DeepSolution: Boosting Complex Engineering Solution Design via Tree-based Exploration and Bi-point Thinking

Zhuoqun Li[1,2], Haiyang Yu[3], Xuanang Chen[1], Hongyu Lin[1*], Yaojie Lu[1], Fei Huang[3], Xianpei Han[1], Yongbin Li[3*], Le Sun[1]

[1]Chinese Information Processing Laboratory, Institute of Software, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Tongyi Lab

{lizhuoqun2021,chenxuanang,hongyu,luyaojie,xianpei,sunle}@iscas.ac.cn
{yifei.yhy,f.huang,shuide.lyb}@alibaba-inc.com

Abstract

Designing solutions for complex engineering challenges is crucial in human production activities. However, previous research in the retrieval-augmented generation (RAG) field has not sufficiently addressed tasks related to the design of complex engineering solutions. To fill this gap, we introduce a new benchmark, SolutionBench, to evaluate a system's ability to generate complete and feasible solutions for engineering problems with multiple complex constraints. To further advance the design of complex engineering solutions, we propose a novel system, SolutionRAG, that leverages the tree-based exploration and bi-point thinking mechanism to generate reliable solutions. Extensive experimental results demonstrate that SolutionRAG achieves state-of-the-art (SOTA) performance on the SolutionBench, highlighting its potential to enhance the automation and reliability of complex engineering solution design in real-world applications. https://github.com/icip-cas/DeepSolution.

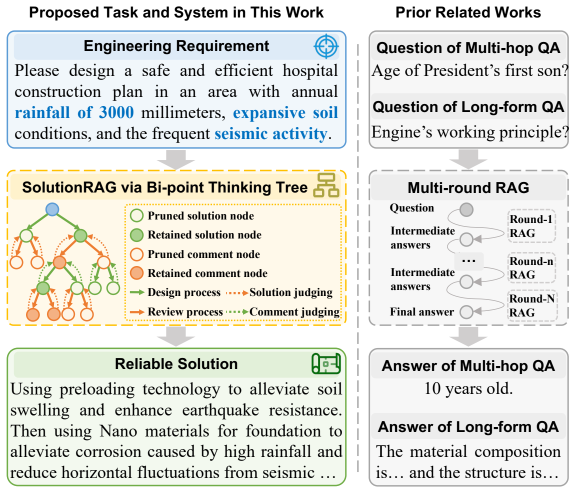

Figure 1: This paper proposes the complex engineering solution design task and a new system that can generate reliable solutions via the bi-point thinking tree.

1 Introduction

Designing solutions for complex engineering requirements is a crucial work in human production activities (Ogot and Kremer, 2004; ElMaraghy et al., 2012). These requirements typically include multiple real-world constraints and expect complete and feasible solutions (e.g., *Design a safe and efficient hospital construction plan in an area with annual rainfall of 3000 millimeters, expansive soil conditions, and frequent seismic activity*). Human experts complete such work by consulting extensive professional knowledge, carefully designing, and strictly deliberating, which require significant time and human resources (Kalogerakis et al., 2010; De Weck et al., 2011). With the continuous development of retrieval-augmented generation (RAG) techniques, the engineering fields anticipate a credible RAG system that can automatically generate reliable solutions for these complex engineering requirements (Yu et al., 2024; Zhou et al., 2024).

Unfortunately, prior works in RAG field do not sufficiently research the complex engineering solution design task. Existing relevant papers mainly focus on Long-form QA or Multi-hop QA (Zhu et al., 2024; Tan et al., 2024), where the questions are integrated or composed of multiple sub-questions and the expected answers are typically assembled knowledge paragraphs or entity fragments. Unlike these tasks, requirements of the complex engineering solution design task involve multiple real-world constraints and demand complete and feasible solutions (Fortus et al., 2005; Jonassen et al., 2006), as shown in Figure 1. Therefore, researching complex engineering solution design based on RAG technology is a valuable gap that needs to be filled.

