ReasonGraph: Visualization of Reasoning Methods and Extended Inference Paths

Zongqian Li
University of Cambridge
zl510@cam.ac.uk

Ehsan Shareghi
Monash University
ehsan.shareghi@monash.edu

Nigel Collier
University of Cambridge
nhc30@cam.ac.uk

Abstract

Large Language Models (LLMs) reasoning processes are challenging to analyze due to their complexity and the lack of organized visualization tools. We present **ReasonGraph**, a web-based platform for visualizing and analyzing LLM reasoning processes. It supports both sequential and tree-based reasoning methods and extended inference outputs while integrating with major LLM providers and over fifty state-of-the-art models. ReasonGraph incorporates an intuitive UI with meta reasoning method selection, configurable visualization parameters, and a modular framework that facilitates efficient extension. Our evaluation shows high parsing reliability, efficient processing, and excellent usability across various downstream applications. By providing a unified visualization framework, ReasonGraph reduces cognitive load in analyzing complex reasoning paths, improves error identification in logical processes, and enables more effective development of LLM-based applications. The platform is open-source, facilitating accessibility and reproducibility in LLM reasoning analysis. [1]

1 Introduction

Reasoning capabilities have become a cornerstone of Large Language Models (LLMs), yet analyzing these complex processes remains a challenge (Huang and Chang, 2023). While LLMs can generate detailed text reasoning output, the lack of process visualization creates barriers to understanding, evaluation, and improvement (Qiao et al., 2023). This limitation carries three key implications: (1) Cognitive Load: Without visual graph, users face increased difficulty in parsing complex reasoning paths, comparing alternative approaches, and identifying the distinctive characteristics of different reasoning methods (Li et al., 2024, 2025); (2) Error Identification: Logical fallacies, circular reasoning, and missing steps remain obscured in lengthy text outputs, impeding effective identification and correction of reasoning flaws; and (3) Downstream Applications: The absence of standardized visualization frameworks restricts the development of logical expression frameworks and productivity tools that could improve and enrich LLM applications. These challenges highlight the essential need for unified visualization solutions that can illustrate diverse reasoning methodologies across the growing ecosystem of LLM providers and models.

To solve these challenges, we present **ReasonGraph**, a web-based platform for visualizing and analyzing LLM reasoning processes. The platform implements six mainstream sequential and tree-based reasoning methods and integrates with major LLM providers including Anthropic, OpenAI, Google, Grok, and Together.AI, supporting over 50 state-of-the-art models. ReasonGraph provides user-friendly UI design with intuitive components, real-time visualization of reasoning methods and extended outputs from reasoning models, meta reasoning method selection, and configurable parameter settings. The platform's modular framework enables easy integration of new reasoning methods, models, and languages while maintaining consistent visualization and analysis capabilities.

Our work makes three main **contributions**:

- **Unified Visualization Platform:** The first web-based platform that enables real-time graphical rendering and analysis of LLM reasoning processes, facilitating comparative analysis across different methods.
- **Modular and Extensible Design:** A flexible framework with modular components for easy reasoning methods and model integrations through standardized APIs.
- **Multi-domain Applications:** An open-source platform that bridges academia, education, and development needs, facilitating accessibility and reproducibility in LLM reasoning analysis.

[1] https://github.com/ZongqianLi/ReasonGraph

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 140–147

The **paper** is organized as follows: Section 2 reviews related work in LLM reasoning methods and visualization approaches. We then detail our UI design principles and layout organization in Section 3, followed by a presentation of our visualization methodology for tree-based and sequential reasoning methods, and extended inference outputs in Section 4. Section 5 elaborates the platform's modular framework and implementation details, while Section 6 demonstrates the platform's versatility through various applications in academia, education, and development. After evaluating the platform from six aspects in Section 7, we conclude in Section 8 with a discussion of future directions.

2 Related Work

LLM **reasoning methods** can be categorized into sequential reasoning and tree-based search approaches. Sequential reasoning, pioneered by Chain-of-Thought prompting (Wei et al., 2022), demonstrates step-by-step problem decomposition and has been improved through multiple variants: Self-consistency (Wang et al., 2023) employs majority voting across multiple reasoning chains, Least-to-Most (Zhou et al., 2023) decomposes complex problems into ordered sub-questions, and Self-refine (Madaan et al., 2023) implements iterative reasoning refinement. Complementarily, tree-based approaches offer broader solution space exploration: Tree-of-Thoughts (Yao et al., 2023) enables state-based branching for parallel path exploration, while Beam Search reasoning (Freitag and Al-Onaizan, 2017) comprehensively evaluates solution paths based on scoring mechanisms, enabling efficient exploration of the reasoning space while maintaining solution diversity.

