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

Abstract

We introduce Buffer of Thoughts (BoT), a novel and versatile thought-augmented reasoning approach for enhancing accuracy, efficiency and robustness of large language models (LLMs). Specifically, we propose *meta-buffer* to store a series of informative high-level thoughts, namely *thought-template*, distilled from the problem-solving processes across various tasks. Then for each problem, we retrieve a relevant thought-template and adaptively instantiate it with specific reasoning structures to conduct efficient reasoning. To guarantee the scalability and stability, we further propose *buffer-manager* to dynamically update the meta-buffer, thus enhancing the capacity of meta-buffer as more tasks are solved. We conduct extensive experiments on 10 challenging reasoning-intensive tasks, and achieve significant performance improvements over previous SOTA methods: 11% on Game of 24, 20% on Geometric Shapes and 51% on Checkmate-in-One. Further analysis demonstrate the superior generalization ability and model robustness of our BoT, while requiring only 12% of the cost of multi-query prompting methods (e.g., tree/graph of thoughts) on average. Notably, we find that our Llama3-8B + BoT has the potential to surpass Llama3-70B model. Our project is available at https://github.com/YangLing0818/buffer-of-thought-llm

1 Introduction

A series of Large Language Models (LLMs) [1–5] like GPT-4 [3], PaLM [2] and LLaMA [6, 7] have showcased the impressive performance in various reasoning tasks. In addition to scaling up the model size to improve the reasoning performance, there are more effective prompting methods that further enhance the functionality and performance of LLMs. We divide these methods into two categories: (i) **single-query reasoning:** these methods [8–10] usually focus on prompt engineering and their reasoning process can be finished within a single query, such as CoT [8] that appends the input query with 'Let's think step by step' to produce rationales for increasing reasoning accuracy, and Few-shot Prompting [11, 12, 9, 13] which provides task-relevant exemplars to assist the answer generation; (ii) **multi-query reasoning:** these methods [14, 15] focus on leveraging multiple LLM queries to elicit different plausible reasoning paths, thus decomposing a complex problem into a series of simpler sub-problems, such as Least-to-Most [16], ToT [14] and GoT [17].

However, both kinds of methods face some limitations: (1) single-query reasoning usually requires prior assumption or relevant exemplars of reasoning process, which makes it impractical to manually design them task by task, thus lacking universality and generalization; (2) Due to the recursive expansion of reasoning paths, multi-query reasoning is usually computationally-intensive when finding a unique intrinsic structure underlying the reasoning process for each specific task; (3)

*Equal Contribution. ✉ yangling0818@163.com

Preprint. Under review.

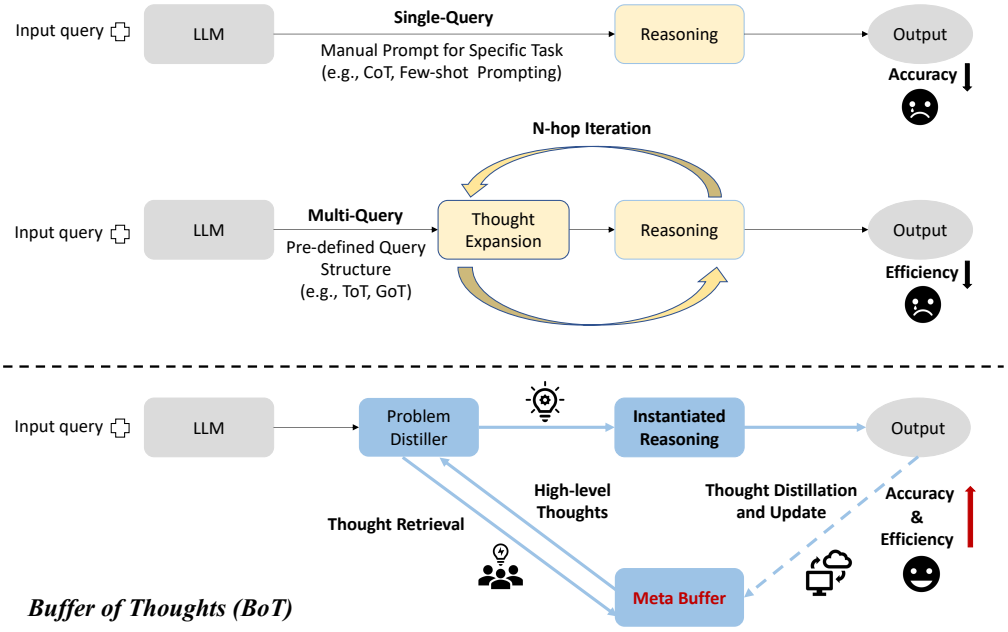

Figure 1: Comparison between single-query [8, 11], multi-query [14, 17], and (c) our BoT methods.

Both single-query and multi-query reasoning processes are limited by their designed exemplars and reasoning structures, and they neglect to derive general and high-level guidelines or thoughts from previously-completed tasks, which are informative for improving efficiency and accuracy when solving similar problems.

To address these limitations, we propose Buffer of Thoughts (BoT), a novel and versatile thought-augmented reasoning framework aimed at enhancing reasoning accuracy, efficiency and robustness of LLMs across various tasks. Specifically, we design *meta-buffer*, a lightweight library housing a series of universal high-level thoughts (*thought-template*), which are distilled from different problem-solving processes and can be shared across tasks. Then, for each problem, we retrieve a relevant thought-template and instantiate it with specific reasoning structure for efficient thought-augmented reasoning. In order to guarantee the scalability and stability of our BoT, we further propose *buffer-manager* to dynamically update the meta-buffer, which effectively enhances the capacity of meta-buffer as more tasks are solved.

Our method has three critical advantages: (i) **Accuracy Improvement:** With the shared thought-templates, we can adaptively instantiate high-level thoughts for addressing different tasks, eliminating the need to build reasoning structures from scratch, thereby improving reasoning accuracy. (ii) **Reasoning Efficiency:** Our thought-augmented reasoning could directly leverage informative historical reasoning structures to conduct reasoning without complex multi-query processes, thus improving reasoning efficiency. (iii) **Model Robustness:** The procedure from thought retrieval to thought instantiation is just like the human thought process, enabling LLMs to address similar problems in a consistent way, thus significantly enhancing the model robustness of our method. Our empirical studies demonstrate that Buffer of Thoughts significantly improves precision, efficiency, and robustness over a diverse array of tasks. Here, we summarize our contributions as follows:

1. We propose a novel thought-augmented reasoning framework Buffer of Thoughts (BoT) for improving the accuracy, efficiency and robustness of LLM-based reasoning.

2. We propose meta-buffer for store informative high-level thoughts distilled from different problems, and adaptively instantiate each thought template to address each specific task.

3. We design buffer-manager to distill thought-templates from various solutions, and is continually improves the capacity of meta-buffer as more tasks are solved.

4. We conduct extensive experiments on 10 challenging reasoning-intensive tasks. Our BoT achieves significant performance improvements over previous SOTA methods: **11% on Game of 24, 20% on Geometric Shapes and 51% on Checkmate-in-One**, while requiring **only 12% of the cost** of multi-query prompting methods on average.