Iterative Refinement. Through boosting mechanism, $\mathbf{F}^{1,\ldots,t}$ is employed to iteratively enhance the prompt, resulting in $\mathbb{I}^{t+1}\left(S, X, Q, \mathbf{F}^{1,\ldots,t}, \{G_i\}\right)$ for the $(t+1)$-th iteration. As the iterations progress, $\mathbf{F}^{1,\ldots,t}$ may encompass many typical, unreasonable thought chains alongside those closer to a solution, all with well-defined analysis outcomes. Therefore, even when starting with a simple prompt, BoT iteratively refines this prompt to produce the correct reasoning steps leading to the accurate solution. After T iterations, we utilize the $\mathbb{I}^{t+1}$ as input prompt for the LLM to gain the final answer.

4 EXPERIMENTS

Datasets. Experiments are performed on benchmark datasets with diverse mathematical problems, including `MMLU` Hendrycks et al. (2021a), `SVAMP` Patel et al. (2021), `GSM8K` Cobbe et al. (2021), `AQuA` Ling et al. (2017) and `MATH` Hendrycks et al. (2021b). Besides, we include a challenging mathematical reasoning task, Game of 24 Yao et al. (2024), where the goal is to use four numbers and basic arithmetic operations (addition, subtraction, multiplication, and division) to obtain 24 in 1 equation. Thus, the solution includes 3 intermediate steps.

Competitors. Apart from the benchmark approach, standard Input-output (IO), the comparison approaches include Chain-of-thought (CoT) Wei et al. (2022), CoT-SC Wang et al. (2022) and Complex CoT Fu et al. (2022), in which the input prompt contains a few-shot examples (8) with human annotations. Also, BoT is also compared with related works, such as Tree of thoughts (ToT) Yao et al. (2024) with the breadth limit 5, Progressive-Hint Prompting (PHP) Zheng et al. (2023), and the state-of-the-art CSV Zhou et al. (2023a).

Large Language Models. We conduct experiments on the two most recent models: GPT-4 OpenAI (2023) and Llama2 Touvron et al. (2023). GPT-4 is accessed via OpenAI APIs, while the llama-2-13b-chat model is downloaded from MetaAI to perform experiments locally. To construct the heterogeneous tree thought structures, BoT randomly chooses the temperature from the range of $[0.2, 0.4, 0.6, 0.7, 0.9, 1.1, 1.5]$ and the top_p from the range of $[0.1, 0.3, 0.5, 0.7, 0.9]$.

Settings. If not explicitly stated, BoT, in all experiments, performs $T = 10$ iterations of running and builds $M = 15$ thought structures, each being a weighted binary tree because this tends to achieve optimal results. Besides, for those benchmark datasets, we set the depth of the tree to be 5 while the corresponding depth in Game of 24 is 3. BoT+CoT means our simple prompt includes 5 examples from CoT Wei et al. (2022). In the ablation study, when no *experience* is accumulated in BoT, 8 examples of CoT will be provided in the prompt.

Metrics. All experiments report the Solve Rate (%) of the task as the evaluation results. To extract target answers from the output $\overline{z}^{T}_{1\ldots n}$ of BoT, we specifically set the formatted description of the answer for LLMs. For commonly used datasets, the desired answer format is "The answer is: ." For the Game of 24, we utilize "Step idx, Current set: , Selected two numbers: , Operation: , Computed new number: , Remaining numbers: , New set: ". Thus, we directly compare the ground truth with the number presented in the new set. Following ToT Yao et al. (2024), we report the Solving Rate across 100 hard games as the metric.

4.1 MAIN RESULTS

The primary experimental results are summarized in Table. 1 and Fig. 3, where we present insights into the overall performance of BoT. Our findings indicate that the proposed BoT with Boosting mechanism 1). obtains competitive problem-solving rates in most datasets without human annotations; 2). is capable of reaching a new state-of-the-art on `GSM8K` and `AQuA` when provided with CoT examples. However, experimental results also demonstrate that BoT heavily relies on *experience*, thus is sensitive to the ability of LLMs.

Specifically, in Table. 1, BoT, starting from a simple initial prompt and performing basic chatting, eventually obtains a `GSM8K` solve rate 0.1% higher than the current state-of-the-art (SOTA) CSV Zhou et al. (2023a), which heavily relies on code interpreter of GPT-4. Considering `AQuA`, BoT is 2.5% higher than SOTA. This demonstrates that by adding error analysis and advice to the prompt without human annotations, LLMs are able to perform well on complex reasoning. The main reason is that a simple prompt can be iteratively refined by accumulating prior *experience* towards accurate problem-solving. After including CoT examples in the prompt, BoT+CoT outperforms SOTA by