Visualization approaches for LLM reasoning processes have developed along two main directions: model behavior analysis and reasoning process illustration. In model behavior analysis, tools such as BertViz (Vig, 2019) and Transformers Interpret (Pierse, 2023), while providing detailed visualizations of attention mechanisms and internal states, are limited to low-level model behaviors without showing higher-level reasoning characteristics. For reasoning process illustration, frameworks such as LangGraph (LangChain.AI, 2025b) in LangChain (LangChain.AI, 2025a) offer only basic flow visualization for LLMs without supporting diverse reasoning methodologies, while general-purpose tools such as Graphviz (GraPHP, 2023) and Mermaid (Mermaid.js, 2025), though flexible in graph creation, lack adaptions for LLM reasoning analysis. ReasonGraph introduced in this paper addresses these limitations by providing an open-source platform that supports multiple reasoning methods and various models, offers real-time visualization updates, and enables comprehensive analysis of reasoning processes.

3 UI Design

The **UI** of ReasonGraph shown in Figure 1 employs a two-column layout with a prominent header section for reasoning process visualization. The header section contains a central query input field, a reasoning method dropdown menu for manual method selection (e.g., Chain-of-Thoughts), and three buttons: "Meta Reasoning" for meta reasoning method selection by the model, "Start Reasoning" for using the currently selected method, and "Long Reasoning" for visualizing extended inference outputs from reasoning models. The main UI consists of two panels: the left panel combines Reasoning Settings for API configuration and model selection with Raw Model Output that displays the model's original text response, while the right panel pairs Visualization Settings for diagram parameters with Visualization Results that renders a graph illustration of the reasoning process, complete with zoom, reset, and export.

The UI design includes four fundamental product **design principles**: (1) Functional completeness: incorporating comprehensive model options, reasoning methods, and parameter settings to support diverse analytical needs; (2) Organized layout: maintaining a clear visual organization with the query input prominently positioned in the header, followed by parallel columns for text and graph outputs; (3) Universal usability: offering both manual method selection and model-recommended approaches to accommodate users' decision-making preferences; (4) Visual aesthetics: utilizing an elegant header background and alternating gray-white sections to create an organized appearance while preserving functional clarity (Li and Cole, 2025).

4 Reasoning Visualization

Figure 2 illustrates the contrast between traditional text output and our organized visualization for a **tree-based search method**, beam search. In its visualization, each node denotes a reasoning step with a designated score, and each level maintains a

