CS 957, System-2 AI Abstraction & Reasoning

Mahdieh Soleymani

May 2025

Sharif University of Technology

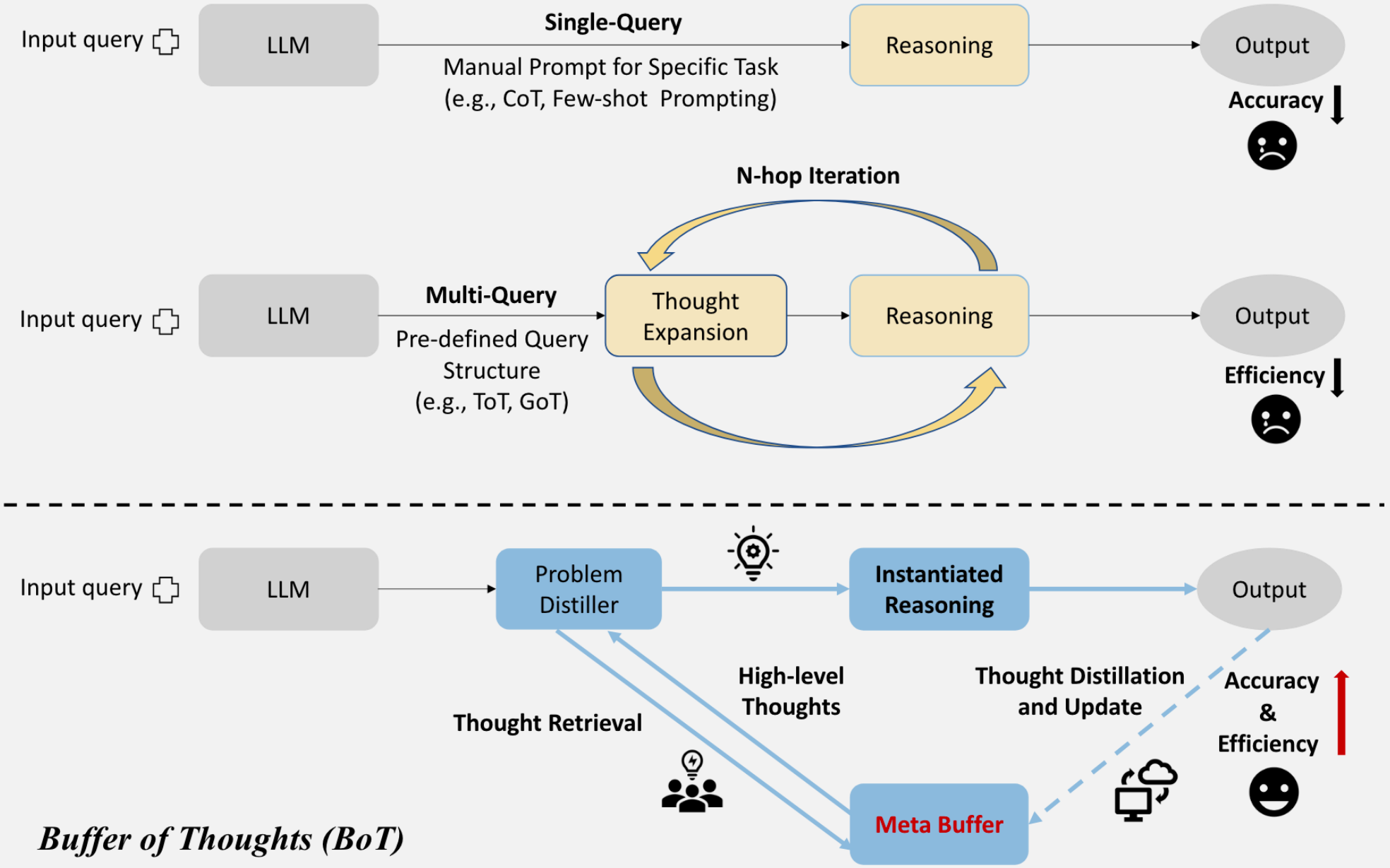
Buffer of Thoughts: Thought-Augmented Reasoning with Large Language Models

Ling Yang[1*✉], **Zhaochen Yu**[1*], **Tianjun Zhang**[2], **Shiyi Cao**[2], **Minkai Xu**[3],
Wentao Zhang[1], **Joseph E. Gonzalez**[2], **Bin Cui**[1]
[1]Peking University, [2]UC Berkeley, [3]Stanford University
Project: https://github.com/YangLing0818/buffer-of-thought-llm

Motivation

- Human often summarize and induce higher-level guidelines when solving problems and then apply them to relevant problems.

- Meta-buffer: a lightweight library that contains high-level thoughts (thought-template) for addressing various types of problems.



L. Yang et al., Buffer of Thoughts: Thought-Augmented Reasoning with Large Language Models, NeurIPS 2024

Input Problem

A certain shopping mall sells a batch of branded shirts, with an average daily sales of 20 pieces and a profit of 40 yuan per piece. In order to expand sales, increase profits, and reduce inventory as soon as possible, the mall has decided to take appropriate price reduction measures. After investigation, it was found that for every 1 yuan decrease in the price of this shirt, an average of 2 more shirts are sold per day. If the mall wants to make an average profit of 1200 yuan per day, how much price should each shirt be reduced?

Chain-of-Thought

1.Calculate the current daily profit:
Current daily profit = 20 * 40 = 800 yuan

2.Calculate additional daily profit from selling more shirts:
Additional daily profit = 2 * 40 = 80 yuan

……

5. Calculate Price reduction:
Price reduction needed = 320 / 2 = 160 yuan

Therefore, each shirt should be reduced by 160 yuan in order for the mall to make an average profit of 1200 yuan per day. ✗

Plan-and-Solve

To devise a plan to solve the problem, let's break it down into steps:

1.Determine the current daily profit:
Calculate the current daily profit using the average daily sales and profit per shirt

…….

3. Set up the equation for the desired daily profit
Create an equation that relates new profit per shirt, increased number of shirts sold, and desired daily profit

……

Step 4: Solve the equation
Expand the equation and collect like terms:
$1200=(20\times40)+(2p\times40)-(20\times p)-(2p\times p)$
Divide by 2 to simplify:
$p^2-30p-200=0$
$p\approx35.6155$ ✗

Buffer of Thoughts

Thought Template T_1

To solve any quadratic equation of the form $ax^2+bx+c=0$, we can follow a general approach based on the method described. Here is the structured template for solving such equations:
Step 1: Calculate the Discriminant
•Compute the discriminant D using the formula $D=b^2-4ac$.
Step 2: Determine the Nature of the Roots
……

……

Thought Template T_N

```
def process_element(element):
    # Define how to process each individual element, e.g., applying a filter or transformation.
def combine_elements(element1, element2):
    # Define how to combine elements, e.g., summing numbers, concatenating strings, etc.
def check_condition(accumulated_result):
    # Define the condition that the accumulated result must meet.
def solve_problem(input_list):.
```

Meta Buffer

Problem Distillation & Thought Retrieval

Thought Template T_1

To solve any quadratic equation of the form $ax^2+bx+c=0$, we can follow a general approach based on the method described. Here is the structured template for solving such equations:
Step 1: Calculate the Discriminant
•Compute the discriminant D using the formula $D=b^2-4ac$.
Step 2: Determine the Nature of the Roots
•If $D\geq0$, the equation has two distinct real roots.
•If $D=0$, the equation has exactly one real root (also known as a repeated or double root).
•If $D<0$, the equation has two complex roots.
Step 3: Compute the Roots
•For $D\geq0$, calculate the roots using the formula $x=\frac{-b\pm\sqrt{D}}{2a}$
For $D<0$, calculate the real and imaginary parts of the complex roots using the formula $x=\frac{-b\pm\sqrt{-D}i}{2a}$
where i is the imaginary unit.

……

Instantiated Reasoning

Solution:
•p be the original profit per shirt (40 yuan).
•x be the amount by which the price of each shirt is reduced.
•q be the original quantity of shirts sold daily (20 pieces).
•r be the rate at which the quantity of shirts sold increases.
The mall wants to maintain an average daily profit of 1200 yuan, so we have:

……

After simplification:
$x^2-30x+200=0$
Now we can solve this quadratic equation using the steps outlined in the template:
Step 1: Calculate the Discriminant
$D=b^2-4ac$
$D=(-30)^2-4*1*200=100$
Step 2: Determine the Nature of the Roots
Since $D>=0$, the equation has two distinct real roots.
Step 3: Compute the Roots
$x=\frac{-b\pm\sqrt{D}}{2a}=\frac{30\pm10}{2}$
The two possible solutions are:
$x=20$ or $x=10$
Due to reducing inventory as soon as possible, x=20 is taken ✓

Buffer Manager

- Buffer-of-Thought aims to provide a general reasoning approach for various tasks
- Thought-template: template description and its corresponding category as (T_i, D_{T_i}, C_k)
- The thought-templates are classified into six categories:
  - Text Comprehension
  - Creative Language Generation
  - Common Sense Reasoning
  - Mathematical Reasoning
  - Code Programming
  - Application Scheduling

Task Description:
Solve an quadratic equation of the form $ax^2 + bx + c = 0$ considering any situations.

Solution Description:
To solve any quadratic equation of the form $ax^2 + bx + c = 0$, we can follow a general approach based on the method described. Here is the structured template for solving such equations:

Thought Template:

Step 1: Calculate the Discriminant

- Compute the discriminant D using the formula $D = b^2 - 4ac$.

Step 2: Determine the Nature of the Roots

- If $D > 0$, the equation has two distinct real roots.
- If $D = 0$, the equation has exactly one real root (also known as a repeated or double root).
- If $D < 0$, the equation has two complex roots.

Step 3: Compute the Roots - For $D \geq 0$, calculate the roots using the formula $x = \frac{-b \pm \sqrt{D}}{2a}$.

- For $D < 0$, calculate the real and imaginary parts of the complex roots using the formula $x = \frac{-b}{2a} \pm \frac{\sqrt{-D}}{2a}i$, where i is the imaginary unit.

Modules of BoT

- **Problem-distiller**: extracting critical task-specific information along with relevant constraints
- **Search in meta-buffer** that contains a series of high-level thoughts (thought-template) and retrieve a most relevant one.
- **Instantiate** the retrieved thought-template with more task-specific reasoning structures and conduct **reasoning** process.
- **Buffer-manager** summarizes the whole problem-solving process and distilling high-level thoughts for increasing the capacity of meta-buffer.

