Blind Spot Navigation in Large Language Model Reasoning with Thought Space Explorer

Jinghan Zhang[1], Fengran Mo[2], Tharindu Cyril Weerasooriya[3],
Xinyue Ye[4], Dongjie Wang[5], Yanjie Fu[6], Kunpeng Liu[1*]
[1]Clemson University, [2]Université de Montréal, [3]Center for Advanced AI, Accenture,
[4]The University of Alabama, [5]University of Kansas, [6]Arizona State University
{jinghaz,kunpenl}@clemson.edu

Abstract

Large language models have shown strong reasoning capabilities through chain-structured methods such as Chain-of-Thought. Recent studies optimize thought structures by generating parallel or tree-like structures, switching long and short reasoning modes, or aligning reasoning steps with task performance. However, these approaches mainly rely on previously generated logical directions of the chains, which ignore the unexplored regions of the solution space. Such a phenomenon is denoted as *blind spots*, which limit the diversity and effectiveness of the reasoning process. To this end, we propose the "Thought Space Explorer" (TSE), a framework for navigating and expanding thought structures to overcome blind spots in LLM reasoning. Our TSE first identifies key nodes with high impact, then generates new nodes by integrating information from multiple chains. Finally, it extends new branches through connection strategies. We conduct a series of experiments on math and QA benchmarks. Compared to existing baseline methods, TSE improves the accuracy of both the final answer and intermediate reasoning steps, while maintaining a better effectiveness-efficiency trade-off for practical deployment.

1 Introduction

Recent advances in large language models (LLMs) have shown great potential in solving complex tasks with reasoning capabilities (Huang and Chang, 2022; Patterson et al., 2022; Achiam et al., 2023; Mao et al., 2023; Dutta et al., 2025) by guiding the LLMs to logically solve the complex task step-by-step. A common practice is to design the Chain-of-Thought (CoT) (Kojima et al., 2022; Yang et al., 2025) to boost reasoning capabilities by evolving the thinking from single outputs to intermediate steps reasoning.

Existing studies (Wang et al., 2022; Yao et al., 2024; Zhang et al., 2024d; Besta et al., 2024; Pandita et al., 2025) attempt to develop various thought structures with multiple chains or branches of thought on top of CoT to arouse the reasoning ability of LLMs. Compared with direct output and CoT, the core advantage of thought structures enables models to explore the solution space of a task from local to global (Hao et al., 2023). For example, as presented in Figure 1, thought structures may initiate exploration from two distinct points *"specialty"* and *"industry"*. Such exploration allows LLMs to generate diverse paths to solutions and thus enhances the model's reasoning capacity. Moreover, the diverse structures can enable models to perform forward and backward evaluations within the explored thought space toward the optimal solution, i.e., a more effective reasoning thought path.

A series of studies are conducted to optimize thought structures with various aspects, including generating parallel thought (Wang et al., 2022), constructing tree-structured reasoning topologies on top of CoT (Yao et al., 2024), and fine-tuning the LLMs with direct preference optimization (DPO) to align thought steps of CoT with task performance (Zhang et al., 2024c), etc. The key idea of these studies is to compare multiple responses or extend existing chains (e.g., *"Coffee over Drone"* or *"Coffee industry → Coffee bottle industry"* as shown in Figure 1) to obtain a better thought chain.

However, these approaches do not explore regions of the solution space that the model itself has never considered. We refer to such unexplored regions as the *blind spots* of LLMs. These blind spots are the areas in the reasoning space that are systematically overlooked, because the model's generation is biased and always leads to the previously explored paths (Zhang et al., 2024a; Sprague et al., 2025; Liu et al., 2025). Merely generating more chains does not enable LLMs to conceive of con-

---
*Corresponding author.

tent previously unthought of. As described in Figure 1, over-generated chains tend to repeat prior thought patterns, leading to two main issues: (1) the absence of feasible solutions—when such solutions lie in blind spot regions, repeatedly filtering or extending existing paths may converge to a local optimum (e.g., exploring only from a *coffee* perspective); and (2) insufficient diversity—especially for open-ended questions, where existing methods have limited impact on exploring the thought space, and excessive extension or filtering might even reduce the diversity of responses (e.g., discarding feasible solutions or creating redundancy through repetitive thinking).

To address these issues, we propose the **T**hought **S**pace **E**xplorer (TSE), a novel framework designed to expand and optimize thought structures. The TSE starts from thought paths already explored and guides the model to explore hidden solution spaces because the existing thought structures often already contain feasible solutions or crucial information pointing towards such solutions. To enhance efficiency and precision, further exploration of the model starts from thought nodes within explored solutions, which ensures that the reasoning process is not a blind exploration but a deeper inquiry based on verified insights.

To identify key points of information from existing thoughts, as shown in Figure 1, we first quantify each thought node's contribution to the conclusion during the model's reasoning process to select key nodes (e.g., in Chain 2, the details about *"drones and delivery"* serve as key information leading toward *"logistics industry"*). We adopt relative gradients as relative importance to select key nodes. Based on these key nodes, the model then generates new thought nodes and proceeds with deeper reasoning in new directions from "original nodes" to "new nodes", facilitating exploration of the solution space through the thought structure. Finally, we perform collaborative reasoning across the entire thought structure to generate the output. Considering the visibility of parameters in LLMs, we reformulate the key steps of this method using LLMs' semantic and evaluation capabilities for black-box or gradient-invisible models.

We evaluate the effectiveness of TSE on four reasoning benchmarks on Qwen3 series models (Yang et al., 2025) and the results show that TSE significantly improves the performance of thought structures compared with existing methods. We further discuss the impact of TSE on path accuracy beyond final output accuracy, the diversity enhancement and the token cost trade-off.

Our contributions include:

- We propose TSE reasoning framework to expand thought structures for exploring solution spaces to alleviate the impact of blind spots for LLM reasoning.

- We investigate various strategies to prioritize and refine the thought structure by identifying the importance of the node in the thought structure.

- Experimental results on three specific reasoning tasks indicate the effectiveness of our TSE compared with the existing reasoning methods without exploring thought structure.

2 Related Work

2.1 LLM Reasoning Structures

The most straightforward method to address reasoning tasks is to generate a conclusion through one-step thinking. However, the LLM might overlook essential intermediate steps and generate incoherent logic or incorrect conclusions (Chu et al., 2023). The advent of CoT (Wei et al., 2022; Wang and Zhou, 2024) optimizes the reasoning step by connecting distinct thoughts into a coherent sequence (Li et al., 2024). Although CoT can improve transparency and coherence, its singular structure limits its capability to handle more complex logical relationships (Jin and Lu, 2024; Sprague et al., 2025). To this end, some studies develop structured reasoning methods, such as self-consistent CoT and Tree-of Thought. (Wang et al., 2022; Zhang et al., 2024d; Yao et al., 2024; Liu et al., 2024; Mo and Xin, 2024; Zhang et al., 2024b,a). These sophisticated thought structures enhance the consistency and systematic nature of reasoning by expanding the ability of the model to manage diverse logical relationships (Xia et al., 2024; Stechly et al., 2024). Thought structures offer distinct advantages by maintaining coherence and depth while increasing the diversity and flexibility of reasoning paths (Liang et al., 2024). However, the reasoning chains within these structures are highly repetitive and thus reduce generation and selection efficiency.

