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

Jinghan Zhang[1], **Fengran Mo**[2], **Tharindu Cyril Weerasooriya**[3], **Kunpeng Liu**[1]*
[1]Clemson University, [2]Université de Montréal, [3]Center for Advanced AI, Accenture,
{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 between 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 defined 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 experiments on math and QA benchmarks, and TSE outperforms baseline methods in accuracy.

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 a direct output to a a chain of intermediate reasoning steps.

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.

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

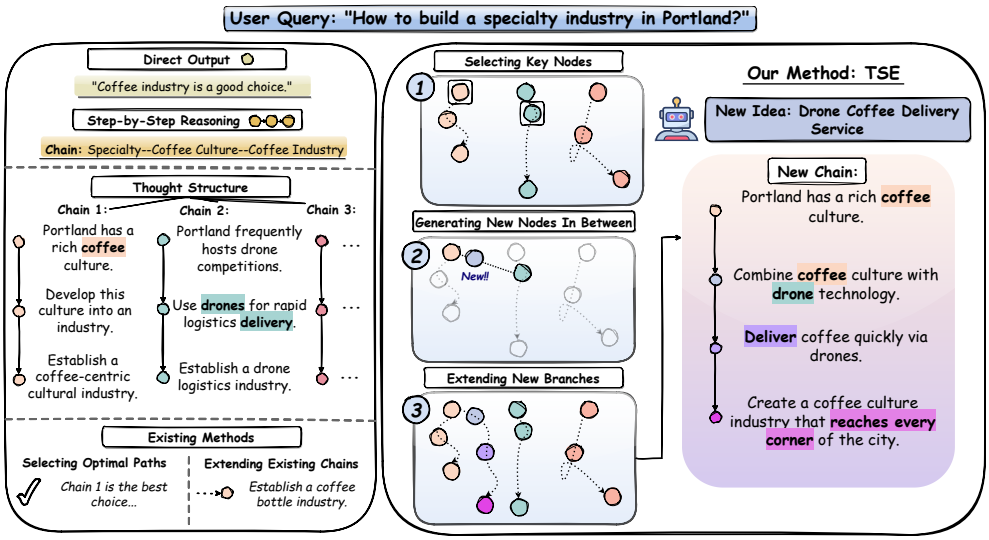

Figure 1: Thought structure optimization through TSE. On the left side, we showcase traditional thought structures and optimization methods, where the LLMs' generation may limited by its thought pattern. On the right side, we show how TSE expands thought structure through a three-step generation of branches.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

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 content 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"* to serve as key information leading toward *"logistics industry"*). We adopt relative gradients as an importance metric 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.

2 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. We implement the TSE through three stages: **(1) Key Node Selection and New Node Generation**, which aims 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) New Node Connection and Chain Expansion**, to connect and expand the reasoning paths, and the new paths explore potential new directions of solutions from the new node; and **(3) Multi-branch Reasoning** to address deficiencies in the model's ability to synthesize and integrate diverse reasoning paths in different directions.

2.1 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) $\mathbf{E}$. Formally, we define them as:

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

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 $\mathcal{P}_S \cup \mathcal{P}_U = \mathcal{P}$.

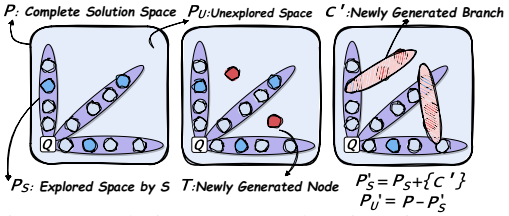

Figure 2: Solution space exploration via our TSE method. By generating new branches of solutions, the explored space of solutions expands.

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: $\max_{\mathcal{S}'} J(\mathcal{S}', \mathcal{Q})$, where J is the reasoning performance metrice and $\mathcal{S}'$ is the expanded thought structure.

