where the `chain_prompt` is the aggregated thought chain $\overline{z}_{1...n}$.

chain feedback format: Can this reasoning chain complete the task and reach the target correctly by executing its reasoning steps? why? Write a analysis report with conclusion under 'Anlysis Report:'.

step feedback format: For each reasoning step, please provide a detailed analysis of whether the current step is a logical inference of the previous step and whether the reasoning step is beneficial to the correct solution. For each reasoning step with errors, please provide an error report and the corresponding advice on revision. For each reasoning step, please provide recommendation or rejection descriptions. Comments should be brief and follow the format: Reasoning step $\langle idx \rangle$. Analysis report: . Advice: . Recommendation or Reject description: .

confidence feedback format: What is your confidence score on these your evaluations and comments? Please select one value from [0.1, 0.3, 0.5, 0.7, 0.9, 1.0]. The score should be placed after 'Confidence score:' for users to read."

With the feedback prompt, LLMs generate reasoning experience $\mathbf{F}^t$ containing conclusion and analysis on the reasoning chain and each reasoning step.

A.3 Reasoning Pipeline

To facilitate the understanding of the proposed Boosting of Thoughts, we summarize the reasoning pipeline in Algorithm Table 1. The source code for this pipeline can be found in the file *examples/BoostingOfThought/BoT_core.py*.

Algorithm 1: Main reasoning pipeline of BoT

Input: Number of iterations T, Number of tree structures M, Question Q.
Output: Aggregated chain $\overline{z}^T_{1...n}$.

1 Initialize a simple prompt $\mathbb{I}^0\left(S, X, Q, \mathbf{F}^0, \{G_i\}\right)$ where $\mathbf{F}^0$ will be an empty string.
2 **for** *each iteration* $t = 1, 2, ..., T$ **do**
3 Use LLMs with the prompt $\mathbb{I}^{t-1}\left(S, X, Q, \mathbf{F}^{t-1}, \{G_i\}\right)$ to create M heterogeneous tree thought structures through Thought Structure Generation.
4 Extract thought chains $\left\{z_{i=1}^{n^m}\right\}_{m=1}^{M}$ from the M thought structures where each $z_{i=1}^{n^m}$ is the best thought chain of m-th tree structure.
5 Aggregate $\left\{z_{i=1}^{n^m}\right\}_{m=1}^{M}$ into a single thought chain $\overline{z}^t_{1...n}$ by using either Best-First Aggregation or Greedy aggregation.
6 Perform Thought Chain Analysis on $\overline{z}^t_{1...n}$ with LLMs to obtain the feedback, which is combined with $\overline{z}^t_{1...n}$ to obtain *experience* $\mathbf{F}^t$.
7 Update the prompt by accumulating $\mathbf{F}^t$, leading to $\mathbb{I}^t\left(S, X, Q, \mathbf{F}^{t-1,t}, \{G_i\}\right)$.
8 **end**
9 Obtain the solution $\overline{z}^T_{1...n}$.

B Insights for Boosting of Thoughts

Boosting of Thoughts derives from our insights that the reasoning ability of large language models (LLMs) for addressing mathematical problems comes directly from experience, which contains the accumulation of the analysis and advice on previous mistakes. Once the prompt embraces valid historical reasoning experience to be recalled by LLMs before performing reasoning, the produced reasoning steps are generally more logical and reasonable, as shown in the comparison between Table 5 and 6. Such insights also made us consider that LLMs do not need to rely heavily on a well-prepared prompt with human annotations (a few chains of thought demonstrations as exemplars in prompts) for each task. Yet, as LLMs are able to learn from experience, we can start from a simple prompt without examples or manually designed content to gradually collect experience during the reasoning process. Eventually, by accumulating experiences in the prompt, LLMs achieve strong reasoning toward addressing complex problems. With these insights, the Boosting of Thoughts is designed as an automated prompting framework, which iteratively collects an ensemble of trial-and-error reasoning experiences for problem-solving with LLMs. We argue that the proposed BoT is not