2.2 Thought Structure Optimizations

To further enhance the capabilities of thought structures, recent researches focus on two main op-
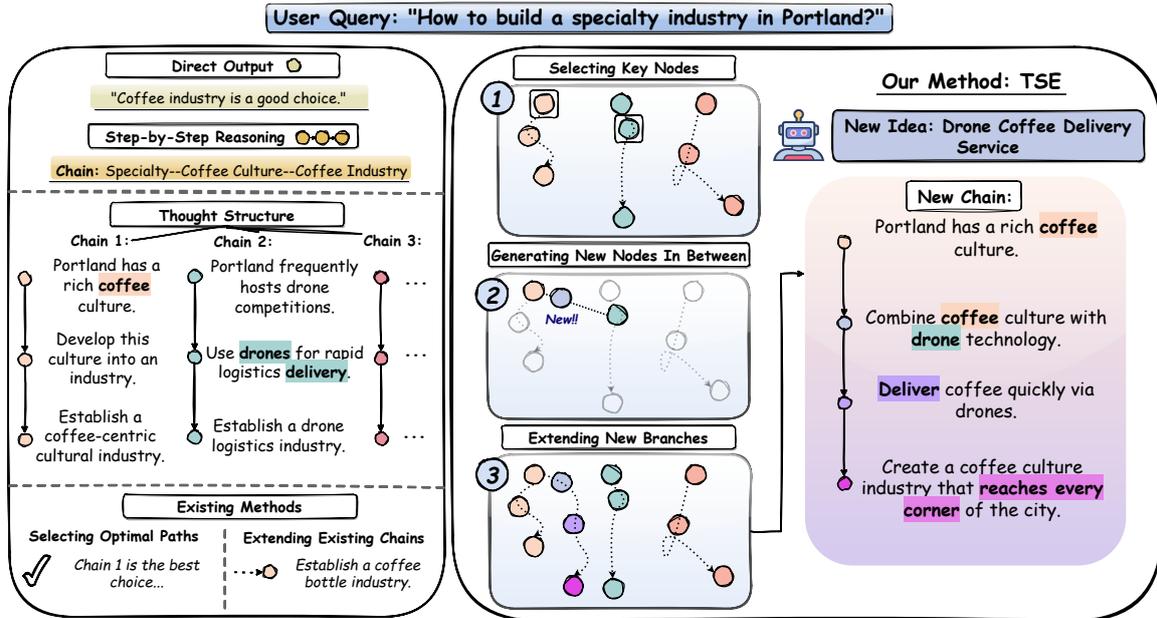
Figure 1: Thought structure optimization through TSE. On the left side, we showcase traditional thought structures and optimization methods, where the LLMs' generation may limited by its thought pattern. On the right side, we show how TSE expands thought structure through a three-step generation of branches. TSE guides LLMs to explore the blind spots between previous thought paths.

timization strategies. The first is selecting optimal paths within the structures (Feng et al., 2023; Long, 2023; Hao et al., 2023; Shinn et al., 2023; Jung et al., 2022). By choosing optimal paths, the model filters the irrelevant and low-quality branches and globally searches for correct or optimal solutions, thus enhancing reasoning efficiency and accuracy. The second is expanding reasoning depth and breadth (Zhu et al., 2022; Besta et al., 2024; Gao et al., 2024; Zhang et al., 2024b,c; Hou et al., 2024; Zhang et al., 2025). By deepening and widening the thought structure, the model can explore a broader array of possibilities and perspectives, thus improving its understanding and capability to solve complex issues (Chen et al., 2025; Pan et al., 2025; Wang et al., 2025). However, these methods might be limited to previously explored spaces and directions, thus failing to adequately investigate the blind spots within the thought space of the model. Different from them, we aim to expand the depth and breadth of thought structures by actively exploring the cognitive blind spots of the model in thought space.

3 Methodology

To expand and optimize thought structures for effective exploration of reasoning spaces, we introduce **TSE**, a self-expansion and exploration

method that allows language models to proactively address deficiencies in reasoning processes and explore new reasoning directions with limited steps of generation. In a step-by-step reasoning scenario, we implement the TSE through three steps: **(1) Key Node Selection and Generation**, aim to select the nodes that are most influential to exploration directions based on the crucial information contained previously, then the model generates a new node to integrate the insights of two key nodes for new exploration directions; **(2) Connection and Expansion**, to connect and expand the reasoning paths, and the new paths explore potential new directions of solutions from the new node; and **(3) Collaborative Reasoning** to address deficiencies in the model's ability to synthesize and integrate diverse reasoning paths in different directions.

3.1 Problem Formulation

Given a specific reasoning task $\mathcal{Q}$, we apply a large language model (LLM) $\mathcal{L}$ to a structured reasoning process $\mathcal{S}$. This structure consists of multiple reasoning sentences as thought nodes, which are connected sequentially. The set of all thought nodes is denoted as $\mathcal{T}$, where each node T_{ij} represents the j-th reasoning step in the i-th thought chain. The thought structure $\mathcal{S}$ can be viewed as a directed graph consisting of vertices (thought nodes) $\mathbf{V}$ and edges (connections between consecutive nodes) $\mathbf{E}$.
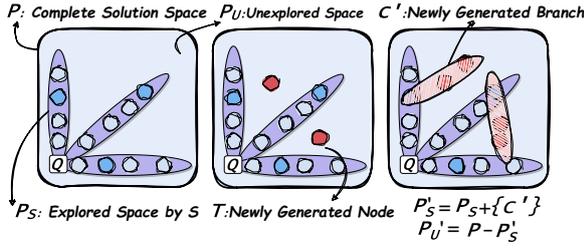
Figure 2: Illustration of solution space exploration via our TSE method. By generating new branches of solutions, the explored space of solutions expands.

Formally, we define them as:

$$\mathbf{V} = \bigcup_{i=1}^{N} \bigcup_{j=1}^{K_i} \{T_{ij}\}, \quad K_i = |C_i|,$$

$$\mathbf{E} = \bigcup_{i=1}^{N} \bigcup_{j=1}^{K_i-1} \{(T_{ij}, T_{i,j+1})\}, \tag{1}$$

where N is the number of thought chains, and K_i is the number of nodes contained in chain C_i. Then, the structure $\mathcal{S}$ is defined as:

$$\mathcal{S} = (\mathbf{V}, \mathbf{E}), \quad C_i = \langle T_{i1}, T_{i2}, \ldots, T_{iK_i} \rangle \tag{2}$$

For a specific task $\mathcal{Q}$, the complete reasoning solution space $\mathcal{P}$ encompasses all possible reasoning paths C_i (thought chains) that can potentially solve $\mathcal{Q}$. As shown in Figure 2, the space that has been explored by the generated thought structure $\mathcal{S}$ is denoted as $\mathcal{P}_S$, and the remaining unexplored space is denoted as $\mathcal{P}_U$, with $\mathcal{P}_S \cup \mathcal{P}_U = \mathcal{P}$.

Our goal is to actively expand the thought structure $\mathcal{S}$ by generating new reasoning branches C' to explore previously untouched subspace $\mathcal{P}_{\mathcal{U}}$. In this way, we increase the likelihood of discovering correct and novel solutions. Formally, we define the optimization objective as:

$$\max_{\mathcal{S}'} J(\mathcal{S}', \mathcal{Q}) \tag{3}$$

where J is the reasoning performance metrice and $\mathcal{S}'$ is the expanded thought structure.

3.2 Key Node Selection and Generation

We aim to select the most impactful nodes from the existing thought structure $\mathcal{S}$ for expansion. Intuitively, these nodes should contain crucial information for the task and satisfy two requirements: (i) enabling effective exploration of promising regions in the solution space by initiating expansion from these key nodes, thus increasing the likelihood of

discovering viable solutions; and (ii) reducing error propagation by conducting additional analysis and verification on these critical nodes, which often represent potential sources of mistakes.