Prompt for Problem Distiller

***[Problem Distiller]**:*
As a highly professional and intelligent expert in information distillation, you excel at extracting essential information to solve problems from user input queries. You adeptly transform this extracted information into a suitable format based on the respective type of the issue.
Please categorize and extract the crucial information required to solve the problem from the user's input query, the distilled information should include.
1. Key information:
Values and information of key variables extracted from user input, which will be handed over to the respective expert for task resolution, ensuring all essential information required to solve the problem is provided.
2. Restrictions:
The objective of the problem and corresponding constraints.
3. Distilled task:
Extend the problem based on 1 and 2, summarize a meta problem that can address the user query and handle more input and output variations. Incorporate the real-world scenario of the extended problem along with the types of key variables and information constraints from the original problem to restrict the key variables in the extended problem. After that, use the user query input key information as input to solve the problem as an example.

Problem Distiller

- **Problem condensation and translation**: Key elements are extracted from input tasks:
  1) Essential parameters and variables
  2) The objectives of the input tasks and their corresponding constraints.
- **Reorganization of distilled information** into a clear, comprehensible format for the subsequent reasoning stage.
- **Translation into high-level concepts** and structures
  - decomposes complex real-world problems into simpler, multi-step calculations, making it easier for later retrieval of high-level thought.

Thought-Augmented Reasoning with Meta Buffer

- Template retrieval for a distilled problem x_d

$$j = \text{argmax}_i(\text{Sim}(f(x_d), \{f(D_{T_i})\}_{i=1}^{N})), \quad \text{where} \quad \text{Sim}(f(x_d), \{f(D_{T_i})\}_{i=0}^{n}) >= \delta$$

  - $f(\cdot)$ is a normal text embedding model
  - T_j denotes the retrieved thought template.
  - δ (0.5~0.7 is recommended) to determine whether the current task is new.

- Instantiated reasoning by LLM

$$S_x = LLM_{\text{instantiation}}(x_d, T_j),$$

- otherwise, the task is identified as a new task.

Prompt for Instantiated Reasoning

[Meta Reasoner]
You are a Meta Reasoner who are extremely knowledgeable in all kinds of fields including Computer Science, Math, Physics, Literature, History, Chemistry, Logical reasoning, Culture, Language..... You are also able to find different high-level thought for different tasks. Here are three reasoning sturctures:
i) Prompt-based structure:
It has a good performance when dealing with problems like Common Sense Reasoning, Application Scheduling
ii) Procedure-based structure
It has a good performance when dealing with creative tasks like Creative Language Generation, and Text Comprehension
iii) Programming-based:
It has a good performance when dealing with Mathematical Reasoning and Code Programming, it can also transform real-world problems into programming problem which could be solved efficiently.
*(**Reasoning instantiation**)*
Your task is:
1. Deliberately consider the context and the problem within the distilled respond from problem distiller and use your understanding of the question within the distilled respond to find a domain expert who are suitable to solve the problem.
2. Consider the distilled information, choose one reasoning structures for the problem.
3. If the thought-template is provided, directly follow the thought-template to instantiate for the given problem.

Buffer Manager

- Meta-buffer stores informative high-level thoughts distilled from different problems:
  - generalize each specific solution to more problems
  - adaptively instantiate each thought template to address each specific task
  - continually improves the capacity of meta-buffer as more tasks are solved.

- To guarantee the scalability and stability of BoT, dynamically updates the meta-buffer

Buffer Manager: Template Distillation

- To extract a general though-template:
  1) Core task summarization: Describing basic types and core challenges of problems
  2) Solution steps description: summarizing the general steps for solving a problem
  3) General answering template: proposing a solution template or approach that can be widely applied to similar problems.

- The new template distilled from input task x can be denoted as:

$$T_{new} = LLM_{distill}(x_d, S_x),$$

Prompt for Template Distillation:

User: [Problem Description] + [Solution Steps or Code]

To extract and summarize the high-level paradigms and general approaches for solving such problems, please follow these steps in your response:

1. Core task summarization:

Identify and describe the basic type and core challenges of the problem, such as classifying it as a mathematical problem (e.g., solving a quadratic equation), a data structure problem (e.g., array sorting), an algorithm problem (e.g., search algorithms), etc. And analyze the most efficient way to solve the problem.

2. Solution Steps Description:

Outline the general solution steps, including how to define the problem, determine variables, list key equations or constraints, choose appropriate solving strategies and methods, and how to verify the correctness of the results.

3. General Answer Template:

Based on the above analysis, propose a template or approach that can be widely applied to this type of problem, including possible variables, functions, class definitions, etc. If it is a programming problem, provide a set of base classes and interfaces that can be used to construct solutions to specific problems.

Please ensure that your response is highly concise and structured, so that specific solutions can be transformed into generalizable methods.

[Optional] Here are some exemplars of the thought-template: *(Choose cross-task or in-task exemplars based on the analysis of the* ***Core task summarization****.)*

Buffer Manager: Dynamic Update of Meta-Buffer

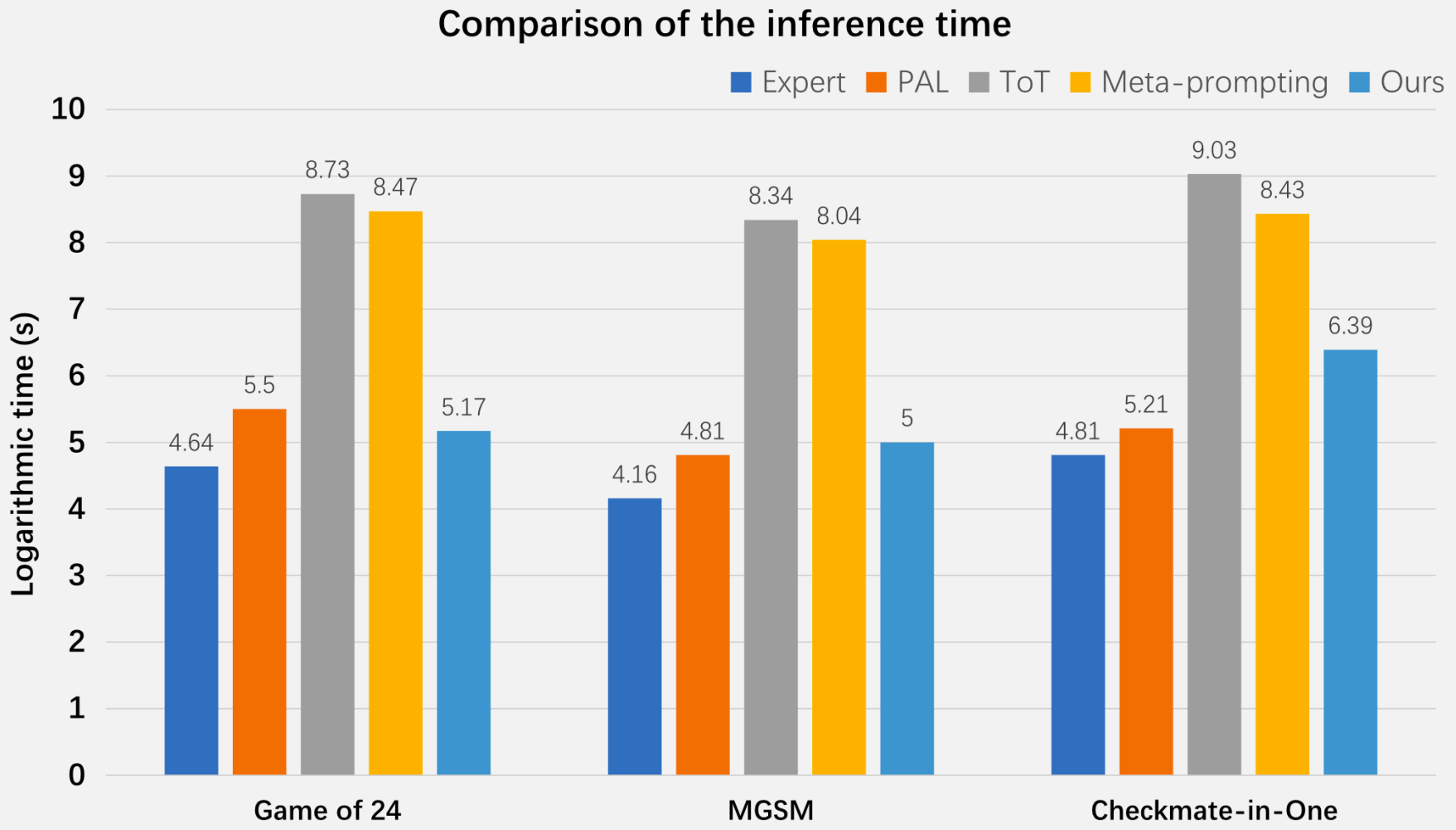
- After template distillation, we need to consider whether the distilled template should be updated into the meta-buffer.
  - It avoids the redundancy while mains newly-generated informative thoughts

If we encounter a problem without a proper thought-template, the distilled thought-templates will be directly stored in the meta-buffer.

else, similarity between the embedding vectors of $D_{T_{new}}$ and $\{D_{T_i}\}_{i=0}^{n}$ are computed and update the meta-buffer with the following rule:

$$\mathrm{Max}(\mathrm{Sim}(f(D_{T_{new}}), \{f(D_{T_i})\}_{i=0}^{n})) < \delta.$$