2.2 Key Node Selection and New Node 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.

2.2.1 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 $\mathcal{L}$: $T_{ij} \mapsto \mathbf{v}_{ij} \in \mathbb{R}^d$. 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: $\hat{y}_i = f(\mathbf{v}_{iK_i})$, 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 **??**. 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 structure $\mathcal{S}$ as:

$$L_i = -\log P(\hat{y}_i \mid \mathbf{v}_{iK_i}), \quad \mathbf{g}_{ij} = \frac{\partial L_i}{\partial \mathbf{v}_{ij}}, \quad G_{ij} = \|\mathbf{g}_{ij}\|_2. \tag{3}$$

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

$$I_{ij} = \frac{G_{ij}}{\sum_{k=1}^{K_i} G_{ik}}. \tag{4}$$

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. The gradient-based selection use gradient magnitude as a joint indicator of information influence and uncertainty. It guides the model to explore from nodes with the highest potential gain. Also, it provides extra verification at nodes that are most likely to error with limited computing. In this way we make a balance between efficient exploration and stable control.

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:

$$T_{il}^1 = \mathcal{L}(T_i^{\text{key}}, T_l^{\text{key}}), \quad i, l \in [1, N],\ i \neq l. \tag{5}$$

2.3 Connection and Expansion

Then we integrate the new node into the thought structure. Since these key nodes are semantically closest to the new node, we choose between the two key nodes (T_i^{key} or T_l^{key}) to decide which one can 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:

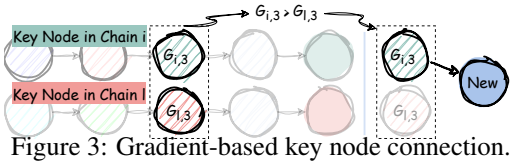

Figure 3: Gradient-based key node connection.

$$T_c = \arg \max_{T_{key} \in T_i^{key}, T_l^{key}} I(T_{key}), \quad C' = \langle T_c, T_{il}^1 \rangle. \tag{6}$$

where C' denotes the new branch initiated from the key node with the higher importance index, as shown in Figure 3. 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 $K = K_l$).

2.4 Multi-branch Reasoning

Finally we reason and produce an output with both original and new branches. Now we have 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:

$$\mathcal{P}'_\mathcal{S} \leftarrow \mathcal{P}_\mathcal{S} \cup \{C'\}, \quad \mathcal{P}'_\mathcal{U} \leftarrow \mathcal{P} - \mathcal{P}'_\mathcal{S}. \tag{7}$$

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

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

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

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:

$$w_{ik}^{\text{key}} = \frac{\exp\left(-L_{ik}^{\text{key}}\right)}{\sum_{T_{im} \in T_i^{\text{key}}} \exp\left(-L_{im}^{\text{key}}\right)}, \tag{8}$$

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:

$$C(\mathcal{Q}) = \sum_{i=1}^{N} \sum_{T_{ik} \in T_i^{\text{key}}} w_{ik}^{\text{key}} \cdot v_{ik}^{\text{key}}. \tag{9}$$

So far, the decision D for task $\mathcal{Q}$ is selected as the candidate with the highest collaborative reasoning score as: $D = \arg\max_{q \in \mathcal{Q}} C(q)$.

3 Experiments

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. For evaluation, we select four widely used math and science benchmarks with different level: **GSM8K** [Cobbe et al., 2021], **AIME24** and **AIME25** [Math-AI, 2024, 2025], and **GPQA-D**iamond [Rein et al., 2024]. 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. 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. Details please refer to Appendix A.

4 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.

Acknowledgment

The author Kunpeng Liu is supported by the National Science Foundation (NSF) via the grant numbers 2550105, 2550106, and 2242812.

References

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

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

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

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

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

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

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

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

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

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

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

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

Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. Large language models know your contextual search intent: A prompting framework for conversational search, 2023. URL `https://arxiv.org/abs/2303.06573`.

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

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

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

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

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

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

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

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

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

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

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

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

Jinghan Zhang, Xiting Wang, Yiqiao Jin, Changyu Chen, Xinhao Zhang, and Kunpeng Liu. Prototypical reward network for data-efficient rlhf, 2024a. URL https://arxiv.org/abs/2406.06606.