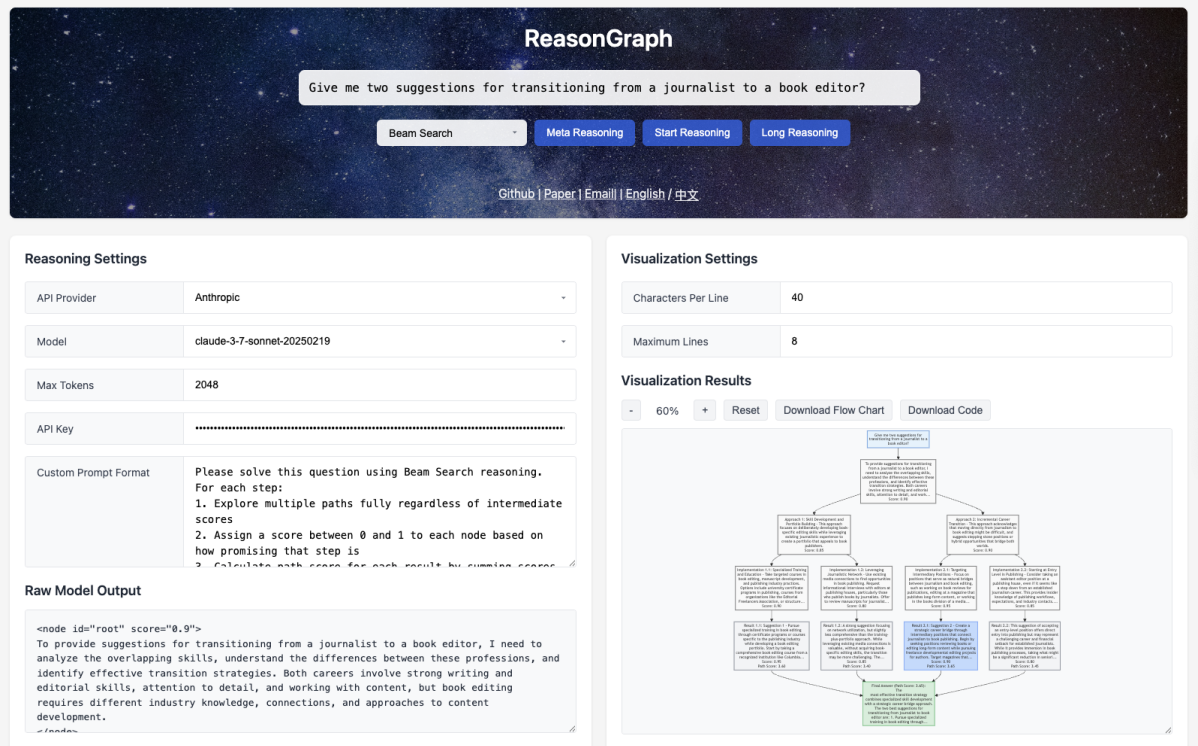


Figure 1: The ReasonGraph **UI** with a query input header and dual-panel layout.

consistent branching width, allowing for comprehensive exploration of solution spaces. The cumulative path scores guide the final solution selection, with the optimal path determined by the highest total score across all levels. While this method shares similarities with Tree-of-Thoughts visualization, the latter differs in its variable branching number and focus on state-space exploration rather than score-based progression. The visualization approach demonstrates clear advantages over raw text output: it provides immediate layout comprehension, enables quick identification of decision points, and facilitates direct comparison of alternative reasoning paths. The graphical illustration also makes the scoring mechanism and path selection process more clear, allowing users to trace the development of reasoning and understand the basis for the final solution.