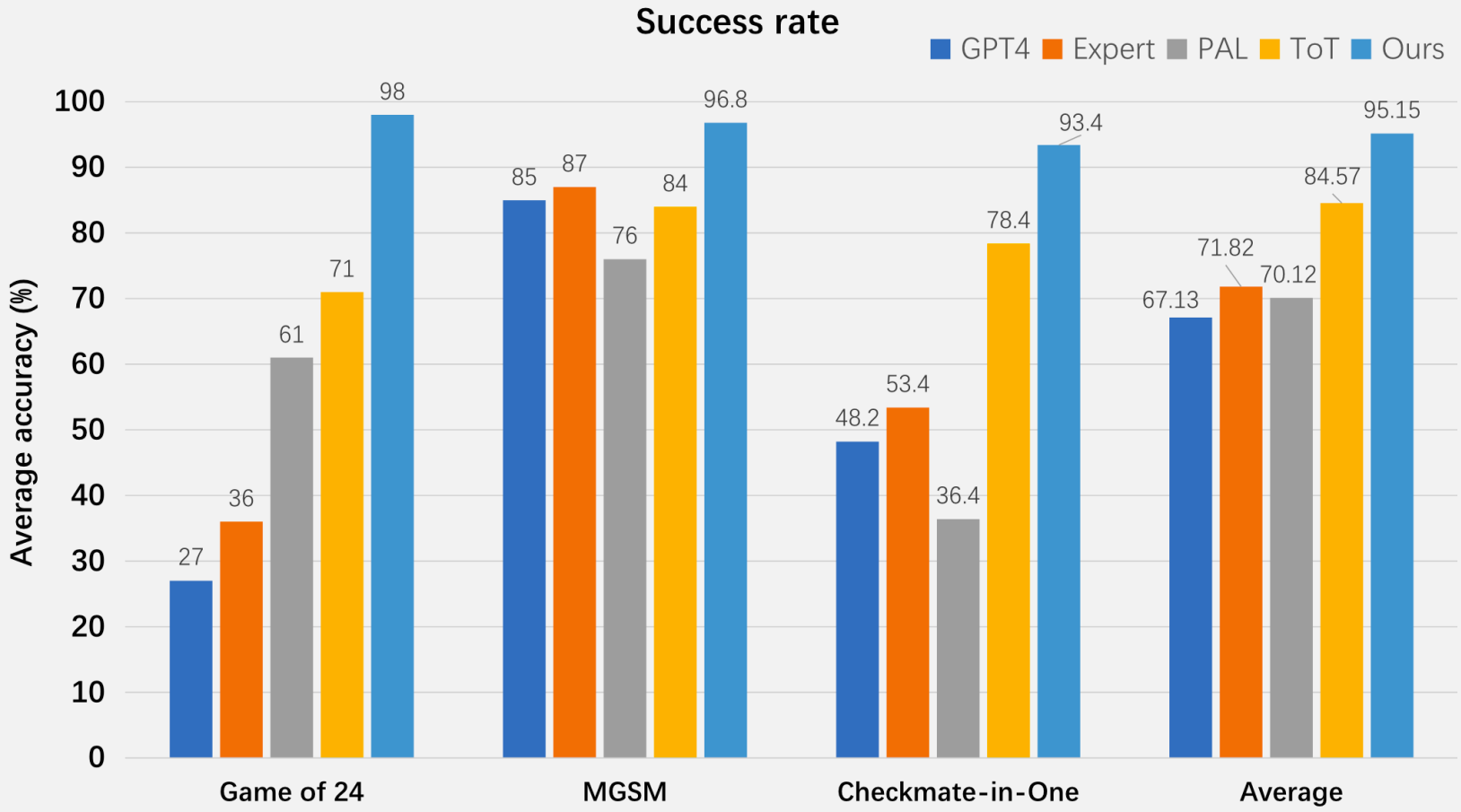
Task	Standard	Single-Query			Multi-Query			**BoT (Ours)**
	GPT4 [3]	GPT4+CoT [8]	Expert [9]	PAL [10]	ToT [14]	GoT [17]	Meta Prompting [15]	
Game of 24	3.0	11.0	3.0	64.0	**74.0**	73.2	67.0	**82.4**
MGSM (avg)	84.4	85.5	85.0	72.0	86.4	**87.0**	84.8	**89.2**
Multi-Step Arithmetic	84.0	83.2	83.2	87.4	88.2	89.2	**90.0**	**99.8**
WordSorting	80.4	83.6	85.2	93.2	96.4	98.4	**99.6**	**100.0**
Python Puzzles	31.1	36.3	33.8	47.3	43.5	41.9	**45.8**	**52.4**
Geometric Shapes	52.6	69.2	55.2	51.2	56.8	54.2	**78.2**	**93.6**
Checkmate-in-One	36.4	32.8	39. 6	10.8	49.2	51.4	**57.2**	**86.4**
Date Understanding	68.4	69.6	68.4	76.2	78.6	77.4	**79.2**	**88.2**
Penguins	71.1	73.6	75.8	**93.3**	84.2	85.4	88.6	**94.7**
Sonnet Writing	62.0	71.2	74.0	36.2	68.4	62.8	**79.6**	**80.0**

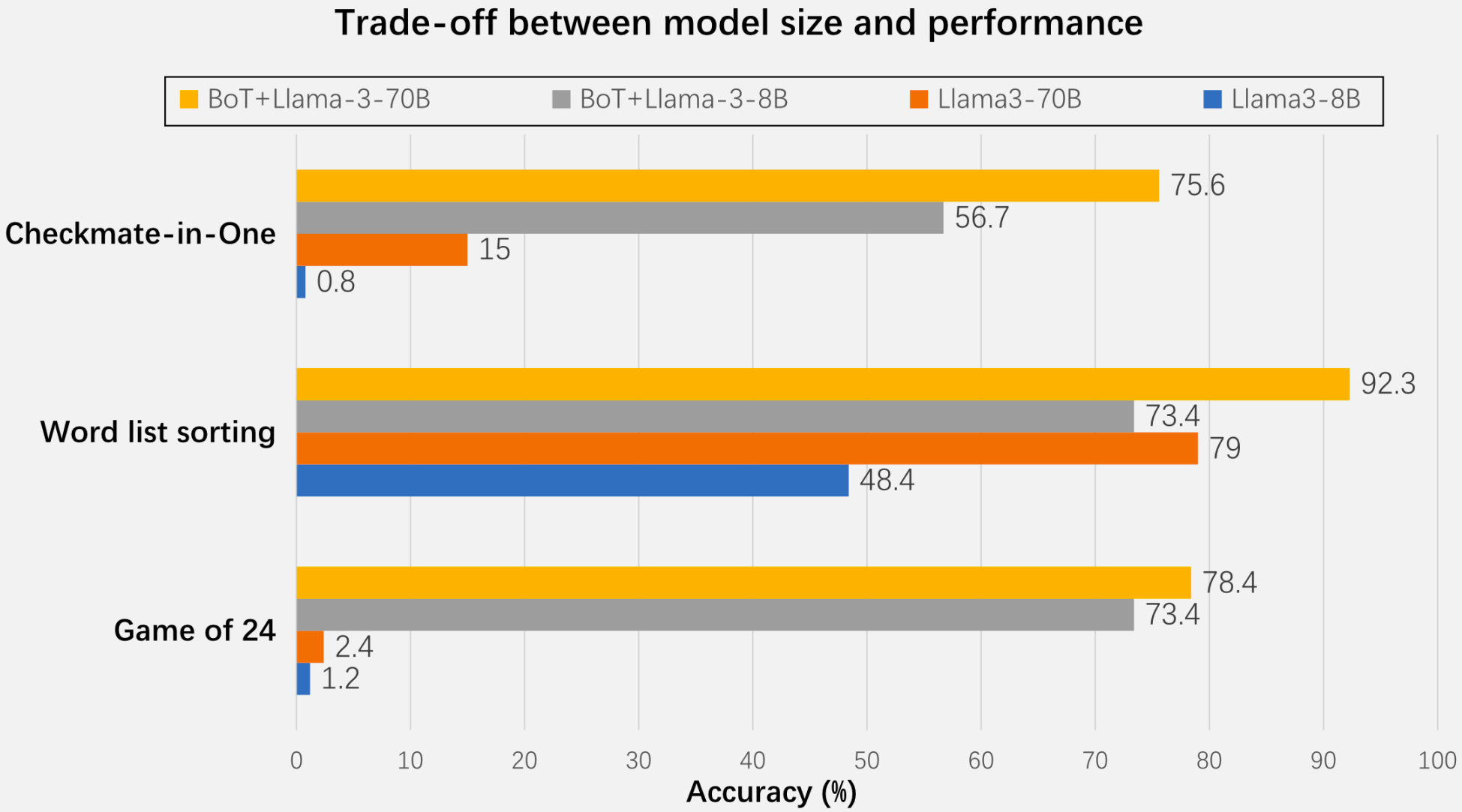


3: Comparison of **logarithmic inference time** between our Buffer of Thoughts and GPT CoT [8], Expert-prompting [9], PAL [10], ToT [14] across different benchmarks.

Reasoning Robustness

- 1000 examples from various benchmarks as a test subset




Trade-off between model size and performance
BoT+Llama-3-70B
BoT+Llama-3-8B
Llama3-70B
Llama3-8B
Checkmate-in-One
75.6
56.7
15
0.8
Word list sorting
92.3
73.4
79
48.4
Game of 24
78.4
73.4
2.4
1.2
0
10
20
30
40
50
60
70
80
90
100
Accuracy (%)

ReasonFlux: Hierarchical LLM Reasoning via Scaling Thought Templates

Ling Yang [1 *] **Zhaochen Yu** [2 *] **Bin Cui** [2] **Mengdi Wang** [1]
[1]Princeton University [2]Peking University
Code: https://github.com/Gen-Verse/ReasonFlux

Motivation

- The existing TTS methods often incur high computational costs
  - limiting their generalization ability to diverse and complex reasoning tasks.

