

Simple Prompt. For any task, in iteration $t = 0$, we create a simple initial prompt $\mathbb{I}^0(S, X, Q, \mathbf{F}^0, \{G_i\})$, where S represents task-agnostic descriptions while the terms X and Q respectively denote the task information and the question. The *experience* part of the prompt is denoted as \mathbf{F}^0 , which should be empty at the beginning. $\{G_i\}$ is a placeholder that is waiting to be filled during building thought structures. In other words, when generating the next thought z_i , $\{G_i\}$ will be substituted with the preceding chain of thoughts $z_{1..i-1}$.

Thought Structures Generation. After collecting *experience* \mathbf{F}^{t-1} , the prompt in the iteration t can be $\mathbb{I}^t(S, X, Q, \mathbf{F}^{1..t-1}, \{G_i\})$. Based on this prompt, BoT generates M thought structures in parallel. BoT is inherently capable of embracing any thought structure, such as the chain Wei et al. (2022) or tree Yao et al. (2024) structure. Considering the exploration of reasoning steps and experimental results, we investigate the tree thought structure. However, BoT introduces two novel modifications to make it better suited for the boosting framework.

- **Weighted Binary Tree.** With a simple prompt in each round, BoT builds the weak thoughts structured in low complexity as they can be further revised in the boosting mechanism. Thus, each thought structure of BoT is a shallow weighted binary tree. For simplicity, we retain the notation $z_{1..i-1}$ to represent the thoughts from the root to the parent of node i . In addition to providing each node i with one thought z_i and its thought evaluation score $V_i \sim p_\theta(z_{1..i}, \mathbb{I}_a, X, Q)$, we incorporate the edge score $V_{i-1,i} \sim p_\theta(z_{i-1}, z_i, \mathbb{I}_e, X, Q)$ between a child node and its parent node, where \mathbb{I}_a and \mathbb{I}_e refer to the instructional descriptions for thought and edge evaluations. $V_{i-1,i}$ represents the LLMs' confidence level in generating this reasoning step. Thus, the next thought generation of BoT in this tree structure is formalized as $p_\theta(z_i | (V_{i-1,i}, V_i, \mathbb{I}^t, X, Q))$.
- **Tree Heterogeneity.** Unlike ToT Yao et al. (2024), which seeks to search for a solution in one large and complex tree, BoT aims to build highly heterogeneous tree thought structures. As a result, complete reasoning chains with various logical in trees of BoT are subsequently assessed as *experience*. Therefore, to increase heterogeneity, thought structure generation embraces different tree growth strategies, such as level-wise growth and leaf-wise growth. The former emphasizes exploration but less exploitation Chen & Guestrin (2016), while the latter does the opposite Ke et al. (2017). Thus, the leaf-wise strategy tends to continue reasoning from the current best thought to reach a better final thought as compared to level-wise growth, but it also tends to get monotonous reasoning chains. Besides, different temperature and Top_p settings of LLMs are applied. Finally, we use a small max_depth value in BoT and label a node as a leaf when its $V_{i-1,i}$ and V_i values are outside the specified range $[0.3, 0.8]$.

Thought Structures Aggregation. Upon obtaining M thought structures, BoT aggregates them into one thought chain denoted as $\bar{z}_{1..n}$. To achieve this, for each thought structure with index m , BoT first selects the chain with the highest evaluation score as $z_{1..n_m}^m := \arg \max_{z_{1..n} \in Z^m} \sum_{i=1}^n V_i + V_{i-1,i}$ where Z^m denotes the set of all thought chains of m -th tree. Subsequently, two strategies exist to obtain $\bar{z}_{1..n}$.

- **Best-First Aggregation.** BoT relies on $\arg \max_{z_{1..n} \in \{Z^m\}_{m=1}^M} \sum_{i=1}^n V_i + V_{i-1,i}$ to choose the best one as $\bar{z}_{1..n}$ from M thought structures. This algorithm is fast but may lead to an unreasonable chain that is hard to guide the following refinement.
- **Greedy Aggregation.** BoT is allowed to perform a greedy search on $\{Z^m\}_{m=1}^M$ to assemble a new thought chain that may not exist and is globally optimal. Starting from the initial thought, generally the root node of the tree, BoT obtains $\bar{z}_1 = \arg \max_{z_j \in \{z_1^m\}_{m=1}^M} V_j + V_{j-1,j}$. Subsequently, to obtain \bar{z}_i for \bar{z}_{i-1} , BoT searches all thoughts where the previous step is \bar{z}_{i-1} in $\{Z^m\}_{m=1}^M$.

Thought Chain Analysis. To gain insights into what should be adjusted to enhance the prompt to generate better thoughts, BoT utilizes the self-evaluation ability Weng et al. (2023) of LLMs to assess $\bar{z}_{1..n}$. Specifically, with the prompt $\mathbb{I}_f^t(\bar{z}_{1..n}, X, Q)$ as input, LLM outputs a feedback paragraph containing issues report of this thought chain $\bar{z}_{1..n}$ and detailed advice. This feedback will be added to $\mathbf{F}^{1..t-1}$ as a new *experience* in thought generation, resulting $\mathbf{F}^{1..t}$.