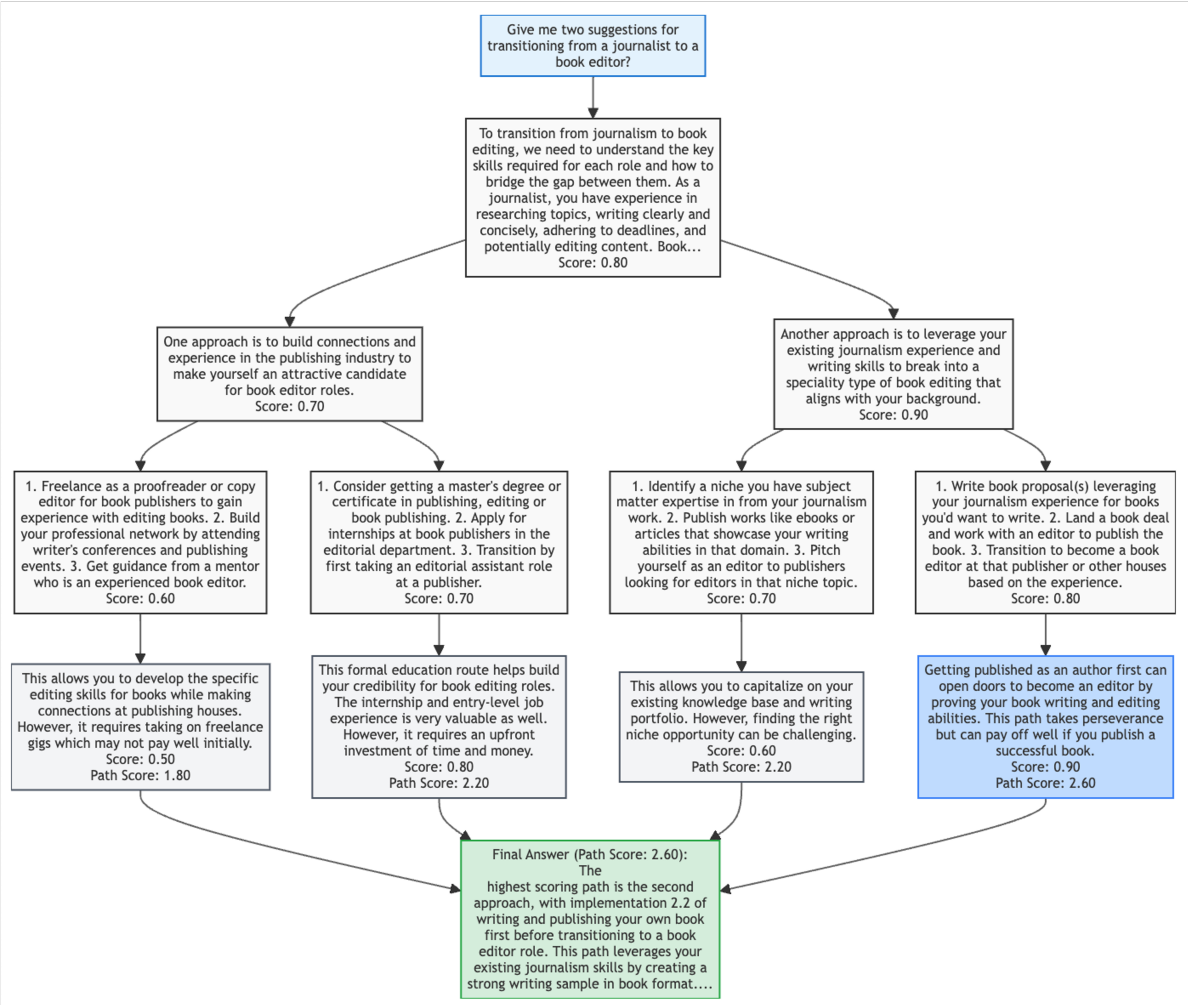
Sequential reasoning processes are visualized through directed graph layouts, as demonstrated in Figure 3. The visualization illuminates the step-by-step progression of different reasoning methods: Chain-of-Thoughts (top-left) displays a linear sequence of deductive steps leading to a final solution; Self-refine (top-center) shows the initial attempt followed by iterative improvements with refinement steps; Least-to-Most (top-right) demonstrates problem decomposition into simpler sub-questions with progressive solution building; and Self-consistency (bottom-left) illustrates multiple parallel reasoning paths converging to a final answer through majority voting. Each method's unique characteristics are exhibited through distinct visual layouts: linear chains for Chain-of-Thoughts, refinement loops for self-refine, leveled decomposition for Least-to-Most, and converging paths for self-consistency reasoning.

Extended inference visualization (Figure 5) integrates linear and tree-based formats to display both thinking processes and results from reasoning models. To address extensive model outputs, each node follows a "Step Name: Content Description" format that summarizes content, enabling rapid comprehension of model thinking without full text review.

5 Framework

ReasonGraph employs a modular framework that facilitates extensible reasoning visualization through separation of components.

The **frontend** tier encapsulates visualization logic and user participation handling. The layer implements an asynchronous event handling module, where user involvements with method selection and parameter configuration trigger corresponding state updates. The visualization module leverages



Here is my approach using beam search to suggest ways to transition from a journalist to a book editor: To transition from journalism to book editing, we need to understand the key skills required for each role and how to bridge the gap between them. As a journalist, you have experience in researching topics, writing clearly and concisely, adhering to deadlines, and potentially editing content. Book editors need many of these same skills but applied to longer-form content like books. One approach is to build connections and experience in the publishing industry to make yourself an attractive candidate for book editor roles (approach1 score: 0.7). 1. Freelance as a proofreader or copy editor for book publishers to gain experience with editing books. 2. Build your professional network by attending writer's conferences and publishing events. 3. Get guidance from a mentor who is an experienced book editor (impl1.1 score: 0.6). This allows you to develop the specific editing skills for books while making connections at publishing houses. However, it requires taking on freelance gigs which may not pay well initially (result1.1 score: 0.5, path score: 1.8). 1. Consider getting a master's degree or certificate in publishing, editing or book publishing. 2. Apply for internships at book publishers ...

Figure 2: Comparison between plain text (bottom) and organized **tree visualization** (top) for the same reasoning process using beam search method. The blue box is the initial question, the darker blue box highlights the selected reasoning path, and the final solution is shown in a green box.

Mermaid.js for dynamic graph rendering, with configurable parameters for node density and layout optimization, enabling real-time updates of reasoning process visualizations.