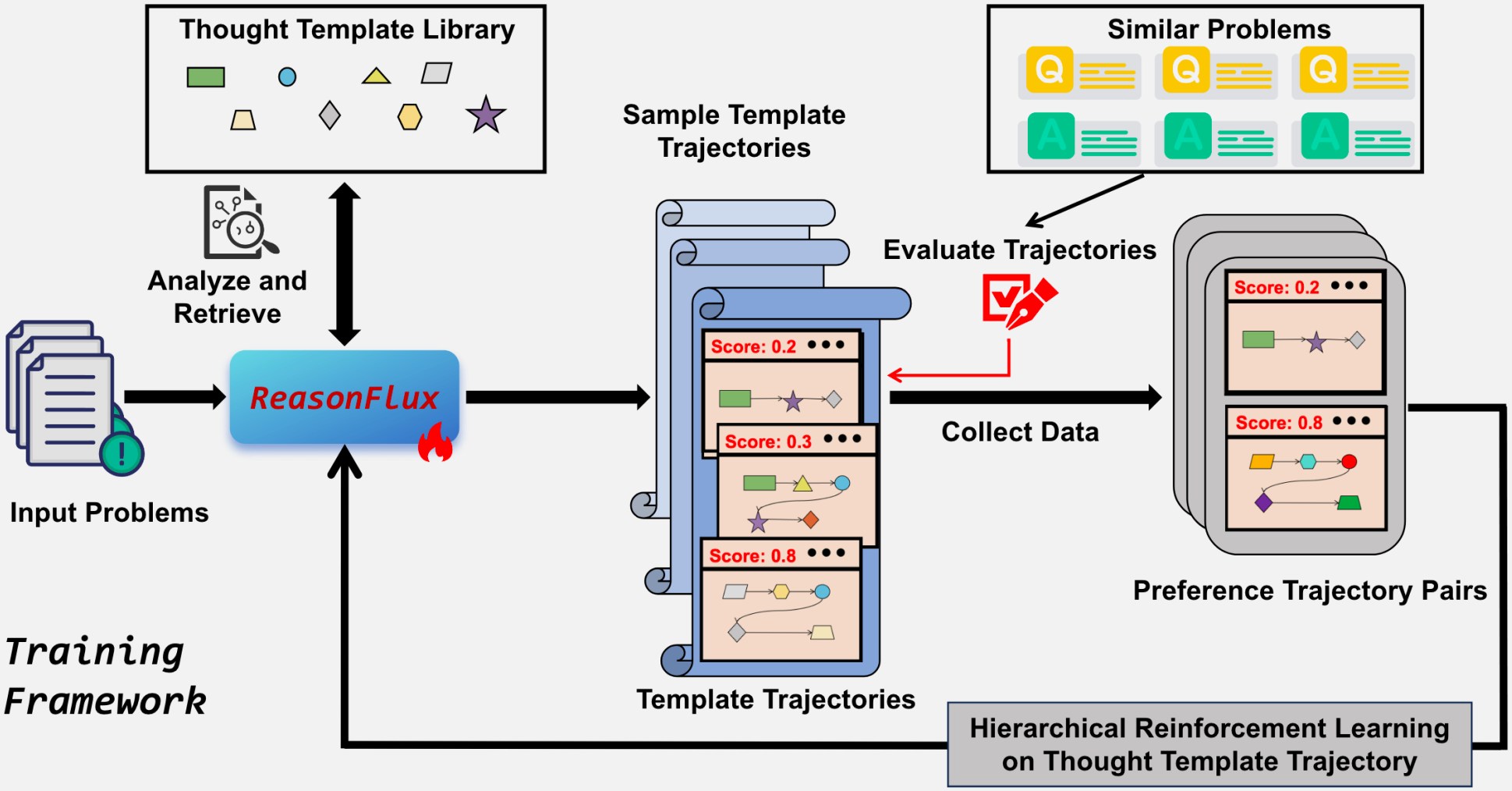
- Essentially, they struggle to effectively balance the exploration-exploitation trade-off during inference scaling.

- Abstraction effectively simplifies the search space of reasoning paths

BoT: Shortcomming

- Despite significant improvements, it may still face challenges when applied to complex reasoning tasks.
  - BoT struggles to address effectively the integration of multiple templates or diverse pieces of retrieved information

ReasonFlux



L. Yang et al., ReasonFlux: Hierarchical LLM Reasoning via Scaling Thought Templates, 2025

Constructing Structured Thought Template Library

carefully selects a wide and diverse range of challenging mathematical reasoning problems from different sources.

An LLM is used to analyze the thought behind the solution and generating concise summaries of problem-solving strategies and identifying common patterns.

Constructing Structured Thought Template Library

- Structured organization, combined with rich metadata, ensures that the most relevant templates are readily available: $D_{temp} = \{T_1, T_2, \ldots, T_m\}$

Example Template

name: $\sqrt{R^2 - x^2}$ Type Trigonometric Substitution
tag: Substitution Method, Trigonometric Substitution, Irrational Function
description: When a radical of the form $\sqrt{R^2 - x^2}$ appears in a problem, and $|x| \leq R$, consider using trigonometric substitution $x = R\sin\theta$ or $x = R\cos\theta$ to eliminate the radical, converting the irrational expression into a trigonometric expression. This allows simplification and problem-solving using the properties and identities of trigonometric functions.
scope: Problems involving function optimization or range, especially those involving irrational functions of the form $\sqrt{R^2 - x^2}$. Equations or inequalities containing radicals of the form $\sqrt{R^2 - x^2}$. Geometric problems related to circles.
application steps:
1. **Determine the range**: Based on the problem conditions, determine the range of x, usually $|x| \leq R$.
... (Steps 2-5 omitted for brevity)
example:
... (Examples omitted for brevity)

Hierarchical LLM Reasoning Framework

- Hierarchical RL to effectively plan out a thought template trajectory for a problem
- Hierarchical RL on a sequence of high-level thought templates effectively simplifying the search space of reasoning paths.
- ReasonFlux acts as an experienced navigator, providing the optimal trajectory denoted as $\mathbb{T}_{traj}$ that enabling the LLM to instantiate abstract thought templates

Why Hierarchical RL?

Performing tasks at various levels of abstractions

Bake a cheesecake

Buy ingredients

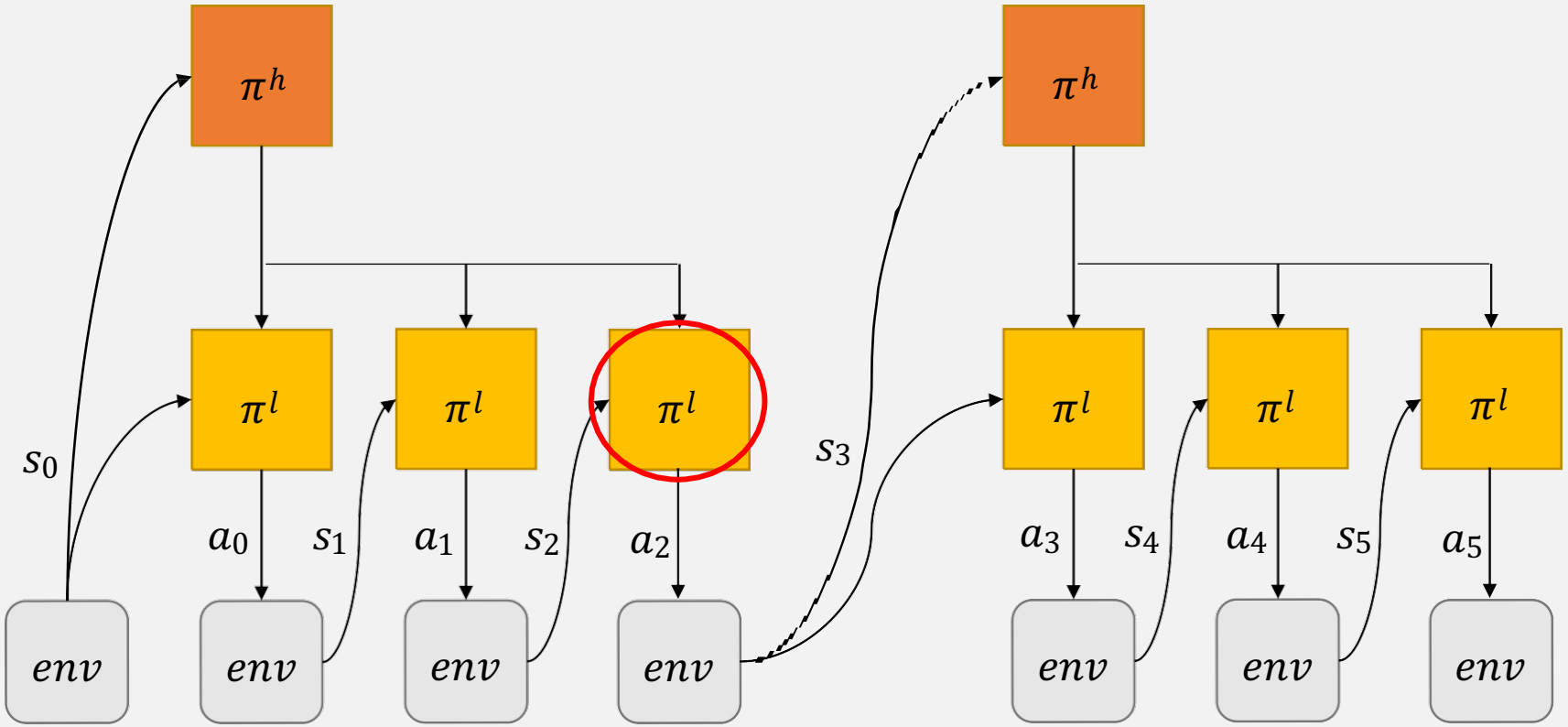
Go to the store

Walk to the door

Take a step

Contract muscle X

Hierarchical RL



Hierarchical RL on Thought Template Trajectory

- ReasonFlux effectively plan out an optimal thought template trajectory for a problem using the following steps:
  1. Structure-based Finetuning
  2. Preference Learning on Thought Template Trajectory
  3. Inference Scaling with Scaling Thought Templates

Structure-based Finetuning

- D_{train} comprises diverse tuples $(T_{\text{nam}}, T_{\text{tag}}, T_{\text{des}}, T_{\text{sco}})$ extracted from D_{temp}.
  - T_{nam}: template names
  - T_{tag}: their associated tags
  - T_{des}: detailed descriptions of their underlying principles
  - T_{sco}: a clear delineation of their applicable scopes

$$\mathcal{L}_{\text{struct}} = -\mathbb{E}_{\mathcal{D}_{\text{train}}} \left[\log \pi(T_{\text{des}}, T_{\text{sco}} | T_{\text{nam}}, T_{\text{tag}})\right]$$

- The fine-tuned model π_{struct} can effectively associate (T_{nam} and T_{tag}) with its functional aspects (T_{des} and T_{sco})

Preference Learning on Thought Template Trajectory

- π_{struct} first analyzes and abstracts the problem information, identifying the core mathematical concepts and relationships

- The navigator π_{struct} configures a trajectory $\mathbb{T}_{traj} = \{s_1, s_2, \dots, s_n\}$
  - s_i represents a high-level step in the reasoning process, associated with a specific template T_i.
  - Each retrieved template T_i is instantiated with specific details from the input problem x
  - It provides fine-grained guidance to a separate inference LLM called π_{inf} to solve the problem.

