip* tokens with less semantic importance (e.g., ) and learn shortcuts between critical reasoning tokens, facilitating controllable CoT compression.

# Reasoning Chain Compression (cont.)

- Uses GPT-4 to label each token as important or not; and trains a model,  $M_B$ , with token classification objective.
- The importance of each token,  $x_i$ , is measured by the probability assigned to that token by this model:  $I_2(x_i) = P(x_i | \mathbf{x}_{\leq n}; \theta_{M_B})$

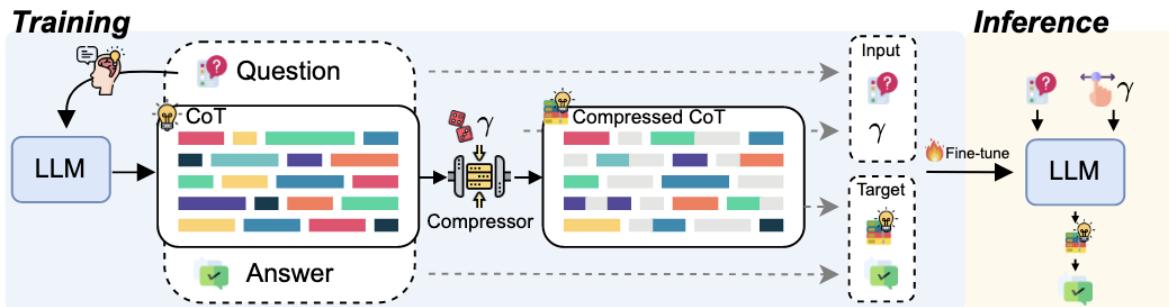


Figure 4: Illustration of TokenSkip. During training, TokenSkip first generates CoT trajectories from the target LLM. These CoTs are then compressed to various ratios sampled from the ratio set. TokenSkip fine-tunes the LLM using compressed CoTs with mixed ratios, enabling controllable CoT inference at any desired  $\gamma \in \{\gamma_0, \dots, \gamma_z\}$ .

# LLM Orchestration



- **Trade-off:** Powerful models offer better results but are expensive, while smaller models are more cost-effective but less capable.
- Can one use simpler models/methods for simpler problems and more sophisticated models/methods for more complex problems?

# LLM Orchestration (cont.)

- Learning to Route LLMs with Confidence Tokens ([Chuang et al., ICML 2025](#))
- LLM confidence is used to determine whether the response from a smaller model can be accepted.
- Proposed a method to teach LLMs to express confidence in whether their answers are correct.
- Compared this with methods that use token probabilities or verbalizing confidence.

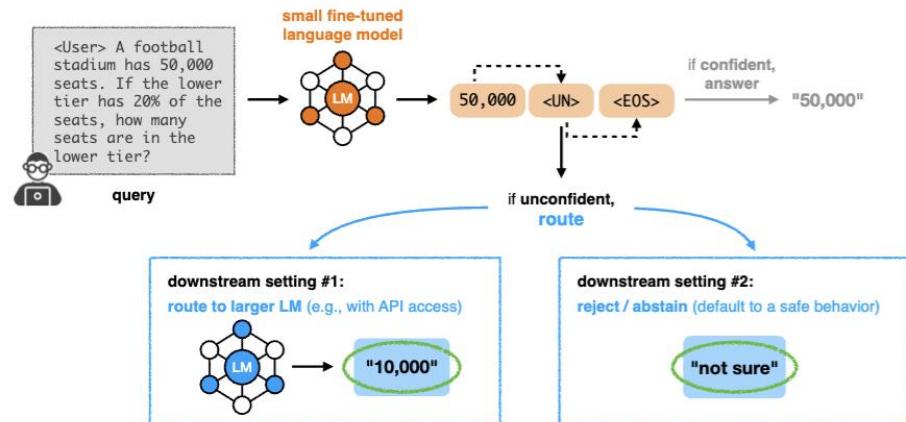


Figure 1: Self-REF attaches a confidence token to each prediction. Routing proceeds based on the confidence.

# LLM Orchestration (cont.)



- RouteLLM ([Ong et al., ICLR, 2025](#)).
- Learns router models that dynamically select between stronger and weaker models during inference.
  - A lightweight classifier
- Objective: minimize inference costs while achieving a specific performance target.
- Supervised approach: Evaluates responses from multiple models to determine the “best” model for each input, trains the router model with this data. (best: cheapest model with the highest value)
- Self-improvement: Uses RL to adjust routing over time.

# Next Week



**Tuesday**

**Alignment and Preference Learning** (Guest Lecturer: Ishika Agarwal)

**Part 1: RL basics**

- state, action, environment, reward, etc.

**Part 2: RL algorithms**

- PPO, DPO, GRPO

**Part 3: RL applications**

- preference learning
- reasoning

**Thursday**

- Midterm 2