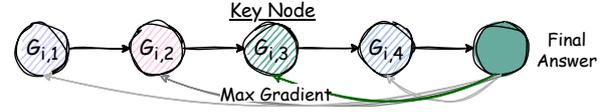#### 3.2.1 Gradient-based Selection

Figure 3: Gradient-based selection.

When the internal states and gradients of the model $\mathcal{L}$ are available, we access the representation of each thought node T_{ij} from the hidden states of $\mathcal{L}$:

$$\mathcal{L} : T_{ij} \mapsto \mathbf{v}_{ij} \in \mathbb{R}^d. \tag{4}$$

The representation of the conclusion node $\mathbf{v}_{iK_i}$ is mapped to the output space as the model's prediction $\hat{y}_i$ as:

$$\hat{y}_i = f(\mathbf{v}_{iK_i}), \tag{5}$$

where $f(\cdot)$ denotes the mapping from the representation space to the output space, and $\hat{y}_i$ is typically a textual answer or decision for task $\mathcal{Q}$.

The self-information loss L_i is a common practice to evaluate the model's confidence in its predictions (Wang and Feng, 2021), where higher confidence corresponds to lower loss values. Thus, we calculate the partial derivative of the loss $\mathbf{g}_{ij}$ with respect to each node's representation $\mathbf{v}_{ij}$ and the Euclidean norm of its gradient G_{ij} to measure the importance of the nodes, as shown in Figure 3. Then, we apply a normalization to determine the relative importance I_{ij} of each node for a consistent and comparative analysis of node importance across different chains within the thought structure $\mathcal{S}$ as:

$$L_i = -\log P(\hat{y}_i \mid \mathbf{v}_{iK_i}),$$

$$\mathbf{g}_{ij} = \frac{\partial L_i}{\partial \mathbf{v}_{ij}}, \quad G_{ij} = \|\mathbf{g}_{ij}\|_2. \tag{6}$$

To enable comparison across nodes and chains, we normalize the gradient magnitudes within each chain C_i as:

$$I_{ij} = \frac{G_{ij}}{\sum_{k=1}^{K_i} G_{ik}}. \tag{7}$$

We regard nodes with larger I_{ij} values as key nodes, as perturbations at these nodes typically have the greatest impact on the final prediction. Each T_i^{key} corresponds to the most influential node selected

from chain C_i. These key nodes serve as the starting points for generating new reasoning branches in the subsequent expansion phase.

3.2.2 Generating New Nodes

With the selected key nodes, the next step is to use them as conditional information for generating new thought nodes. We generate the new nodes by combining two key nodes from the set T^{key}, denoted as T_i^{key} and T_l^{key}. Given such a pair, the model generates a new candidate node T_{il}^1 as

$$T_{il}^1 = \mathcal{L}(T_i^{\text{key}}, T_l^{\text{key}}), \quad i, l \in [1, N], i \neq l. \quad (8)$$
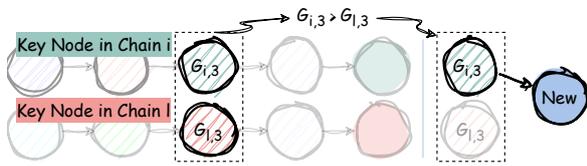
3.3 Connection and Expansion

Figure 4: Gradient-based key node connection.

To integrate the new node into the thought structure, we must decide which of the two key nodes (T_i^{key} or T_l^{key}) should serve as the connection point for extending a new branch. Therefore, we select the connection node as the key node that exhibits stronger semantic relevance to the newly generated node and contributes more significantly to reasoning. The relative gradient selection chooses the connection node between T_i^{key} and T_l^{key} by comparing their importance indices $I(T_i^{\text{key}})$ and $I(T_l^{\text{key}})$. Formally, we first select the connection node and then initialize a new branch as:

$$\begin{aligned} T_c = \arg \max_{T_{key} \in T_i^{key}, T_l^{key}} I(T_{key}), \\ C' = \langle T_c, T_{il}^1 \rangle. \end{aligned} \quad (9)$$

where C' denotes the new branch initiated from the key node with the higher importance index, as shown in Figure 4. Starting from the newly generated node T_{il}^1, the model $\mathcal{L}$ continues to generate subsequent steps conditioned on T_c. Since T_{il}^1 integrates information from two key nodes, the branch tends to explore novel reasoning directions that were not present in the original chains. The branch is extended until the target depth K is reached. By default, K is inherited from the chain containing T_c (i.e., $K = K_i$ if $T_c = T_i^{\text{key}}$, otherwise $K = K_l$).

3.4 Collaborative Reasoning

Given a task $\mathcal{Q}$ and its complete but unseen solution space $\mathcal{P}$, the model $\mathcal{L}$ generates new thought branches on top of the original thought structure $\mathcal{S}$. During this process, each new branch C' expands the explored subspace $\mathcal{P}_\mathcal{S} \in \mathcal{P}$ by mining potential solutions based on the established structure as:

$$\mathcal{P}_\mathcal{S}' \leftarrow \mathcal{P}_\mathcal{S} \cup \{C'\}, \quad \mathcal{P}_\mathcal{U}' \leftarrow \mathcal{P} - \mathcal{P}_\mathcal{S}'. \quad (10)$$

The refined structure $\mathcal{S}'$, compared to $\mathcal{S}$, explores a larger portion of the solution space with $|\mathcal{P}_\mathcal{S}'| \geq |\mathcal{P}_\mathcal{S}|$. Based on $\mathcal{S}'$, we can integrate both original and newly discovered reasoning paths to form a unified conclusion. We consider all thought chains in the refined structure $\mathcal{S}'$ and use gradient information to recalculate and select the key nodes of each chain. For a key node T_{ik}^{key}, we assign a weight based on its relative contribution to the solution as:

$$w_{ik}^{\text{key}} = \frac{\exp\left(-L_{ik}^{\text{key}}\right)}{\sum_{T_{im} \in T_i^{\text{key}}} \exp\left(-L_{im}^{\text{key}}\right)}, \quad (11)$$

where $L_{i_{key}}$ represents the self-information loss at node $\mathcal{T}_{i_{key}}$, which reflects the model's confidence and potential error at that node. The contribution of each node is represented as v_{ik}^{key}, which is obtained from its embedding through a linear projection. This score quantifies how strongly the node supports the overall reasoning process, capturing factors such as semantic relevance to the task or inference correctness. Then, we compute the collaborative reasoning score by aggregating the weighted contributions of all key nodes across all chains for the given reasoning task $\mathcal{Q}$ as:

$$C(\mathcal{Q}) = \sum_{i=1}^{N} \sum_{T_{ik} \in T_i^{\text{key}}} w_{ik}^{\text{key}} \cdot v_{ik}^{\text{key}}. \quad (12)$$

Finally, the decision D for task $\mathcal{Q}$ is selected as the candidate with the highest collaborative reasoning score as:

$$D = \arg \max_{q \in \mathcal{Q}} C(q). \quad (13)$$

3.5 Semantic Blind Spot Navigation

In gradient-available models, TSE leverages internal gradients to identify and expand key reasoning nodes. However, in many practical scenarios, the proprietary LLMs are only accessible through APIs, where internal states and gradients cannot be observed. To adopt TSE in such cases, we reformulate the key steps of exploration using the

semantic and self-evaluation capabilities of LLMs. This module consists of three components: (i) Self-prompting Selection of key nodes, (ii) Semantic Relevance–based Connection Selection, and (iii) LM-as-a-judge for final reasoning evaluation.