To fill this gap, we first introduce a new benchmark, **SolutionBench**, to evaluate whether a system can generate complete and feasible solutions for complex engineering requirements with multiple constraints. Firstly, to ensure the data source's

*Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4380–4396

authority, authenticity, and diversity, we collect thousands of engineering reports about solution design from authoritative journals in various engineering domains. Then, to build data that is convenient for testing and evaluation, we refer to the generative information extraction technologies (Lu et al., 2022; Zhang et al., 2025) and employ LLMs to extract useful content from these reports based on a manually formatted template, capturing real-world complex requirements, expert-authored solutions, analytical knowledge used to interpret the requirements, technical knowledge applied in addressing the requirements, and explanations for the expert's solution design process. Finally, we manually verify and revise the extracted content, merge all knowledge within the same domain into a unified knowledge base, and then harvest a complete benchmark for complex engineering solution design that covers eight engineering domains.

To further advance complex engineering solution design, we propose **SolutionRAG**, which can generate reliable solutions through tree-based exploration and bi-point thinking. Firstly, the improvement process from suboptimal solutions to reliable solutions is flexible, rather than with a fixed reasoning pattern. Therefore, SolutionRAG conducts the tree-based exploration, where each branch represents a different improvement direction. Secondly, due to the presence of multiple real-world constraints within the requirements, system-generated solutions cannot guarantee the satisfaction of all constraints. Therefore, SolutionRAG employs the bi-point thinking, which alternates between solution designing and reviewing during the tree growth, gradually improving reliability of generated solutions. Finally, to balance inference efficiency and performance, SolutionRAG implements pruning based on node evaluation, which can keep the inference process along the most promising solutions and the most helpful reviewed comments.

In experiments, we evaluate various types of methods on SolutionBench to assess their ability in complex engineering solution design, including deep thinking models without RAG, standard RAG approaches, multi-round iterative RAG methods, and our SolutionRAG. Experimental results show that LLMs relying solely on internal knowledge cannot effectively solve such tasks. Previous RAG methods also fail to generate satisfactory solutions. In contrast, our proposed SolutionRAG proves to be a more advanced approach. The main contributions of this paper can be summarized as follows:

- We construct SolutionBench, which can evaluate a system's ability for complex engineering solution design from real-world scenarios.
- We propose SolutionRAG, which can boost complex engineering solution design through tree-based exploration and bi-point thinking.
- We conduct extensive experiments, and results show existing methods perform poorly and SolutionRAG is an advanced improvement.

2 SolutionBench

As mentioned above, research on complex engineering solution design tasks has significant value in enhancing the productivity of human society, but previous works in RAG field do not explore this in depth. Therefore, this paper introduces a new benchmark, SolutionBench, which can evaluate a system's ability to design solutions for complex engineering requirements. Specifically, as illustrated in Figure 2, we first collect engineering technical reports about complex solution design from authoritative journals across various engineering fields. Then, based on manually formatted extraction templates, we use powerful LLMs to implement useful content extraction. Finally, after manually checking and removing redundancy, the extracted content is integrated into a complete benchmark. Here is detailed process of constructing SolutionBench:

2.1 Authoritative Data Source

To ensure the credibility of benchmark, we primarily consider two key factors when determining data sources: the authority and authenticity of data, as well as the diversity of engineering domains.

Authority and Authenticity. In order to ensure the benchmark's evaluation results can accurately reflect the system's capabilities under real engineering requirements, it is essential to ensure the data sources come from authoritative experts and real-world scenarios. To this end, we select authoritative journals in engineering fields as data sources, choosing engineering reports that involve complex engineering solution design. The requirements in these reports are derived from real industrial scenarios and provided by industry experts under strict peer review, thus ensuring the authenticity and authority of data sources. The detailed list of used engineering journals is shown in Appendix A.