Preference Learning on Thought Template Trajectory

- **Trajectory reward** $R(\mathbb{T}_{\text{traj}})$: The average accuracy achieved by π_{inf} on the similar problems to x_i when it is guided by instantiated templates along the trajectory $\mathbb{T}_{\text{traj}}$

$$R(\mathbb{T}_{\text{traj}}) = \frac{1}{|\mathcal{X}_{sim}|} \sum_{x_i \in \mathcal{X}_{sim}} \text{Acc}(\pi_{inf}(x_i, \mathbb{T}_{\text{traj}}))$$

Preference Learning on Thought Template Trajectory

- For each problem x, multiple different $\mathbb{T}_{\text{traj}}$ are sampled and evaluated utilizing $R(\mathbb{T}_{\text{traj}})$
- For each two trajectories, a pair of $\mathbb{T}^{+}_{\text{traj}}$ and $\mathbb{T}^{-}_{\text{traj}}$ is formed where $R(\mathbb{T}^{+}_{\text{traj}}) > R(\mathbb{T}^{-}_{\text{traj}})$.
- π_{struct} is optimized as:

$$\mathcal{L}_{\text{TTR}}(\theta) = -\mathbb{E}_{(x,(\mathbb{T}^{+}_{\text{traj}},\mathbb{T}^{-}_{\text{traj}}))\sim\mathcal{D}_{pair}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(\mathbb{T}^{+}_{\text{traj}}|x)}{\pi_{sft}(\mathbb{T}^{+}_{\text{traj}}|x)} - \beta\log\frac{\pi_\theta(\mathbb{T}^{-}_{\text{traj}}|x)}{\pi_{sft}(\mathbb{T}^{-}_{\text{traj}}|x)}\right)\right]$$

- π_θ is initialized as π_{struct} and optimized using the above cost

Direct Preference Optimization (DPO)

RLHF Objective

(get **high reward**, stay **close** to reference model)

any reward function

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi(y|x)}\left[r(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid x)\|\pi_{\mathrm{ref}}(\cdot \mid x))$$

Direct Preference Optimization (DPO)

RLHF Objective

(get **high reward**, stay **close** to reference model)

any reward function

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi(y|x)}\left[r(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid x)\|\pi_{\mathrm{ref}}(\cdot \mid x))$$

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

$$\pi^*(y \mid x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

with $Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$ ← Note **intractable sum** over possible responses; can't immediately use this

Direct Preference Optimization (DPO)

RLHF Objective

(get **high reward**, stay **close** to reference model)

any reward function

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi(y|x)}\left[r(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid x)\|\pi_{\mathrm{ref}}(\cdot \mid x))$$

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

$$\pi^*(y \mid x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

with $Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$

Note **intractable sum** over possible responses; can't immediately use this

Rearrange

(write **any reward function** as function of **optimal policy**)

Ratio is **positive** if policy likes response more than reference model, **negative** if policy likes response less than ref. model

$$r(x,y) = \underbrace{\beta\log\frac{\pi^*(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta\log Z(x)}_{\text{some parameterization of a reward function}}$$

Putting it together for DPO

- Derived reward model: $r_{\pi_\theta}(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$

Log Z term cancels as the loss only measures differences in rewards

- A loss for training r via the Bradley-Terry model of preferences:

Derived from the Bradley-Terry model of human preferences:

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma (r(x, y_w) - r(x, y_l)) \right]$$

Reward for winning sample

Reward for losing sample

- DPO: A loss function on policy:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]$$

Putting it together for DPO

A loss function on reward functions

+

A transformation between reward functions and policies

=

A loss function on policies

Derived from the Bradley-Terry model of human preferences:

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma(r(x, y_w) - r(x, y_l))\right]$$

$$r_{\pi_\theta}(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

When substituting, the **log Z term cancels**, because the loss only cares about **difference** in rewards

Reward of **preferred** response

Reward of **dispreferred** response

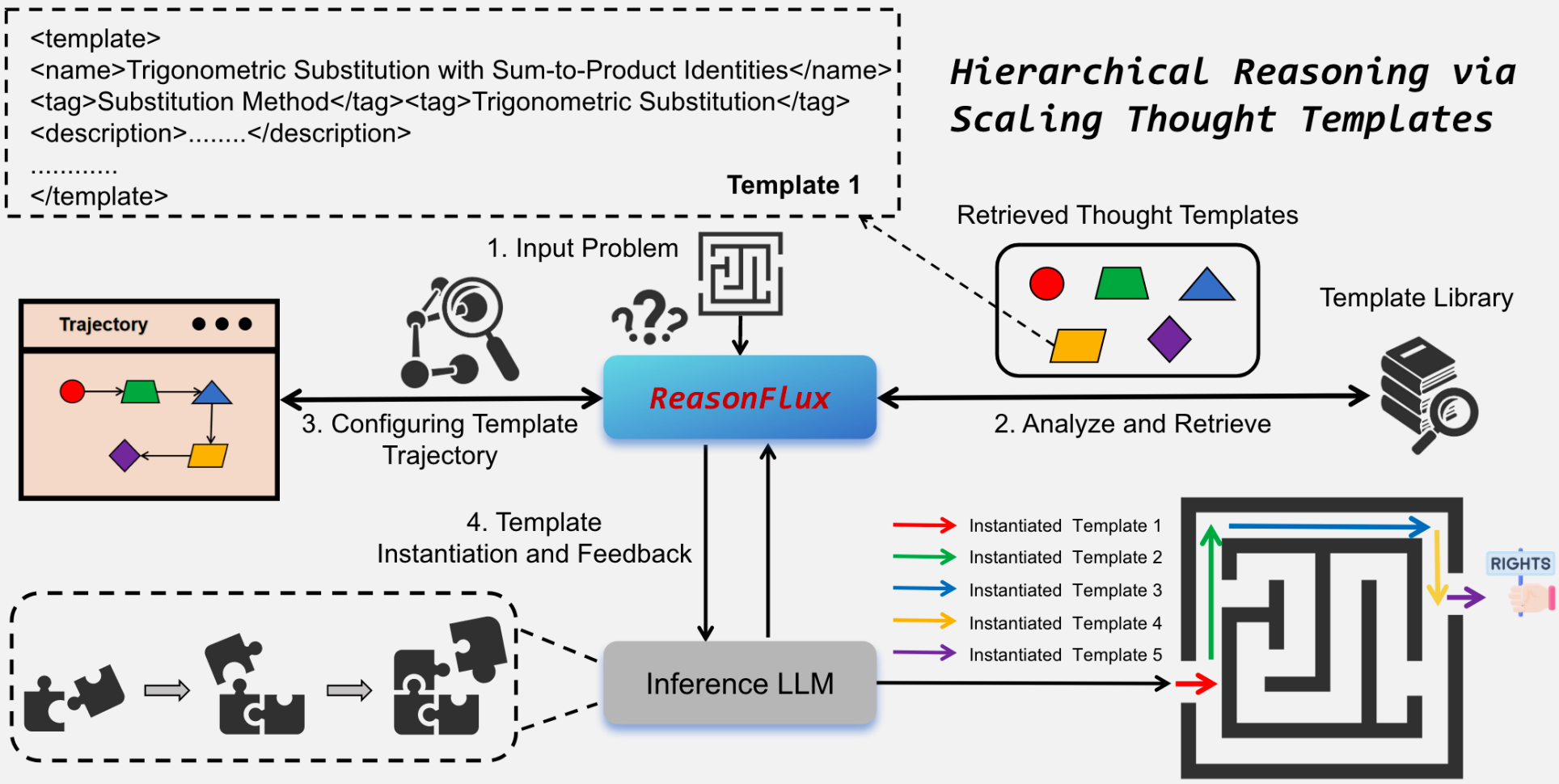
$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

Inference Scaling with Scaling Thought Templates

- After hierarchical RL process, we refer to optimized navigator π_θ as ReasonFlux.

- A novel inference scaling system is formed by multi-round interplay between the ReasonFlux, D_{temp}, and a downstream inference LLM π_{inf}

Inference Scaling with Scaling Thought Templates



Inference Scaling with Scaling Thought Templates

- ReasonFlux first analyzes and extracts concepts and relationships in x called $a(x)$
- Next, it configures an optimal template trajectory $\mathbb{T}^*_{\text{traj}} = \{s_1^*, s_2^*, \dots, s_n^*\}$
  - Each step s_i^* is associated with a specific template name T_{nam} and T_{tag}
- Then, it searches and retrieves a set of most relevant templates from D_{temp}
  - $T_{\text{rag}} = \{T_1, T_2, \dots, T_n\}$: n retrieved templates for n steps in the configured trajectory

$$T_{\text{rag}} = \textit{ReasonFlux}(\{T^i_{\text{nam}}, T^i_{\text{tag}}\}_{i=1}^n, \mathcal{D}_{\text{temp}}),$$

- Finally, based on $\mathbb{T}^*_{\text{traj}}$ and retrieved templates T_{rag}, ReasonFlux instructs π_{inf} to instantiate each steps s_i^*:

$$\hat{s}_i = \pi_{inf}(x_i, s_i, T_i),$$

Inference Scaling with Scaling Thought Templates (Cont.)

- Dynamic and interactive interplay between planning and execution
  - The interaction between ReasonFlux and π_{inf} is not a one-way process

- After obtaining $\hat{s}_i$, it is evaluated by ReasonFlux, and the adjustment is found as $\delta_i = \text{ReasonFlux}(\mathbb{T}^*_{traj}, \hat{s}_i)$.

- Then, ReasonFlux decides whether to refine the trajectory, potentially adjusting subsequent steps or even retrieving alternative templates:

$$\mathbb{T}^*_{\text{traj}} \leftarrow \textit{ReasonFlux}(\mathbb{T}^*_{\text{traj}}, \delta_i)$$

Training Details

- 8 NVIDIA A100 GPUs are used.
- Qwen2.5-32B-Instruct is the base model and also adopted as the inference LLM.
- In the structure-based finetuning stage, the initialized π_{struct} is fine-tuned on 15K samples extended from the template library D_{temp}.
- In the template trajectory optimization process, ReasonFlux is trained with 10K collected pair-wise trajectories from MATH (7.5k), and self-curated CN high-school competition-level data (2K) for 6 epochs

Task	**ReasonFlux 32B**	**DeepSeek V3**	**OpenAI o1-preview**	**OpenAI o1-mini**	**QWQ 32B-preview**	**GPT 4o**
MATH	**91.2**	90.2	85.5	90.0	90.6	76.6
AIME 2024	**56.7**	39.2	44.6	56.7	50.0	9.3
Olympiad Bench	**63.3**	55.4	-	65.3	61.2	43.3
GaokaoEn 2023	**83.6**	-	71.4	78.4	65.3	67.5
AMC2023	**85.0**	80.0	90.0	95.0	-	47.5

Table 1. Performance Comparison on Various Math Reasoning Benchmarks (Pass@1 Accuracy)

Table 2. Pass@1 accuracy comparison on various mathematical reasoning benchmarks.

