



Figure 2: The overview of the pipeline in each iteration of BoT. To show how boosting is achieved in this *experience*-driven iteration process, we present detailed intermediate results obtained from an experiment on ChapGPT-4 on the Game of 24 dataset. Given Q : “The given four numbers are: 2, 4, 5, 5”, BoT performs three stages sequentially. With the simple prompt \mathbb{I}^t as input, The Thought Structures Generation (Stage 1) outputs massive heterogeneous tree thought structures. Thought Structures Aggregation (Stage 2) aggregated them into a thought chain $\bar{Z}_{1...n}$, which is analyzed in Stage 3 to produce *experience* to further enhance the prompt.

Instead of pre-preparing examples in the prompt, a more adaptive way is to design prompts to guide the LLMs to gradually generate the thought z_i during the reasoning process. This can be formalized as $z_i \sim p_\theta(z_i | \mathbb{I}(z_{1...i-1}, X, Q))$. Finally, the solution is formalized as $y \sim p_\theta(y | \mathbb{I}(z_{1...n}, X, Q))$. The representative approach, ToT [Yao et al. \(2024\)](#), further extends this sequential reasoning steps into a tree structure in which C next-step thoughts can be generated. Thus, the *thought structure* can be chain or tree.

3.2 FRAMEWORK

Existing literature that aims to generate the prompt with correct CoT examples or design delicate thought generation structures has three limitations. First, ineffective thoughts in those approaches are generally ignored or discarded. However, a human, who is not an expert in one field, particularly relies on analyzing previous errors to collect more experience to perform correctly on the next try. Second, they are less scalable because, for each task, an example of generating the next thoughts, such as $\mathbb{I}(z_1 | z_0, X, Q)$, should be provided in the prompt. Finally, the thought structure, such as the tree [Yao et al. \(2024\)](#), is generated to be overly complex to explore more reasoning steps for a better solution. This is largely due to the obtained solution may not be further revised.

In this paper, we argue that the prompt can be enhanced by continuously collecting the analysis of LLMs on those ineffective thoughts – wrong reasoning steps in a chain of thought. Therefore, even a simple prompt, such as $\mathbb{I}(X, Q)$, potentially leading to ineffective thoughts, can be progressively refined by relying on such analysis to gain powerful thoughts toward the solution.

We propose Boosting of Thoughts (BoT), an automated prompting framework incorporating, which achieves prompt *boosting* with an *experience*-driven iteration process commencing with a **simple prompt**. As summarized in Fig. 2, each iteration t of BoT includes three stages. The Thought Structures Generation stage is able to fully explore reasoning chains generated by LLMs with the input prompt \mathbb{I}^t . In the second stage, these thought structures are aggregated to form a reasoning chain, which is to be analyzed by LLMs in the third stage to produce feedback containing error reports and detailed revision advice. Combining the aggregated reasoning chain with the feedback results in a new *experience*, denoted as \mathbf{F}^t . Thus, the prompt is enhanced by accumulating these experiences $\mathbf{F}^{1...t}$ over iterations.