3.5.1 Self-prompting Selection

For a black-box model, we leverage LLM's capabilities to rank nodes within each path by their semantic and logical relevance to the task. Concretely, for a thought chain C_i, the model $\mathcal{L}$ evaluates each node T_{ij} under the task context $\mathcal{Q}$ and assigns an importance score. The most critical node with the highest score is then selected as the key node T_i^{key}:

$$T_i^{\text{key}} = \arg \max_{T_{ij} \in C_i} \text{Rank}_{\mathcal{L}}(T_{ij} \mid \mathcal{Q}), \qquad (14)$$

where $\text{Rank}_{\mathcal{L}}(\cdot)$ denotes the ranking mechanism induced by carefully designed prompts that ask the model to compare nodes according to their semantic contribution and logical necessity.

3.5.2 Semantic Relevance–based Connection Selection

After obtaining the key nodes, we need to decide which one should serve as the connection point when generating new branches. To achieve so, we rely on semantic relevance to enable the model $\mathcal{L}$ to compare the newly generated node T_{il}^1 with its candidate parent nodes T_i^{key} and T_l^{key}, and select the node with stronger semantic alignment as the connection point T_{selected} as:

$$C' = \langle T_{\text{selected}}, T_{il}^1, T_{il}^2, \ldots, T_{il}^K \rangle, \qquad (15)$$

where K denotes the maximum depth of expansion. This ensures that the extended branch remains logically coherent while exploring novel directions.

3.5.3 LLM-as-a-Judge

Finally, for reasoning tasks that require interpretability or involve multiple competing branches, we employ the LLM itself as the judge to evaluate and aggregate the reasoning paths. In this stage, model $\mathcal{L}$ assigns a score to each candidate chain C_q based on specific factors, including soundness, coherence, innovation, and clarity. Then, the judgment scores are produced via an evaluation template as:

$$\text{Score}_q = \mathcal{L}(C_q | \text{Eval Template}). \qquad (16)$$

The final decision D for task $\mathcal{Q}$ is chosen through a voting or max-score mechanism:

$$D = \arg \max_{q \in \mathcal{Q}} \text{Score}_q. \qquad (17)$$

In this way, these steps allow exploration and validation without access to gradients, and guide the model to more diverse and reliable reasoning paths.

4 Experiments

We conduct a series of experiments to evaluate the reasoning performance of TSE. We first compare it with state-of-the-art baseline methods on several widely used benchmarks to test its overall accuracy. Then we investigate how TSE enhances the quality of reasoning paths beyond the final answers to the questions, which includes validating the path accuracy (the effectiveness of the reasoning exploration) and the path diversity (whether the exploration leads the model search to broad paths for the final answer). Finally, we discuss the cost-accuracy trade-off of these methods. For more discuss and comparison of black box model and non-gradient exploration, please refer to 5.

4.1 Experimental Setup

Settings. We evaluate the pass@1 rate on the latest `Qwen3-4B/8B` (Yang et al., 2025) models with sampling temperature fixed at 0.7. Unless otherwise specified, we generate five parallel thought chains with a maximum depth of 5 for each question and use this structure as the basis for TSE. All experiments are conducted on 4 H200 GPUs.

Evaluation Datasets. For evaluation, we select four widely used math and science benchmarks with different level: **GSM8K** (Cobbe et al., 2021), a large collection of grade-school math word problems testing multi-step arithmetic reasoning. **AIME24** and **AIME25** (Math-AI, 2024, 2025), each containing 30 competition-style math problems covering arithmetic, algebra, and geometry from the American Invitational Mathematics Examination; and **GPQA-D**iamond (Rein et al., 2024), a curated subset of GPQA, which contain 198 PhD-level science questions authored by domain experts in physics, chemistry, and biology.

Baselines. We compare TSE method with (1) **Direct** output without thinking mode, (2) **Think** (Yang et al., 2025): output with Qwen3's long CoT thinking mode, (3) **ToT**) (Gomez, 2023): a reasoning method with tree-structure for thoughts, (4) **RATT** (Zhang et al., 2024b; Semnani et al., 2023): a tree-structure reasoning method with RAG with WikiPedia as external knowledge base, and (5) **Self**-Route (He et al., 2025): an automated reasoning path mode switch method. For ToT and

RATT, we set the number of nodes with one generation ($N = 3$). For Self-Route, the router uses MATH-500 (Lightman et al., 2023) and GPQA non-diamond subset for training.

4.2 Experiment Results

Overall Performance. Our main results for accuracy on four benchmarks are shown in Table 1. We find that our TSE method consistently demonstrates superior effectiveness compared to other reasoning approaches. For math tasks in GSM8K and AIME25, TSE achieves the highest accuracy for both Qwen3-4B and Qwen3-8B. In AIME24, TSE remains the highest in the 4B model setting and on par with Self-Route by underperforming only one question in 8B model setting. For the QA task, TSE remains the second-best competitive performance, since the best approach RATT uses retrieval and can access external scientific knowledge to answer complex questions. On average, TSE reaches 59.2% and 51.6% accuracy improvement compared to the non-thinking outputs in terms of Qwen3-4B and Qwen3-8B, respectively. These results indicate that TSE is effective in solving tasks of various difficulties and explores the correct reasoning directions sophisticatedly.

Path Accuracy. We then investigate how TSE improves the accuracy of reasoning paths. Beyond providing correct final outputs, the ideal reasoning process should also provide logically clear and accurate intermediate steps for users to verify the trustworthiness. Thus, we output the reasoning chains generated by the Qwen3-8B model on GSM8K and evaluate their correctness by GPT-4o (Hurst et al., 2024). The principle is that the GSM8K contains relatively simple math word problems with deterministic answers and requires pure arithmetic reasoning, and GPT-4o can effectively validate each step's correctness.

As shown in Figure 5, we observe that in some cases the final answer is correct while the reasoning path contains errors. This phenomenon further highlights the importance of monitoring and correcting the process rather than just the results. The results indicate that TSE not only achieves the highest final answer accuracy but also has the best path accuracy, which demonstrates its superiority in generating reasoning paths that are logically more consistent and verifiable.

Diversity Measurement. We further validate how TSE's exploration contributes to the diversity
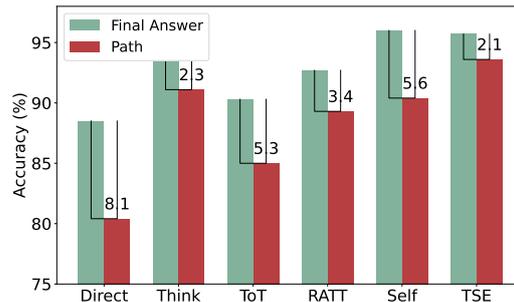

Figure 5: Path accuracy compared with final answer accuracy on GSM8K with Qwen3-8B.

of reasoning. In our experiments, we use Jaccard Similarity (Niwattanakul et al., 2013) to measure the diversity of reasoning trajectories produced by different methods. We segment each reasoning chain into a set of steps by delimiters or step indices. For a given question, we randomly select three reasoning chains that are both correct in answer and path as reference chains, and compute the Jaccard Similarity with all other chains. The Jaccard similarity is defined as

$$J(C_i, C_l) = \frac{|C_i \cap C_l|}{|C_i \cup C_l|},$$

where the intersection $\cap$ represents the shared steps and the union $\cup$ represents the total distinct steps. A higher Jaccard Similarity indicates a larger overlap between generated chains and thus results in lower diversity among generations. For multiple reasoning chains, we compute the Jaccard Similarity for all chain pairs and take the average. We adopt the complement $1 - \bar{J}$ as the overall diversity metric for reasoning paths.