The **backend** framework is organized around three core modules implemented in Flask: a Configuration Manager for state update, an API Factory for LLM integration, and a Reasoning Methods module for reasoning approach encapsulation. The backend employs a RESTful API layer that ensures component connectivity and robust error handling, making it suitable for both academia and production scenarios.

The framework implements **modularity** at both API and reasoning method levels. The API Factory provides a unified API for multiple LLM providers through the BaseAPI class, while each reasoning method is encapsulated as an independent module with standardized API for parsing and visualization. This design enables dynamic switching between providers and reasoning methods, facilitating platform extension without framework modifications and ensuring adaptability to LLM capabilities.

6 Applications

ReasonGraph serves diverse use cases across academia, education, and development domains. For academic applications, it enables thorough analysis of LLM reasoning processes, facilitating comparative studies of different reasoning meth-

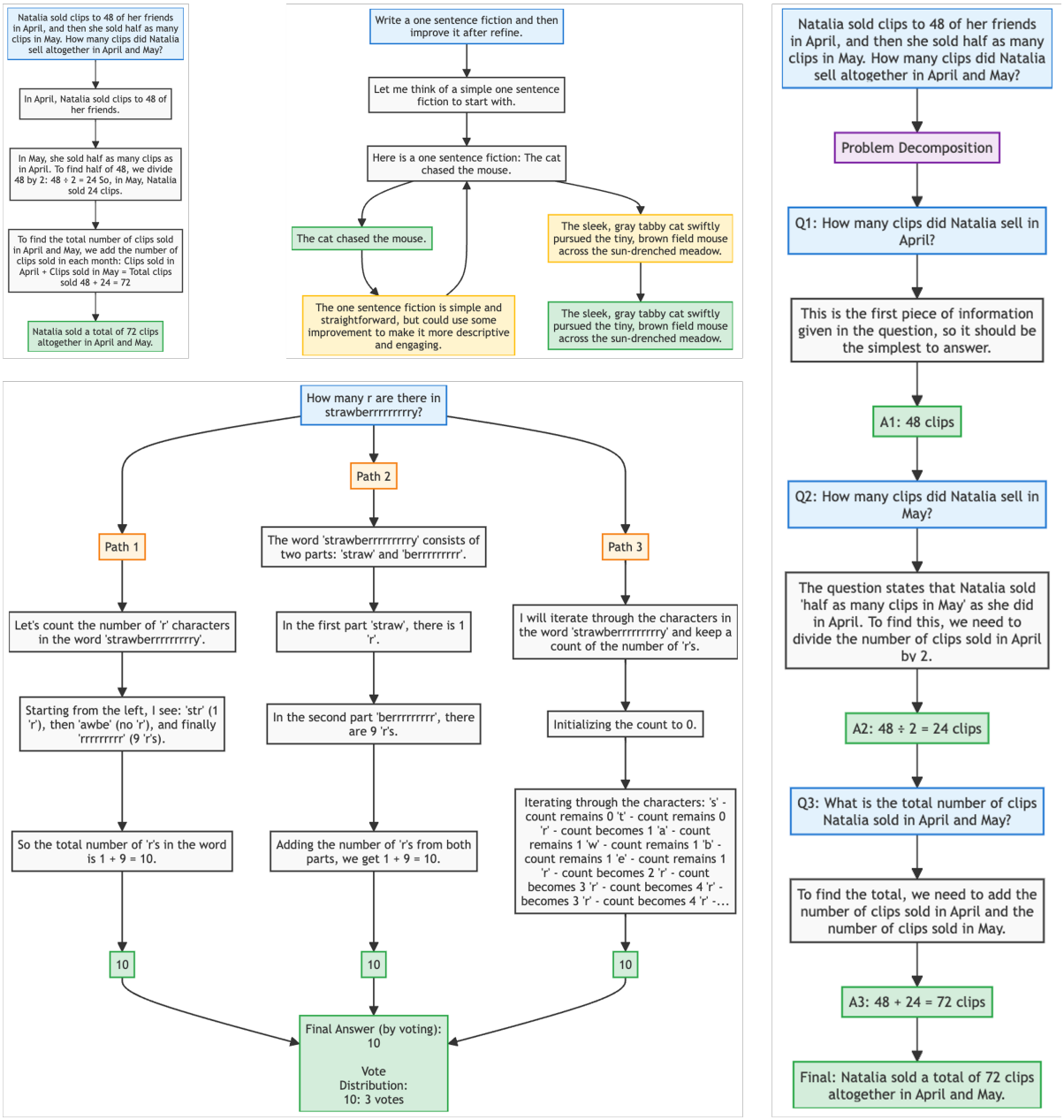


Figure 3: Visualization examples of four **sequential reasoning methods**: Chain-of-Thoughts (top-left), Self-refine (top-center), Least-to-Most (top-right), and Self-consistency (bottom-left). In Self-refine, yellow boxes indicate reflection and improvement steps; in Least-to-Most, light blue boxes are original and decomposed questions while green boxes show intermediate and final answers.