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

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

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

Appendix

A Experiment Details

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 **??**.

A.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.

A.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 4, 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

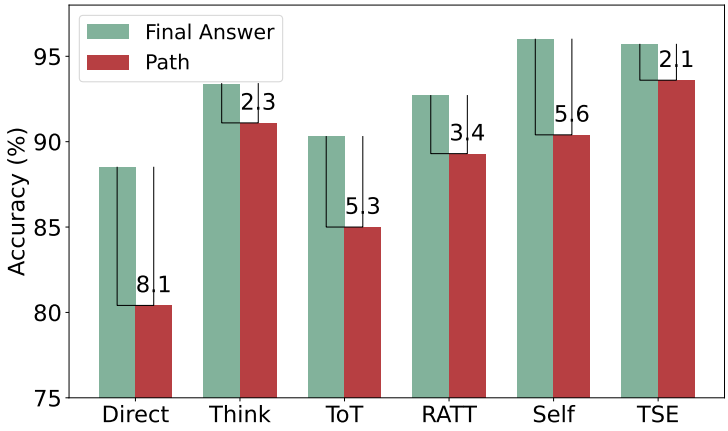

Figure 4: Path accuracy compared with final answer accuracy on GSM8K with Qwen3-8B.

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 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 5, 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 6, 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.

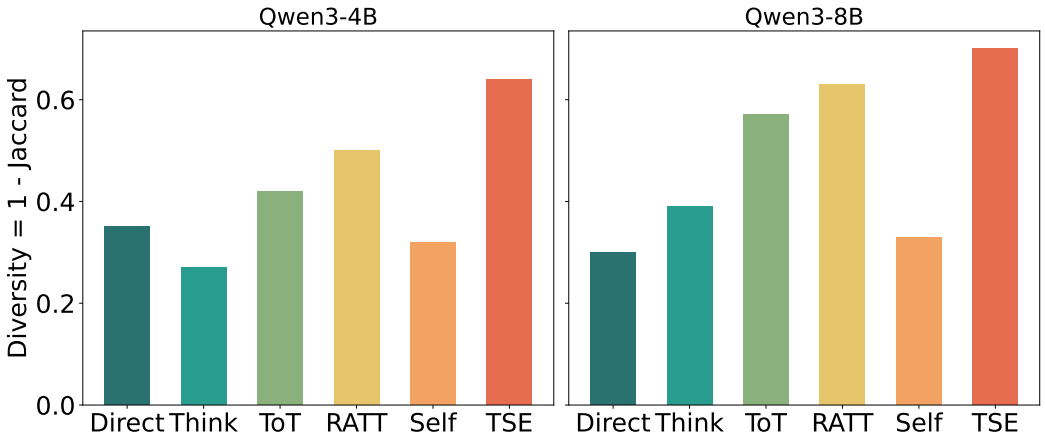

Figure 5: Reasoning diversity across different methods on GPQA-D with Qwen3-4B/8B models.

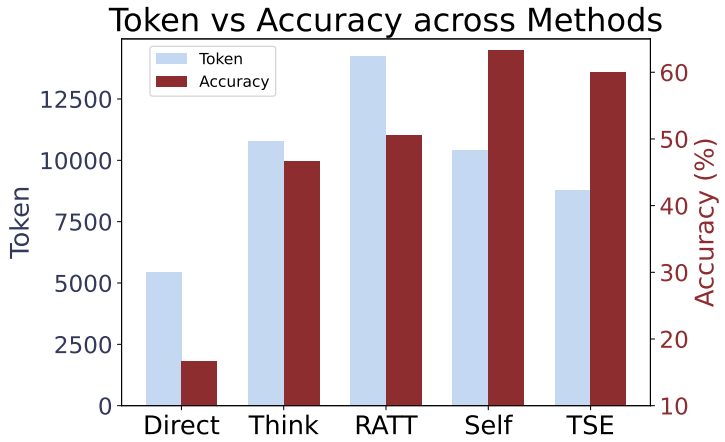

Figure 6: Token usage against accuracy on AIME24 with Qwen3-8B.