As shown in Figure 6, for both Qwen3-4B and Qwen3-8B models on the GPQA-D dataset, TSE has the highest diversity, surpassing ToT and RATT that contain complex tree structures and naturally have wider exploration fields. Meanwhile, the Self-Route method, although it has competitive output accuracy, its reasoning paths are less diverse and have low path accuracy. Such observations indicate that TSE explores genuinely new reasoning directions rather than simply expanding existing ones. More importantly, TSE translates this exploration into higher-quality reasoning trajectories, thus enhancing both diversity and correctness that other methods might fail to realize.

Token Usage. As TSE requires additional exploration during content generation, we further analyze its computational cost. As shown in Figure 7,

Model	Method	GSM8K		AIME24		AIME25		GPQA-D		Avg.	
		Acc	↑%	Acc	↑%	Acc	↑%	Acc	↑%	Acc	↑%
Qwen3-4B	Direct	86.2	–	20.0	–	20.0	–	36.4	–	40.7	–
	Think	92.0	6.7	60.0	200.0	48.9	144.5	45.0	23.6	61.5	51.1
	ToT	92.7	7.5	63.3	216.5	40.0	100.0	46.0	26.4	60.5	48.8
	RATT	92.7	7.5	56.7	183.5	46.7	133.5	54.5	33.2	61.2	50.4
	Self-Route	93.1	8.0	56.7	183.5	46.7	133.5	43.0	18.1	59.9	47.2
	TSE (Ours)	94.0	9.0	66.7	233.5	50.0	150.0	48.5	33.2	64.8	59.2
Qwen3-8B	Direct	88.5	–	16.7	–	23.3	–	44.4	–	43.2	–
	Think	93.4	8.1	46.7	179.6	46.7	100.4	56.6	27.5	61.4	42.1
	ToT	90.3	2.0	40.0	139.5	33.3	42.9	48.0	8.1	52.9	22.5
	RATT	92.7	4.7	50.0	199.4	36.7	57.5	59.1	20.5	58.2	34.7
	Self-Route	96.0	8.5	63.3	278.4	43.3	85.8	55.1	24.1	64.4	49.1
	TSE (Ours)	95.7	5.5	60.0	239.5	53.3	128.8	58.6	32.0	65.5	51.6

Table 1: Accuracy (%) and relative improvement over the Direct baseline across four benchmarks. Best accuracy is shaded, second-best is underlined.
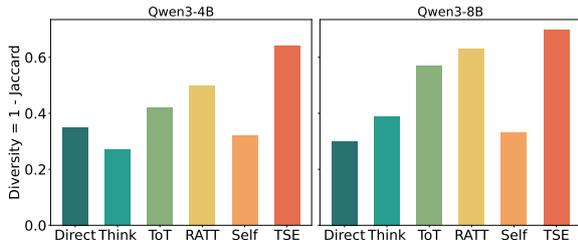

Figure 6: Reasoning diversity across different methods on GPQA-D with Qwen3-4B/8B models.

Figure 7: Token usage against accuracy on AIME24 with Qwen3-8B.

the direct output baseline without generating reasoning tokens has the lowest cost but substantially lower accuracy than all other methods. The Think method, with more than twice the tokens, achieves higher accuracy but still lower than the RATT and Self-Route methods. For ToT, as the number of generated nodes grows exponentially, it has an extremely high token consumption than others, which is not comparable, so we omit it from the figure. The RATT has the highest token usage as it contains a retrieval process, which may not help in arithmetic for math tasks, but can significantly improve QA task performance. Self-Route achieves better accuracy with fewer tokens compared to Think. For TSE, it has competitive accuracy and exhibits the best token-accuracy trade-off, which demonstrates TSE's better effectiveness-efficiency trade-off and its practical value for deployment under limited resources.
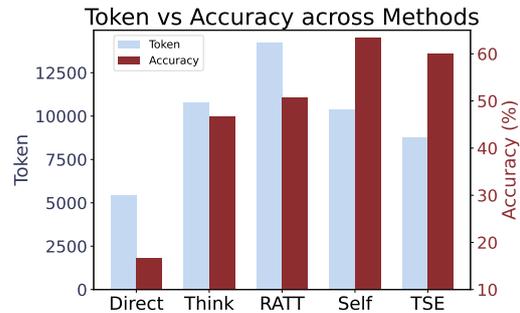
5 Conclusion

In this study, we introduce TSE, a novel approach to enhance the reasoning structures of LLMs. TSE generates new thought branches based on existing thought paths to explore previously overlooked solutions. The generated new reasoning nodes and chains are incorporated into thought structures to explore diverse reasoning directions in terms of a reasoning task. Our experiments across multiple reasoning datasets demonstrate the effectiveness of the TSE. The detailed analysis reveals the utilization of each component in TSE during the integration of diverse thought processes.

Limitations

While TSE demonstrates consistent improvements in reasoning accuracy, path quality, and diversity, several limitations remain. First, the expansion process introduces additional generation cost: although TSE achieves a favorable token–accuracy trade-off compared to baselines, it still requires more tokens than direct decoding. Second, the semantic-based variant depends on prompting quality and the model's intrinsic evaluation ability, which may yield unstable results across different black-box APIs. Third, our experiments are conducted on widely used benchmarks (GSM8K, AIME24/25, GPQA-Diamond); further validation on more diverse domains (e.g., legal or medical reasoning) is needed to assess generality. Finally, TSE focuses on static reasoning structures, whereas real-world tasks often require adaptive depth control or interactive feedback, which remain open directions.

These limitations suggest promising avenues for future work, including developing lighter expansion strategies, designing more robust semantic evaluation protocols, and extending TSE to dynamic or domain-specific reasoning scenarios.

Acknowledgment

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 16, pages 17682–17690.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2023. A survey of chain of thought reasoning: Advances, frontiers and future. *arXiv preprint arXiv:2309.15402*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Sujan Dutta, Deepak Pandita, Tharindu Cyril Weerasooriya, Marcos Zampieri, Christopher M Homan, and Ashiqur R KhudaBukhsh. 2025. Annotator reliability through in-context learning (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 29356–29358.

Xidong Feng, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2023. Alphazero-like tree-search can guide large language model decoding and training. *arXiv preprint arXiv:2309.17179*.

Zitian Gao, Boye Niu, Xuzheng He, Haotian Xu, Hongzhang Liu, Aiwei Liu, Xuming Hu, and Lijie Wen. 2024. Interpretable contrastive monte carlo tree search reasoning. *arXiv preprint arXiv:2410.01707*.

Kye Gomez. 2023. Tree of thoughts. https://github.com/kyegomez/tree-of-thoughts.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.

Yang He, Xiao Ding, Bibo Cai, Yufei Zhang, Kai Xiong, Zhouhao Sun, Bing Qin, and Ting Liu. 2025. Self-route: Automatic mode switching via capability estimation for efficient reasoning. *arXiv preprint arXiv:2505.20664*.

Guiyang Hou, Wenqi Zhang, Yongliang Shen, Linjuan Wu, and Weiming Lu. 2024. Timetom: Temporal space is the key to unlocking the door of large language models' theory-of-mind. *arXiv preprint arXiv:2407.01455*.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Hugging Face. 2024. Llama-3.1-8B-Instruct.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Ziqi Jin and Wei Lu. 2024. Self-harmonized chain of thought. *arXiv preprint arXiv:2409.04057*.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. *arXiv preprint arXiv:2205.11822*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. 2024. Chain of thought empowers transformers to solve inherently serial problems. *arXiv preprint arXiv:2402.12875*.