ods and evaluation of model capabilities across various tasks. In educational contexts, the platform serves as an efficient tool for teaching logical reasoning principles and demonstrating LLM decision-making processes, while helping students understand the strengths and limitations of different reasoning approaches. For development purposes, ReasonGraph helps prompt engineering optimization by visualizing how different prompts influence reasoning paths and assists in selecting optimal reasoning methods for specific task types.

7 Evaluation

We evaluated ReasonGraph across four user categories (Junior and Senior participants with Beginner and Experienced backgrounds) to assess whether the platform is beneficial to users with varying abilities. Results in Table 1 demonstrate the robustness of the platform in three key aspects: (1) parsing reliability, with our rule-based XML approach achieving near-perfect accuracy (4.9/5.0) in extracting and visualizing reasoning paths from

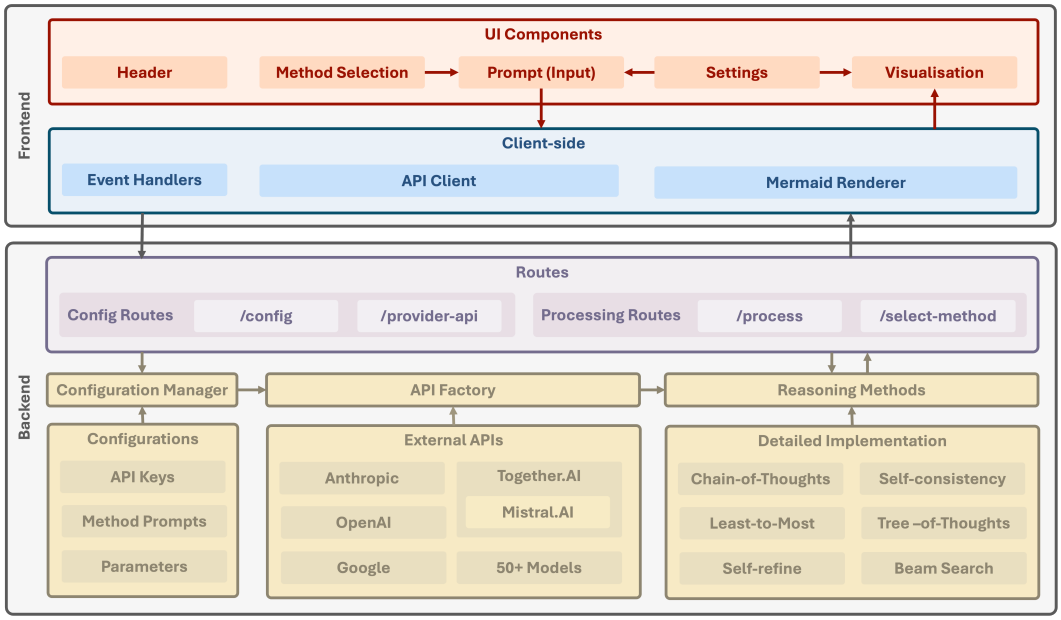


Figure 4: The **framework** of ReasonGraph, consisting of four main layers: UI Components for user involvement, Client-side for frontend processing, RESTful Routes for API bridge, and a modular backend comprising Configuration Manager, API Factory for LLM integration, and Reasoning Methods implementation.