Model	MATH-500	AIME 2024	AMC 2023	Olympiad Bench	Gaokao En 2023
Frontier LLMs					
GPT-4o	76.6	9.3	47.5	43.3	67.5
Claude3.5-Sonnet	78.3	16.0	-	-	-
GPT-o1-preview	85.5	44.6	90.0	-	71.4
GPT-o1-mini	90.0	56.7	95.0	65.3	78.4
Open-Sourced Reasoning LLMs					
DeepSeek-Coder-V2-Instruct	75.3	13.3	57.5	37.6	64.7
Mathstral-7B-v0.1	57.8	0.0	37.5	21.5	46.0
NuminaMath-72B-CoT	64.0	3.3	70.0	32.6	58.4
LLaMA3.1-8B-Instruct	51.4	6.7	25.0	15.4	38.4
LLaMA3.1-70B-Instruct	65.4	23.3	50.0	27.7	54.0
LLaMA3.1-405B-Instruct	73.8	-	-	34.8	-
Qwen2.5-Math-72B-Instruct	85.6	30.0	70.0	49.0	71.9
rStar-Math	88.2	43.3	80.0	63.1	78.2
DeepSeek-V3	90.2	39.2	80.0	55.4	-
ReasonFlux-32B	**91.2**	**56.7**	**85.0**	**63.3**	**83.6**
1.5B-Level Base Model					
Qwen2.5-Math-1.5B	51.2	0.0	22.5	16.7	46.5
Qwen2.5-Math-1.5B-Instruct	60.0	10.0	60.0	38.1	65.5
ReasonFlux-1.5B	**70.4**	**20.0**	**72.5**	**49.0**	**76.6**

Problem:

Let x, y, and z be positive real numbers satisfying the system of equations:
$\sqrt{2x - xy} + \sqrt{2y - xy} = 1$, $\sqrt{2y - yz} + \sqrt{2z - yz} = \sqrt{2}$, $\sqrt{2z - zx} + \sqrt{2x - zx} = \sqrt{3}$.
Then $[(1 - x)(1 - y)(1 - z)]^2$ can be written as $\frac{\square}{n}$, where m and n are relatively prime positive integers. Find $m + n$.

o1-mini

Step 1: Look for Symmetry or Patterns
Procedure:
The equations seem symmetric in a cyclic manner. Maybe there's a relationship or pattern that can help simplify them further.
Let's denote:
a = $\sqrt{x(2-y)}$, b = $\sqrt{y(2-z)}$,
c = $\sqrt{z(2-x)}$

Step 2: Assume Variables are Equal
Sometimes, assuming variables are equal can simplify the problem. Let's assume x = y = z .
If x = y = z , then the first equation becomes: $\sqrt{2x - \square^2} + \sqrt{2x - \square^2} = 1$
➔ $2\sqrt{2x - \square^2} = 1$ ➔ $\sqrt{2x - \square^2} = \frac{1}{2}$
Squaring both sides:
$2x - \square^2 = \frac{1}{4}$ ➔ $\square^2 - 2x + \frac{1}{4} = 0$
.....
This is a contradiction. Therefore, our assumption that x = y = z is incorrect.

Step 4: Try Expressing Variables in Terms of One Another
Let's try to express one variable in terms of another. From the first equation:
$\sqrt{x(2-y)} + \sqrt{y(2-x)} = 1$
This seems complicated. Maybe I can square both sides to eliminate the square roots.
Squaring both sides:
$(\sqrt{x(2-y)} + \sqrt{y(2-x)})^2 = 1^2$
This still looks complicated. Maybe there's a better approach.

.....

ReasonFlux

Analysis and configuration of the reasoning trajectory

The key to solving it is to determine the conditions for the existence of four distinct intersection points and to find the equation of the circle passing through these four points.
1.By analyzing the positional relationship between the vertices of the two parabolas, we can initially determine the range of values for the parameter k.
2. By appropriately adding and subtracting the two parabola equations to eliminate the coefficients of the quadratic terms, we can obtain the equation of a circle that passes through the intersection points of these two parabolas.
3. based on the radius constraint of the circle given in the problem, we further narrow down the range of k. Finally, verifying the boundary values to ensure they satisfy the condition of four intersection points allows us to determine the least and greatest elements of S.

Template 1.Preliminarily Determine the Range of k : Analyze the the Vertices of the Parabolas, and determine the condition for the existence of Intersection points
Instantiation:
Since x,y,z are all real numbers, and (2-x), (2-y),(2-z) > 0, So x, y, z ∈ (0, 2]

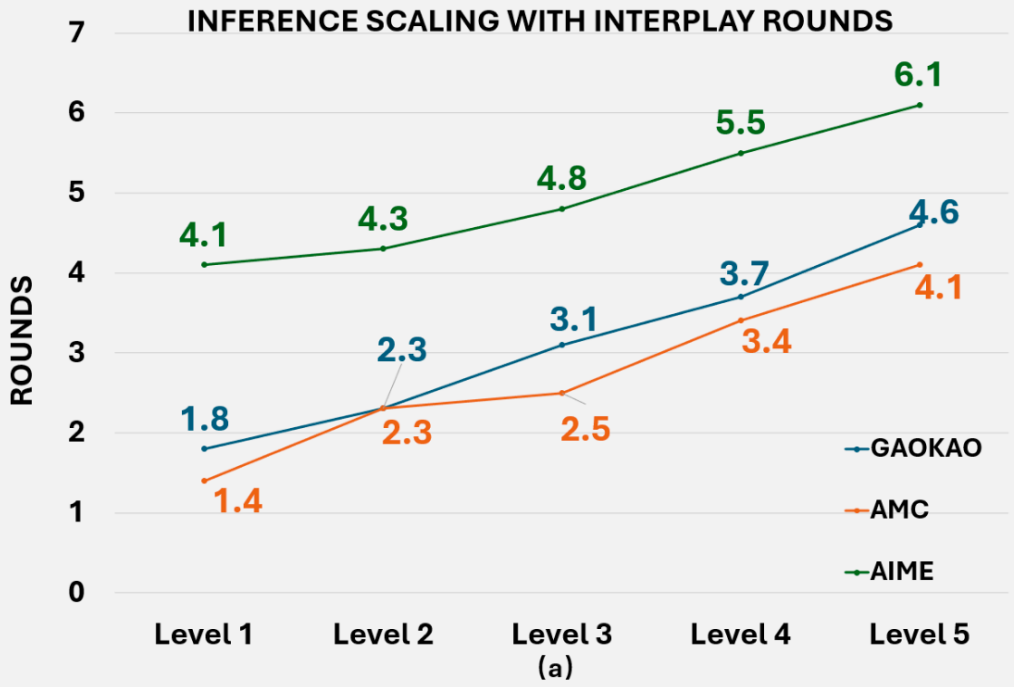
Template 2. Trigonometric Substitution:
Since x, y, $z \in (0, 2]$, and the expressions under the square roots can be written in the form $\sqrt{x(2-x)}$ which matches the trigonometric substitution form $\sqrt{\square^2 - \square^2}$

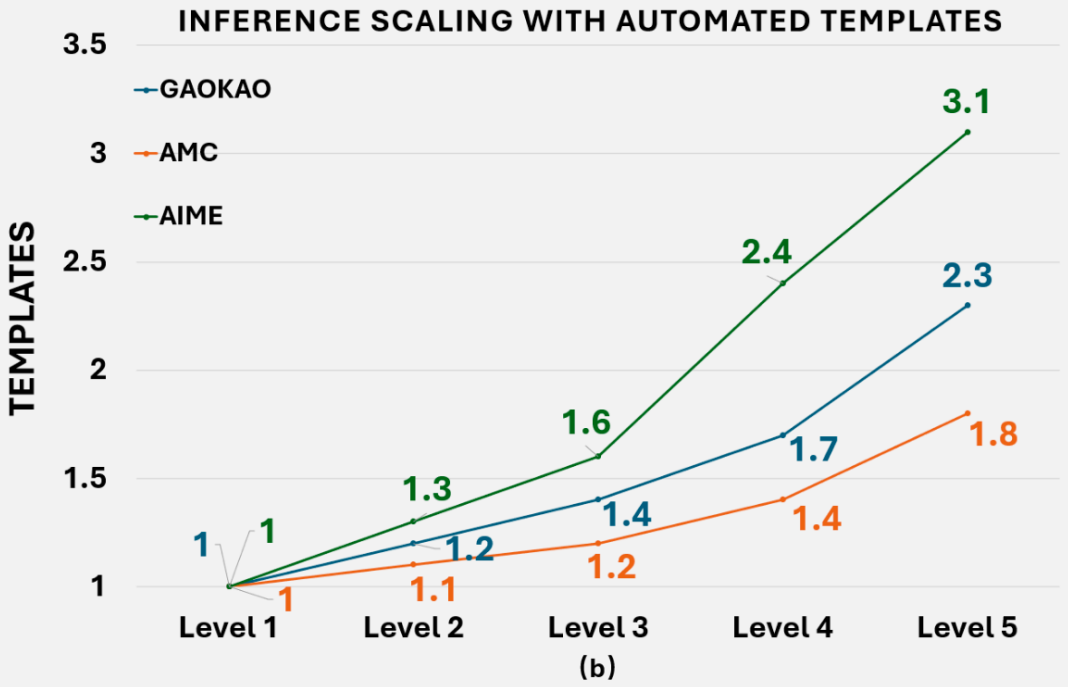
Instantiation:
we let x = 2 sin²α y = 2 sin²β, z = 2sin² γ, where α, β, γ, ∈ (0, π/2]

Template 3. Simplify the equation system:
Substitute trigonometric substitutions into the original system of equations using (sin²α + cos²α)(sin²β + cos²β) = 1 and the trigonometric identity sin²θ + cos²θ = 1 for simplification.
Instantiation:
1. First, substitute x and y:
sqrt(2*(2 sin²α) - (2 sin²α)(2 sin²β)) + sqrt(2*(2 sin²β) - (2 sin²α)(2 sin²β)) = 1
sqrt(4 sin²α (1 - sin²β)) + sqrt(4 sin²β (1 - sin²α)) = 1
2. Now, 1 - sin²θ = cos²θ, so:
sqrt(4 sin²α cos²β) + sqrt(4 sin²β cos²α) = 1
--> sin(α + β) = 1/2
--> sin(β + γ) = sqrt(2)/2
--> sin(α + γ) = sqrt(3)/2