Xun Liang, Shichao Song, Zifan Zheng, Hanyu Wang, Qingchen Yu, Xunkai Li, Rong-Hua Li, Feiyu Xiong, and Zhiyu Li. 2024. Internal consistency and self-feedback in large language models: A survey. *arXiv preprint arXiv:2407.14507*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Ryan Liu, Jiayi Geng, Addison J. Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L. Griffiths. 2025. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. *Preprint*, arXiv:2410.21333.

Tongxuan Liu, Wenjiang Xu, Weizhe Huang, Xingyu Wang, Jiaxing Wang, Hailong Yang, and Jing Li. 2024. Logic-of-thought: Injecting logic into contexts for full reasoning in large language models. *arXiv preprint arXiv:2409.17539*.

Jieyi Long. 2023. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*.

Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. 2023. Large language models know your contextual search intent: A prompting framework for conversational search. *Preprint*, arXiv:2303.06573.

Math-AI. 2024. Aime 2024. https://huggingface.co/datasets/math-ai/aime24. Accessed: 2025-10-06.

Math-AI. 2025. Aime 2025. https://huggingface.co/datasets/math-ai/aime25. Accessed: 2025-10-06.

Shentong Mo and Miao Xin. 2024. Tree of uncertain thoughts reasoning for large language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12742–12746. IEEE.

Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. 2013. Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 380–384.

OpenAI. 2023. Gpt-4. https://openai.com/index/gpt-4-research/.

OpenAI. 2024. GPT-4o-mini: Advancing Cost-Efficient Intelligence. Accessed: 2024-10-05.

Qianjun Pan, Wenkai Ji, Yuyang Ding, Junsong Li, Shilian Chen, Junyi Wang, Jie Zhou, Qin Chen, Min Zhang, Yulan Wu, and 1 others. 2025. A survey of slow thinking-based reasoning llms using reinforced learning and inference-time scaling law. *arXiv preprint arXiv:2505.02665*.

Deepak Pandita, Tharindu Cyril Weerasooriya, Ankit Parag Shah, Christopher M Homan, and Wei Wei. 2025. Prorefine: Inference-time prompt refinement with textual feedback. *arXiv preprint arXiv:2506.05305*.

David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R So, Maud Texier, and Jeff Dean. 2022. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 55(7):18–28.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Sina J Semnani, Violet Z Yao, Heidi C Zhang, and Monica S Lam. 2023. Wikichat: Stopping the hallucination of large language model chatbots by few-shot grounding on wikipedia. *arXiv preprint arXiv:2305.14292*.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*, 2(5):9.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2025. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *Preprint*, arXiv:2409.12183.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. 2024. Chain of thoughtlessness: An analysis of cot in planning. *arXiv preprint arXiv:2405.04776*.

Weikuan Wang and Ao Feng. 2021. Self-information loss compensation learning for machine-generated text detection. *Mathematical Problems in Engineering*, 2021(1):6669468.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *arXiv preprint arXiv:2402.10200*.

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. 2025. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yu Xia, Rui Wang, Xu Liu, Mingyan Li, Tong Yu, Xiang Chen, Julian McAuley, and Shuai Li. 2024. Beyond chain-of-thought: A survey of chain-of-x paradigms for llms. *arXiv preprint arXiv:2404.15676*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Jinghan Zhang, Xiting Wang, Yiqiao Jin, Changyu Chen, Xinhao Zhang, and Kunpeng Liu. 2024a. Prototypical reward network for data-efficient rlhf. *Preprint*, arXiv:2406.06606.

Jinghan Zhang, Xiting Wang, Fengran Mo, Yeyang Zhou, Wanfu Gao, and Kunpeng Liu. 2025. Entropy-based exploration conduction for multi-step reasoning. *arXiv preprint arXiv:2503.15848*.

Jinghan Zhang, Xiting Wang, Weijieying Ren, Lu Jiang, Dongjie Wang, and Kunpeng Liu. 2024b. Ratt: Athought structure for coherent and correct llmreasoning. *arXiv preprint arXiv:2406.02746*.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. 2024c. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *arXiv preprint arXiv:2406.09136*.

Yifan Zhang, Yang Yuan, and Andrew Chi-Chih Yao. 2024d. On the diagram of thought. *arXiv preprint arXiv:2409.10038*.

Xinyu Zhu, Junjie Wang, Lin Zhang, Yuxiang Zhang, Yongfeng Huang, Ruyi Gan, Jiaxing Zhang, and Yujiu Yang. 2022. Solving math word problems via cooperative reasoning induced language models. *arXiv preprint arXiv:2210.16257*.

Appendix

A Experiments Comparing Black-box and White-box Models

A.1 Experimental Settings

Implementation Details and Baselines. We conduct experiments on three reasoning tasks that require mathematical, hierarchical, and comprehensive reasoning abilities to evaluate the effectiveness of our TSE method. We conduct these experiments utilizing the `GPT-4o-mini` (OpenAI, 2024) and `Llama-3.1-8B-Instruct` (Hugging Face, 2024). Unless otherwise specified, we generate five parallel thought chains with a depth of 5 for each question and use this structure as the basis for new node generation. The temperature for all models is set to the default value of 0.7, with a maximum token limit of 50. All tasks are performed on an NVIDIA 4090 GPU. We apply our method to the simplest multi-chain thought structure and compare the results with several baseline methods, including CoT, Vanilla CoT-SC, ToT, and RATT. This comparison aims to illustrate how our method enhances the thought structure compared to existing approaches.

Task Description. We evaluate TSE and the baseline methods on three task-specific reasoning datasets: (1) *Game of 24* (Yao et al., 2023), a mathematical challenge to use the four basic arithmetic operations to make four given numbers equal 24. The task requires the language models to combine multiple operations to achieve a target outcome, which can evaluate arithmetic reasoning and logical problem-solving capabilities. (2) *Mini Crosswords* (Yao et al., 2023), a game of 5×5 mini crosswords and each input includes the 5 horizontal and 5 vertical clues. To solve this task, the model requires deeper exploration and strategic integration of linguistic clues, allowing us to understand how effectively the evaluated model can expand traditional solution paths and uncover new insights within a complex search space; (3) *Creative Writing* (Yao et al., 2023), a task to construct a coherent passage with four paragraphs, each ending with one of four given sentences.This task compels LLMs to generate imaginative text with logically sound and contextually rich. Detailed information is provided in Appendix.

A.2 Task Performance

The task performance is reported in Table 2 and Figure 8. We can see our TSE consistently outperforms compared baseline methods across different tasks on most of the metrics. The effectiveness of TSE is attributed to expanding thought structures and exploring solution space to contribute to different aspects of reasoning. Then, we analyze each task separately as follows.

Game of 24. As shown in Table 2, the TSE-refined CoT-SC method significantly outperforms other thought structures, including vanilla CoT-SC, with an improvement of 58.56%, ToT by 40.50%, and RATT by 79.04%. Additionally, our enhancement of the basic CoT-SC structure achieves accuracy on GPT-4o-mini that matches the performance of the more complex ToT ($b = 5$) implemented on GPT-4 (Yao et al., 2024). These results highlight the substantial improvements in reasoning capabilities brought about by our method.

Mini Crossword. In this task, we evaluate performance based on two metrics: the proportion of correct letters (out of 25 per game) and the proportion of successfully solved games. As shown in Figure 8b, the CoT-SC with TSE achieves an impressive accuracy rate of 82.4%, significantly outperforming vanilla CoT-SC (by 30.2%), surpassing ToT by 6.9%. However, for the Game metric, RATT performs slightly better than our method, as it can leverage external knowledge through RAG. Specifically, vanilla CoT-SC suffers from limited exploration of thought paths, which reduces its ability to consistently generate accurate answers. These results show that TSE enhances the model's depth and diversity in exploring potential solutions and generates more accurate and coherent outputs.