Dimension	Metric	Junior		Senior		Avg.	Plain Text
		Beg.	Exp.	Beg.	Exp.		
Functionality	Model Variety, Reasoning Method Coverage, Output Visualization	4.8	4.8	4.6	4.4	**4.65**	1.65
Accuracy	Text Parsing, Graph Generation, Code Implementation	4.8	5	5	4.8	**4.9**	-
Usability	Installation Complexity, Operation Difficulty, Code Comprehension	4.4	4.8	4.8	5	**4.75**	4.7
Aesthetics	Web UI Design, Generated Visualizations, Code Design	4.6	5	4.8	5	**4.85**	-
Efficiency	Web Performance, Visualization Rendering, Framework Extensibility	5	4.8	5	4.6	**4.85**	-
Potential	Model Integration, Reasoning Method Support, Downstream Applications	5	4.8	4.8	4.4	**4.75**	1.9

Table 1: Evaluation of ReasonGraph across user categories (Junior or Senior levels and Beginner (Beg.) or Experienced (Exp.) background) on six dimensions using a 1-5 scale (higher is better), compared against plain text API. The detailed descriptions of the evaluation metrics are introduced in Table 2.

properly formatted LLM outputs; (2) processing efficiency, where the Mermaid-based visualization generation time (4.85/5.0) is negligible compared to the LLM's reasoning time, maintaining consistent performance across all six reasoning methods; and (3) platform usability, with high scores (4.75/5.0) confirming that most users successfully used the platform without assistance, while the minimal variance in junior and senior groups indicates a smoothing of the learning process. Notably, the platform outperformed plain text, validating ReasonGraph's usefulness in facilitating LLM reasoning analysis. The open-source implementation garnered **over 400** stars on GitHub at the time of paper submission, further indicating community interest and adoption.

8 Conclusions

This paper introduces **ReasonGraph**, a web-based platform that enables visualization and analysis of LLM reasoning processes across six mainstream methods and over 50 models. Through its modular framework and real-time visualization capabilities, the platform achieves high usability across diverse applications in academia, education, and development, improving the understanding and application of LLM reasoning processes.

Future work will pursue four key directions. First, we will leverage the open-source community to integrate additional reasoning methods and expand model API support. Second, we plan to develop the platform based on community feedback and user suggestions, improving platform usability and functionality. Third, we will continue exploring downstream applications such as reasoning evaluation, educational tutorials, and prompting tools. Finally, we aim to implement editable nodes in the visualization flowcharts, enabling direct modification of reasoning processes through the graph workspace.

Limitations

The current development of ReasonGraph has been primarily done by individual efforts, which naturally limits its scope. A broader open-source community effort is needed to improve the platform's performance, identify potential issues in usage, and collaboratively improve the platform's overall function completeness.

Ethics Statement

User participation in the evaluation was approved by MMLL REC, Cambridge.

Availability Statement

The codes related to this paper have been uploaded to https://github.com/ZongqianLi/ReasonGraph. ReasonGraph can be tried online at https://huggingface.co/spaces/ZongqianLi/ReasonGraph.

References

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, Vancouver. Association for Computational Linguistics.

GraPHP. 2023. Graphviz. GitHub repository.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

LangChain.AI. 2025a. Langchain. GitHub repository.

LangChain.AI. 2025b. Langgraph. GitHub repository.

Zongqian Li and Jacqueline M. Cole. 2025. Auto-generating question-answering datasets with domain-specific knowledge for language models in scientific tasks. *Digital Discovery*.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. 2025. Prompt compression for large language models: A survey. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7182–7195, Albuquerque, New Mexico. Association for Computational Linguistics.

Zongqian Li, Yixuan Su, and Nigel Collier. 2024. 500xcompressor: Generalized prompt compression for large language models. *Preprint*, arXiv:2408.03094.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Mermaid.js. 2025. Mermaid. GitHub repository.

Charles Pierse. 2023. Transformers interpret. GitHub repository.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5368–5393, Toronto, Canada. Association for Computational Linguistics.

Jesse Vig. 2019. Visualizing attention in transformer-based language representation models. *Preprint*, arXiv:1904.02679.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

A Appendix

A.1 Extended Inference Visualization

Figure 5 shows an example for visualizing the extended inference process of reasoning models in ReasonGraph.