Inference Scaling Laws for Template-Augmented Reasoning

- How ReasonFlux automatically trade off between cost and performance



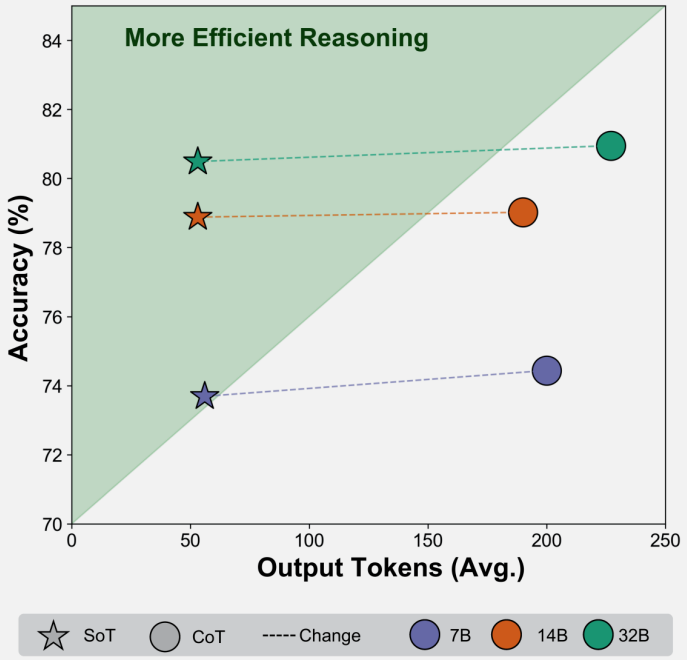


Sketch-of-Thought: Efficient LLM Reasoning with Adaptive Cognitive-Inspired Sketching

Simon A. Aytes[1] **Jinheon Baek**[1] **Sung Ju Hwang**[1,2]
KAIST[1] DeepAuto.ai[2]
{saytes, jinheon.baek, sungju.hwang}@kaist.ac.kr

Sketch-of-Thought (SoT)

- A novel prompting framework with three specialized reasoning paradigms grounded in cognitive science principles.
- We develop a lightweight auxiliary model that dynamically chooses the optimal reasoning paradigm for each query.



Reasoning Paradigms

- Conceptual Chaining
  - draws on associative memory networks to connect ideas with minimal verbalization
- Chunked Symbolism
  - applies working memory chunking theory to organize mathematical reasoning into concise symbolic representations
- Expert Lexicons
  - emulates the efficient domain-specific shorthand used by specialists.

- A lightweight router model dynamically selects the optimal reasoning paradigm ensuring that the most efficient reasoning strategy

Conceptual Chaining

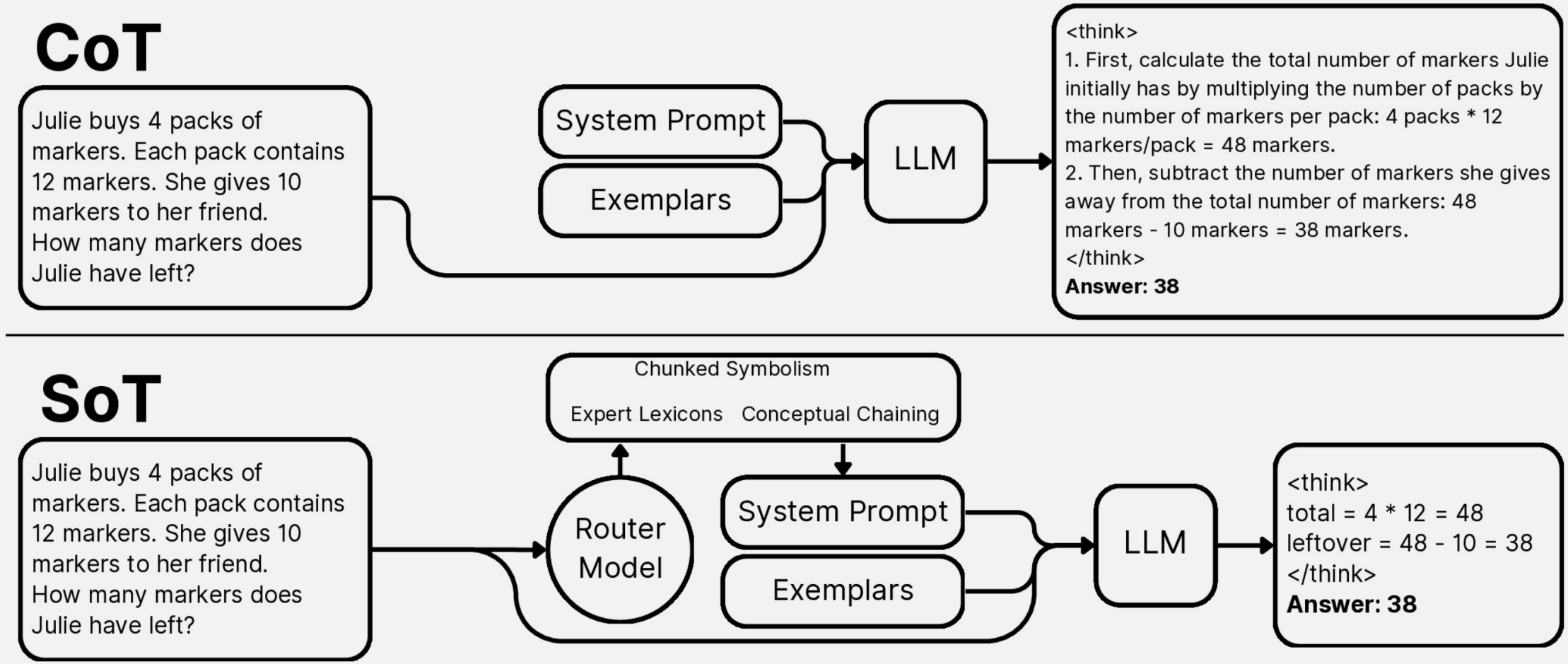
Q: What is the name of the currency used in Seoul?
A: <think> #Seoul → #South Korea → Won </think>
Answer: Korean Won

Expert Lexicons

Q: A patient with STEMI is given MONA therapy. They are allergic to aspirin. Are they at risk with this treatment?
A: <think> STEMI → ST-Elevation MI, MONA → Morphine, O2, Nitrates, Aspirin, so Aspirin ∈ MONA </think>
Answer: Yes

Chunked Symbolism

Q: A car accelerates at 2.5 m/s^2 for 10 seconds. If its initial velocity was 15 m/s, what is its final velocity?
A: <think> a = 2.5 m/s^2, t = 10 s, vi = 15 m/s vf = 15 + (2.5 × 10), vf = 40 m/s </think>
Answer: 40 m/s



Aytes et al., Sketch-of-Thought: Efficient LLM Reasoning with Adaptive Cognitive-Inspired Sketching, 2025

Router Model Training

- Collecting approximately 14,200 samples across reasoning tasks.
- Each sample was labeled with one of the three paradigms using GPT-4o (OpenAI, 2024)
- DistilBERT is finetuned due to its favorable balance of efficiency and performance.

Reasoning Task																
	Mathematical		**Commonsense**		**Logical**		**Multi-Hop**		**Scientific**		**Specialized**		**Overall**			
Method	%	Tokens	%	Tokens	%	Tokens	%	Tokens	%	Tokens	%	Tokens	% ↑	Tokens ↓	Red. ↑	Δ Acc ↑
Qwen-2.5-32B																
CoT	84.17	221	90.33	186	71.23	297	79.44	154	92.89	212	67.66	291	80.95	227	-	-
SoT	86.94	88	90.66	30	71.00	65	81.89	43	91.33	31	61.11	62	80.49	53	**76.22**	**-0.46**
SC+CoT	84.33	665	91.00	560	71.67	892	81.00	464	93.34	638	67.33	875	81.44	682	-	-
SC+SoT	87.50	265	90.66	92	71.33	197	82.67	129	92.00	94	61.66	187	80.97	161	**76.22**	**-0.47**
Qwen-2.5-14B																
CoT	83.00	189	90.44	155	67.00	248	77.67	148	90.89	164	65.11	234	79.02	190	-	-
SoT	82.72	78	89.78	35	67.44	63	79.89	45	90.89	37	62.56	62	78.88	53	**71.80**	**-0.14**
SC+CoT	83.17	569	92.33	467	69.33	744	76.33	446	91.00	493	66.33	703	79.75	570	-	-
SC+SoT	83.67	234	90.00	106	68.66	190	80.00	135	91.33	111	62.00	187	79.28	161	**71.80**	**-0.47**
Qwen-2.5-7B																
CoT	77.41	180	85.78	172	63.22	279	76.78	137	86.44	183	57.00	246	74.44	200	-	-
SoT	77.05	73	85.11	27	59.78	61	77.22	44	85.00	27	58.00	105	73.69	56	**71.90**	**-0.74**
SC+CoT	79.33	542	86.44	516	66.11	837	78.33	412	87.00	550	58.34	739	75.93	600	-	-
SC+SoT	78.83	219	85.66	81	60.34	184	77.33	134	85.00	82	59.00	317	74.36	169	**71.90**	**-1.57**

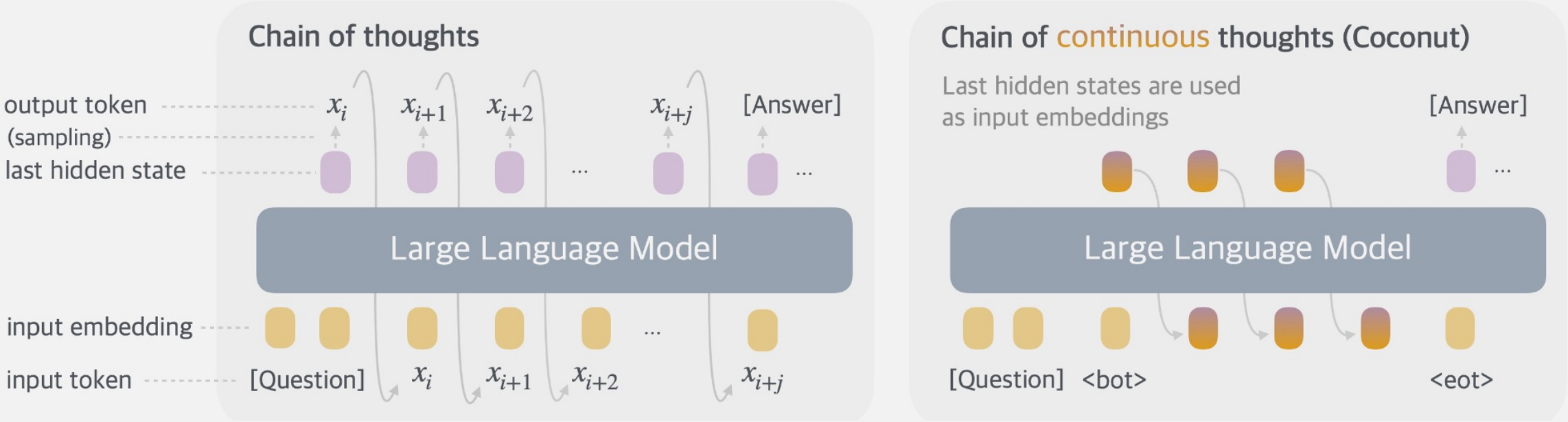
Training Large Language Models to Reason in a Continuous Latent Space