Creative Writing. As shown in Table 2, the evaluation criteria included *Soundness*, *Innovation*, *Content Coherence of Reasoning*, and *Clarity of Expression*, applying consistently across all evaluations. We rate each metric on a scale from 0 to 10 in 0.5-point increments, and calculate the overall performance score as the average of these four dimensions. A detailed figure can be found in the Appendix.

A.3 Ablation Study

Key Node Selection. To validate the importance of key node selection, we compare (1) the original CoT-SC, (2) the random selection of one node per chain for generation, and (3) our key node selection method. As shown in Table 4, our method significantly improves on *Game of 24* compared to original and random approaches. In *Creative Writing*, particularly for innovation, our method

Task	Game of 24-GPT	Creative Writing-GPT				
	Success Rate (%)	Soundness	Innovation	Coherence	Expression	Overall
CoT	13.30	5.26	5.24	5.03	4.95	5.12
CoT-SC	46.70	5.41	5.17	5.17	5.16	5.23
ToT	52.70	5.90	5.36	5.40	5.43	5.52
RATT	<u>41.30</u>	<u>6.02</u>	<u>6.22</u>	<u>5.69</u>	**5.67**	<u>5.90</u>
TSE	**74.00**	**6.19**	**6.55**	**5.74**	<u>5.64</u>	**6.03**
Task	Game of 24-Llama	Creative Writing-Llama				
	Success Rate (%)	Soundness	Innovation	Coherence	Expression	Overall
CoT	1.33	4.63	3.98	4.25	4.20	4.27
CoT-SC	5.33	4.84	4.35	4.51	4.46	4.54
ToT	8.00	4.86	4.95	4.78	4.65	4.81
RATT	<u>8.67</u>	**5.10**	<u>4.97</u>	<u>4.89</u>	<u>4.97</u>	<u>4.98</u>
TSE	**18.75**	<u>5.03</u>	**5.11**	**5.05**	**5.15**	**5.09**

Table 2: Overall performance on reasoning tasks based on GPT-4o-mini and Llama. **Bold** and <u>underline</u> indicate the best and the second-best results.

also demonstrates notable enhancements, proving its effectiveness in selecting nodes for generation. The superior performance of our key node selection method is primarily due to its ability to identify nodes containing crucial information for problem-solving. The model only utilizes the most influential information and thus avoids redundancy, thus enhancing the model's efficiency and creativity.

Connection Node Selection. We then test the necessity of connection node selection by comparing (1) randomly selecting key nodes to connect, (2) selecting the key nodes based on lower inference layers, and (3) our method. As shown in Table 4, our method is significantly superior to random selection and layer-based selection on overall reasoning performance, particularly in coherence. This is because random selection lacks specificity and cannot guarantee that the chosen nodes will effectively support the content of new nodes. Although layer-based selection considers the structural hierarchy of information, it does not necessarily reflect the actual importance or applicability of the information. Our method analyzes relationships and reasoning paths between nodes, thus maintaining and strengthening the information flow's coherence and depth.

Collaborative Reasoning. In this experiment, we test the importance of the collaboration method in TSE. We compare four collaboration methods: (1) aggregating answers through majority voting, (2) randomly sampling partial thought chains, (3)

using only the outcomes from newly generated chains, and (4) implementing the TSE collaborative method. As shown in Table 4, TSE consistently outperforms other methods in both tasks. Meanwhile, using only new chains also achieves a high success rate in *Game of 24*, demonstrating their potential to offer innovative solutions. However, since these new chains lack integration with original chain information, their performance is still weaker than that of TSE's comprehensive collaboration methods. In *Creative Writing*, using only new chains proved better than majority voting. This suggests that while new chains can offer innovative content, they may lack coherence and depth without sufficient integration with the original chains.

A.4 Strategy Analysis

Impact of Key Node Selection Methods. In this experiment, we compare the impact of two key node selection methods—gradient-based and self-prompting—on Llama's CoT-SC reasoning performance. As shown in Figure 9, both methods achieve comparable results in the *Game of 24*, while self-prompting performs better in *Creative Writing*. This indicates both approaches effectively select critical nodes, but self-prompting additionally helps the model revisit overlooked thought connections and blind spots. Consequently, the model generates more reliable information and produces more creative content. Thus, for tasks demanding higher creativity, self-prompting can encourage the model to move beyond conventional thought

Method	GPT			Llama		
	Success Rate (%)	Overall	Coherence	Success Rate (%)	Overall	Coherence
Key Node Selection						
Original	13.3	5.12	5.03	5.33	4.68	4.85
Random	51.5	5.20	4.97	3.95	4.54	4.51
TSE	74.0	6.03	5.74	18 75	5.09	5.05
Connection Node Selection						
Random	62.8	5.91	5.66	9.52	4.82	4.85
Layer-based	67.3	5.95	5.69	6.12	4.84	4.88
TSE	74.0	6.03	5.74	18.75	5.09	5.05
Collaborative Reasoning						
Majority-Vote	46.7	5.81	5.87	10.0	4.85	4.85
Random Sampling	27.3	5.90	5.89	12.5	4.92	4.90
New Chains Only	64.0	5.95	5.87	18.06	5.11	5.06
TSE	74.0	6.03	5.90	18.75	5.09	5.05

Table 3: Impact of node selection methods.

patterns and foster innovative solutions.

(a) Performance by letter. (b) Performance by game.
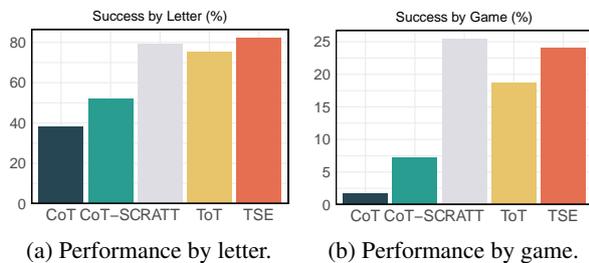
Figure 8: Performance of GPT-4o-mini on Mini Cross-Word.
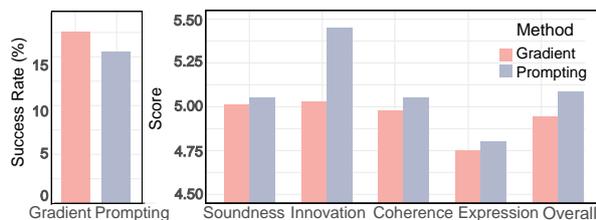
Figure 9: Performance of key node selection methods.

Comparison of Connection Node Selection Methods. We then compare gradient comparison and semantic relevance for selecting connection nodes when generating new reasoning paths on Llama. As shown in Table 4, semantic relevance selection achieves better performance on both *Game of 24* and coherence in *Creative Writing*. The difference possibly arises from two main reasons: (1) Nodes with larger relative gradients do not necessarily have a stronger influence on the content when generating new nodes. (2) Compared to gradient-based selection, the semantic relevance method more effectively captures the actual content

and meaning relationships between nodes, leading to more suitable connection points for new nodes. This demonstrates that considering semantic relevance when selecting connection points provides more effective support for reasoning performance.