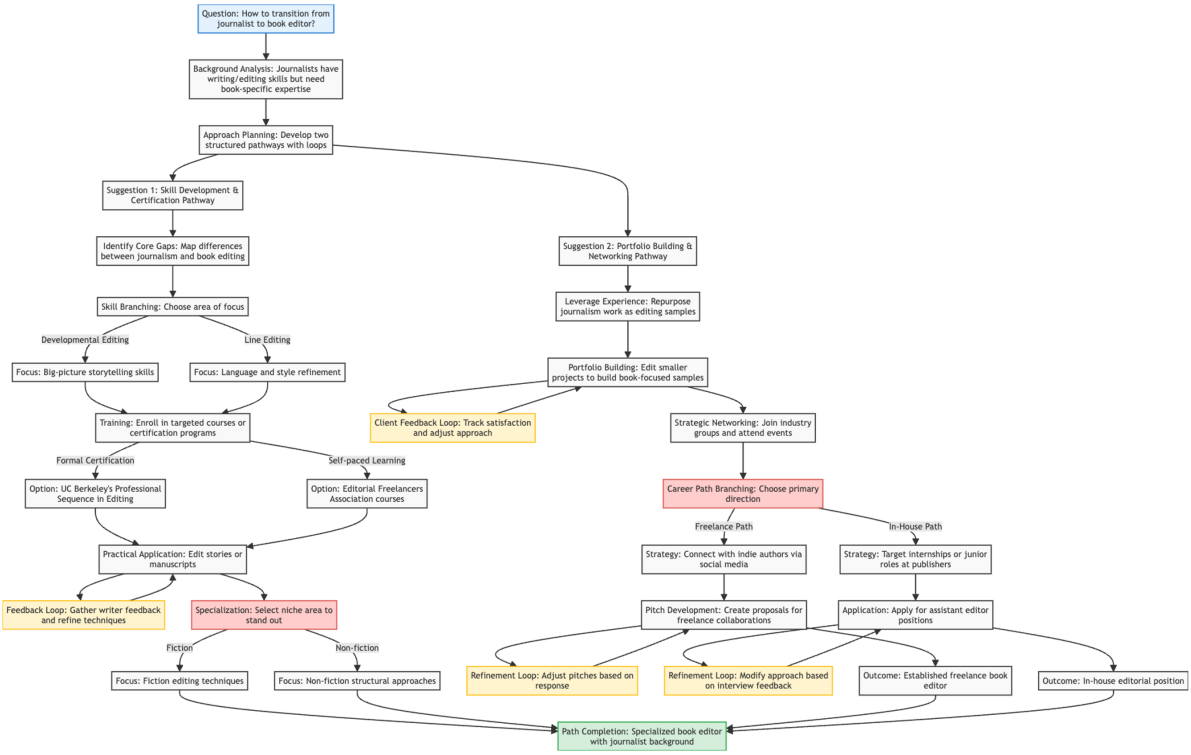


Figure 5: Visualization of extended inference outputs from reasoning models in ReasonGraph, exemplified by a career transition pathway. Blue nodes are questions, green indicate answers or conclusions, yellow show refinement or feedback steps, and red highlight key insights or breakthrough moments. Each node follows the format "Step Name: Content Description".

Dimension	Metric	Description
Functionality	Model Variety	Range of integrated LLM providers and models supported.
	Reasoning Method Coverage	Support for various reasoning methodologies.
	Output Visualization	Visualization capabilities for reasoning processes.
Accuracy	Text Parsing	Reliability of extracting reasoning from LLM outputs.
	Graph Generation	Precision of visualization from text to graphics.
	Code Implementation	Accuracy and completeness of the codes.
Usability	Installation Complexity	Ease of environment creation and platform installation.
	Operation Difficulty	Intuitiveness of platform operations for users.
	Code Comprehension	Readability and documentation quality for developers.
Aesthetics	Web UI Design	Visual appeal and organization of the UI.
	Generated Visualizations	Clarity and readability of generated flowcharts.
	Code Design	Elegance and organization of the codes in the package.
Efficiency	Web Performance	Responsiveness and loading speed of the web platform.
	Visualization Rendering	Speed of flowchart generation processes.
	Framework Extensibility	Ease of extending with new components and methods.
Potential	Model Integration	Adaptability to incorporate new LLM providers and models.
	Reasoning Method Support	Ability to support additional reasoning methods.
	Downstream Applications	Utility across different downstream areas and applications.

Table 2: Detailed description of evaluation metrics for ReasonGraph.

A.2 Evaluation Metrics

Table 2 shows the evaluation metrics used to evaluate ReasonGraph across six dimensions.