Shibo Hao[1,2,*], **Sainbayar Sukhbaatar**[1], **DiJia Su**[1], **Xian Li**[1], **Zhiting Hu**[2], **Jason Weston**[1], **Yuandong Tian**[1]

[1]FAIR at Meta, [2]UC San Diego
[*]Work done at Meta

COCONUT (Chain of Continuous Thought)

- While NTP is an effective training objective, it imposes a fundamental constraint on the LLM as a reasoning machine
  - The reasoning process of LLMs must be generated in word tokens.
  - The removal of language space constraints has led to a novel reasoning pattern
- LLM reasoning in an unrestricted latent space instead of using human language
  - human language is optimized for communication rather than reasoning



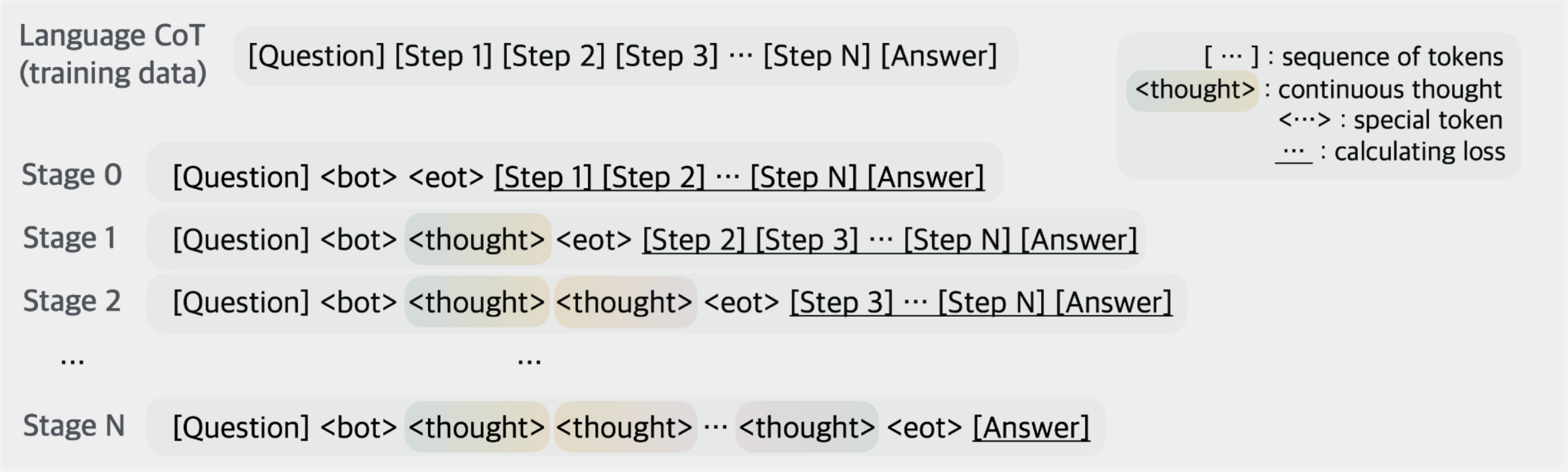
The last hidden state of the LLM as a representation of the reasoning state (i.e., “continuous thought”).

Hao et al., Training Large Language Models to Reason in a Continuous Latent Space, Dec 2024

COCONUT

- LLM switches between the “language mode” and “latent mode”
  - In language mode, the model operates as a standard LLM
  - In latent mode, it directly utilizes the last hidden state as the next input embedding.
    - The last hidden state represents the current reasoning state called “continuous thought”.

- Special tokens <bot> and <eot> are employed to mark the beginning and end of the latent mode, respectively.


Language CoT
(training data)
[Question] [Step 1] [Step 2] [Step 3] ⋯ [Step N] [Answer]
[⋯] : sequence of tokens
<thought> : continuous thought
<⋯> : special token
⋯ : calculating loss
Stage 0
[Question] <bot> <eot> [Step 1] [Step 2] ⋯ [Step N] [Answer]
Stage 1
[Question] <bot> <thought> <eot> [Step 2] [Step 3] ⋯ [Step N] [Answer]
Stage 2
[Question] <bot> <thought> <thought> <eot> [Step 3] ⋯ [Step N] [Answer]
...
...
Stage N
[Question] <bot> <thought> <thought> ⋯ <thought> <eot> [Answer]

Multi-stage Training Curriculum

- For an input $x = (x_1, \dots, x_T)$:

$$(h_1, h_2, \dots, h_t) = \text{Transformer}(e(x_1), e(x_2), \dots, e(x_t))$$

$$M(x_{t+1}|x_{1:t}) = \text{softmax}(W^T h_t) \qquad e(x_i) = Ex_i + p(i)$$

- Assume that latent reasoning occurs between positions i and j, i.e., $x_i = < bot >$ and $x_j = < eot >$.

$$E_t = [e(x_1), e(x_2), \dots, e(x_i), h_i, h_{i+1}, \dots, h_{j-1}, e(x_j), \dots, e(x_t)]$$

Training Procedure

- In the initial stage, language CoT data are leveraged to supervise continuous thought by implementing a multi-stage training curriculum

- In the subsequent stages, at the k-th stage, the first k reasoning steps in the CoT are replaced with k × c continuous thoughts
  - c is a hyperparameter controlling the number of latent thoughts replacing a single language reasoning step.

- During the training process, we mask the loss on questions and latent thoughts.
  - It's possible for the LLM to learn a more effective representation compared to language reasoning steps.

Training Details

- Continuous thoughts are fully differentiable, allowing backpropagation.

- $n + 1$ forward passes are performed when n latent thoughts are scheduled in the current training stage
  - computing a new latent thought with each pass
  - and then conducting an additional forward pass to obtain a loss on the remaining text sequence.

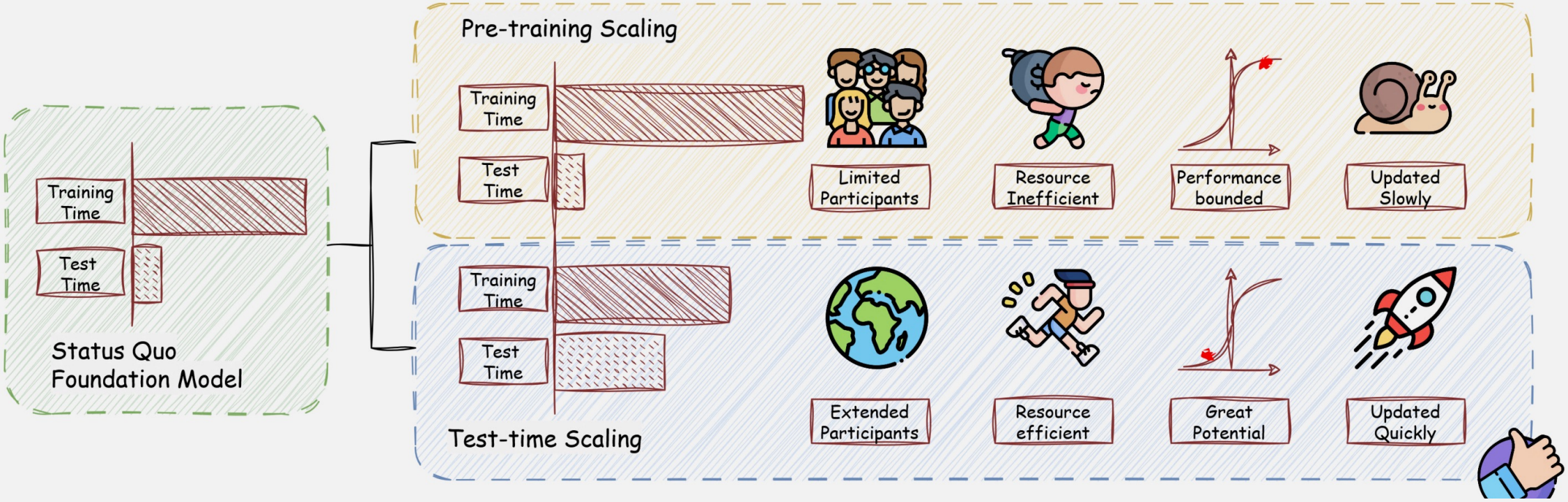
Inference Process

- Similar to the standard LLM decoding, except that in latent mode, we directly feed the last hidden state as the next input embedding.

- A challenge lies in determining when to switch between latent and language modes.
  - A <bot> token is inserted immediately following the question tokens.
  - For <eot>, two potential strategies are considered:
    - train a binary classifier on latent thoughts to enable the model to autonomously decide when to terminate the latent thoughts, or
    - always pad the latent thoughts to a constant length. We found that both approaches work comparably well.

Method	GSM8k		ProntoQA		ProsQA	
	Acc. (%)	# Tokens	Acc. (%)	# Tokens	Acc. (%)	# Tokens
CoT	42.9 ±0.2	25.0	98.8 ±0.8	92.5	77.5 ±1.9	49.4
No-CoT	16.5 ±0.5	2.2	93.8 ±0.7	3.0	76.7 ±1.0	8.2
iCoT	30.0*	2.2	99.8 ±0.3	3.0	98.2 ±0.3	8.2
Pause Token	16.4 ±1.8	2.2	77.7 ±21.0	3.0	75.9 ±0.7	8.2
COCONUT (Ours)	34.1 ±1.5	8.2	99.8 ±0.2	9.0	97.0 ±0.3	14.2
- *w/o curriculum*	14.4 ±0.8	8.2	52.4 ±0.4	9.0	76.1 ±0.2	14.2
- *w/o thought*	21.6 ±0.5	2.3	99.9 ±0.1	3.0	95.5 ±1.1	8.2
- *pause as thought*	24.1 ±0.7	2.2	100.0 ±0.1	3.0	96.6 ±0.8	8.2

Summary: TTS vs. Pretraining



Zhang et al., What, How, Where, and How Well? A Survey on Test-Time Scaling in Large Language Models, 2025