Comparison of Collaborative Reasoning Methods. In this experiment, we compare the performance of CWS and LM-as-a-Judge. As shown in Table 4, in task *Game of 24*, which requires dense logic and has well-defined objectives, the LM-as-a-Judge method accurately and quickly identifies the correct answers by directly evaluating each thought chain and selecting the optimal solution. This method relies on the model's ability to assess individual outputs, allowing for rapid selection of the best solution—especially useful in tasks that require precise calculations and quick responses. In task *Creative Writing*, which requires innovation and diversity, CWS evaluates and weights the contributions of each thought chain. The final output represents a blend of multiple perspectives and enhances the richness and depth of the narrative significantly. We further analyze the efficacy and provide a case study to illustrate the effectiveness of TSE generation and collaboration below.

B Experimental Settings

B.1 Baselines

We apply our method to the simplest multi-chain thought structure and compare the results with several baseline methods, including CoT, Vanilla CoT-SC, ToT, and RATT. This comparison aims to illustrate how our method enhances the thought structure compared to existing approaches. For a con-

Method	Game of 24 Success Rate (%)	Creative Writing Overall	Coherence
Connection Node Selection Methods			
Gradient	13.61	4.88	4.86
Relevance	16.32	4.91	4.94
Collaborative Reasoning Methods			
CWS	12.93	4.97	5.01
LM-as-a-Judge	18.05	5.09	5.05

Table 4: Comparison of connection node selection and collaborative reasoning strategy.

sistent evaluation, we standardize the depth to five across all methods. Specifically, we set the number of chains to $k = 5$ and generate $b = 5$ candidates at each step for CoT-SC and ToT, respectively. For RATT, the model generates and integrates five candidate results at each decision point and uses Wikipedia[1] as the external resource.

B.2 Task Description

We evaluate TSE and the baseline methods on three reasoning datasets with specific tasks: (1) *Game of 24*, a mathematical challenge whose objective is to use the four basic arithmetic operations to make four given numbers equal 24. The task requires the language models to combine multiple operations to achieve a target outcome, which can evaluate arithmetic reasoning and logical problem-solving capabilities. (2) *Mini Crosswords*, a game of 5×5 mini crosswords and each input includes the 5 horizontal and 5 vertical clues. The expected output is a completed 55 crossword board containing 25 letters. To solve this task, the model requires deeper exploration and strategic integration of linguistic clues, allowing us to understand how effectively the evaluated model can expand traditional solution paths and uncover new insights within a complex search space; (3) *Creative Writing*, a task to construct a coherent passage with four paragraphs, each ending with one of four given sentences.This task compels LLMs to generate imaginative text withlogically sound, and contextually rich. To ensure a rigorous and comprehensive assessment, we conduct half of the evaluations using GPT-4 (OpenAI, 2023), while the other half are performed by a panel of expert human annotators. In this task, we select 100 open-ended questions from several prompts listed on *Reedsy. com*[2] as our input.

B.3 Thought Chain Efficacy

In this experiment, we conduct a detailed comparison between the success rates of original thought chains and new chains generated by our method in the *Game of 24* task. Experimental results indicate that for each problem, the success rate of having at least one correct answer in the original CoT-SC is approximately 50.7%, while the new chains generated by our method achieve a success rate of 64.0%. Moreover, despite improving accuracy, the overlap of problems successfully solved by both the old and new chains is only 35.3%. This low level of overlap suggests that our method effectively explores new areas that the original thought structures do not address and brings a significant increase in the overall success rate.

B.4 Creative Writing

In Figure 10 we provide detailed results on task *Creative Writing*. TSE method outperforms other methods in three aspects and is only slightly weaker than RATT with RAG.
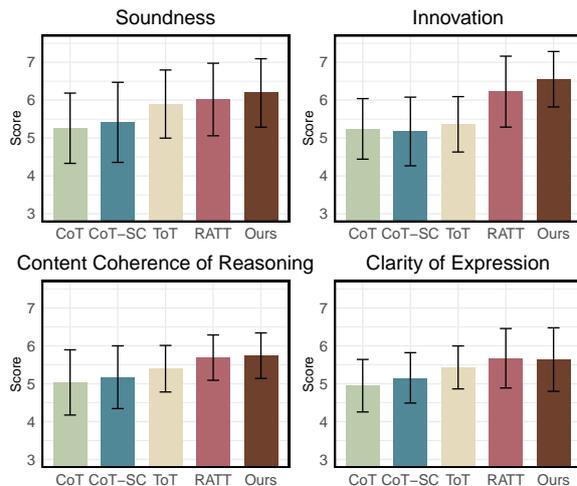

Figure 10: Performance of different methods on GPT-4o-mini for task Creative Writing.

C Detailed Information of Case Study

C.1 Case Analysis

As our approach significantly enhances the reasoning capabilities of thought structures while maintaining minimal computational cost, it effectively addresses gaps within existing structures. Figure 11 and Table 5 provides a specific example of this enhancement by comparing the original thought chains with the new chains generated by our method. In the provided case study, we see

the reasoning prompt *"Explain why it is important for children to learn mathematics"*. The original thought chains (Thought Chain 1 and Thought Chain 2) are linear and somewhat limited in scope, focusing and repeating on foundational aspects like *"Critical Thinking", "Academic Success"* and *"Career Opportunities"*. However, the new chain generated by our method explores additional dimensions, such as *'"Enhancement of Cognitive and Memory Skills"* and *"Preparation for the Digital Age"*. These new perspectives enrich the argument and provide a more comprehensive understanding of the importance of mathematics.

Figure 11: Case study of comparison of original and new chains.

Thought Chain 1	Thought Chain 2	New Chain
Foundation for Critical Thinking: Mathematics fosters critical and logical thinking, enhancing problem-solving and analytical skills necessary for daily life.	**Foundational Skills:** Mathematics equips children with essential daily skills like counting, measuring, and problem-solving, crucial for tasks such as budgeting, shopping, and time management.	**Career Opportunities:** Proficiency in mathematics is crucial for careers in engineering, finance, data analysis, and technology, preparing children for diverse job prospects in a competitive market.
Real-World Applications: Mathematics is crucial for managing daily activities like finances, cooking, and scheduling, empowering children with the confidence to handle them effectively.	**Cognitive Development:** Math enhances logical reasoning, critical thinking, and problem-solving abilities, benefiting other subjects and real-life situations.	**Academic Success:** A strong foundation in mathematics is essential for excelling across subjects like science, technology, engineering, and social studies, enabling access to advanced learning opportunities.
Academic Success: A strong mathematics foundation is essential for excelling in subjects like science, technology, engineering, and social studies, paving the way for advanced educational opportunities.	**Academic Success:** Proficiency in mathematics underpins success in STEM fields, crucial for today's job market, and opens doors to advanced educational opportunities.	**Foundational Skills for Future Success:** Mastering mathematics early equips children with critical thinking and problem-solving skills essential for academic and professional success in fields like science, technology, and engineering, preparing them for a competitive global environment.
Career Opportunities: Proficiency in mathematics is key for careers in engineering, finance, technology, medicine, and data science, equipping children for diverse job prospects in a competitive market.	**Career Opportunities:** Proficiency in mathematics is crucial for careers in engineering, finance, data analysis, and technology, preparing children for a broad range of opportunities in a competitive job market.	**Enhancement of Cognitive and Memory Skills:** Regular math practice boosts cognitive skills like attention to detail, memory retention, and mental arithmetic, essential for efficient learning and problem-solving in academic and everyday life.
Development of Persistence and Resilience: Mathematics teaches children systematic problem-solving, a skill invaluable in academics and beyond.	**Problem-Solving Skills:** Mathematics teaches children systematic problem-solving, a skill invaluable in academics and beyond.	**Preparation for the Digital Age:** Mathematics prepares children for the digital world by teaching them to understand and manipulate numerical data, essential skills for coding and navigating technological advancements.

Table 5: Reasoning Prompt: Explain why it is important for children to learn mathematics.