Domain Diversity. Since the need for complex engineering solution design is urgent in multiple

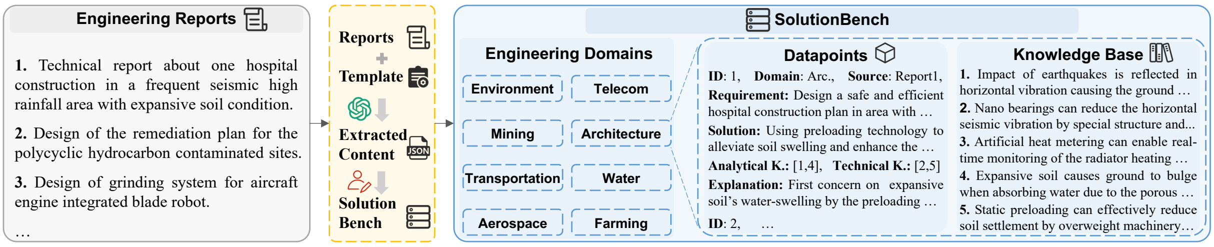

Figure 2: Illustration of the SolutionBench construction method, which includes collecting technology reports from engineering journals to ensure authority and authenticity, extracting useful content based on a manually formatted template and powerful LLMs, and finally harvesting the benchmark after manual verification and merging.

engineering domains, the data sources used to construct benchmark must cover a broad range of domains to ensure comprehensive evaluation. To this end, we select authoritative journals from eight major categories based on the discipline classification mechanism of the search websites: Environment, Mining, Transportation, Aerospace, Telecom, Architecture, Water Resource, and Farming. The coverage of these fields ensures that the data sources include diverse engineering scenarios, providing a broad reference for system evaluation.

2.2 Template-based Extraction via LLM

To transform original engineering technical reports into data for evaluation and scoring, we format a template manually and extract following content from each report via LLMs: requirement, solution, analytical knowledge, technical knowledge, and explanation, based on the generative information extraction (Lu et al., 2022; Zhang et al., 2025).

Template. In order to facilitate the testing and scoring, we format an extraction template including following keys: (1) Requirement, which refers to the complex needs from real engineering scenarios addressed in reports, (2) Solution, which is the complete and reliable solution designed by top industry experts, (3) Analytical Knowledge, which is the professional knowledge used by experts when analyzing the complex requirements during solution design process (e.g., *Impact of earthquakes is mainly reflected in horizontal vibration*), (4) Technical Knowledge, which is the professional knowledge used by experts to address the complex requirements and develop the complete solutions (e.g., *Nano bearings can reduce the horizontal seismic vibration by special structure*), (5) Explanation, which outlines how the experts use analytical knowledge and technical knowledge to analyze the complex requirements and gradually design complete solutions. This explanation can serve as an auxiliary reference during the evaluation process. The complete template used to implement the extraction process is shown in the Appendix B.

Extraction Process. Since the original engineering reports are in PDF format and cannot be directly processed for content extraction, we first use the marker tool[1] to convert the PDF files into plain text. And then we input the plain text along with the manually formatted template into GPT-4o (OpenAI, 2024a), extracting content as described in the template. Finally we transform extracted content into JSON format and save it for further process.

2.3 Manual Data Verification

To further ensure the credibility of the benchmark, we manually check correctness and remove the redundancy for the extracted content.

Correctness Checking. Since the LLM is a probabilistic model and cannot guarantee that every extracted piece of content aligns with our specifications, we manually check each extracted content. On one hand, we examine whether the content matches the information in original engineering reports, on the other hand, we assess whether the content adheres to definitions in the template. For incorrect content, we directly correct it manually.

Redundancy Removing. Since we select many technical reports as data sources for each engineering domain, the analytical knowledge and technical knowledge used to address complex requirements from the same domain may be similar or even identical, resulting in redundancy when constructing a large knowledge base. Therefore, we manually check duplicates for the knowledge in each domain. If duplicates are found, we manually merge the redundant knowledge to one knowledge.

[1] https://github.com/VikParuchuri/marker

Engineering Domain	# Datapoint	# Knowledge
Environment (Env.)	119	554
Mining (Min.)	117	543
Transportation (Tra.)	124	870
Aerospace (Aer.)	115	802
Telecom (Tel.)	116	840
Architecture (Arc.)	118	858
Water Resource (Wat.)	119	802
Farming (Far.)	122	868

Table 1: Statistics of the SolutionBench, which include data and knowledge across eight engineering domains. The number of datapoints in dataset and the number of knowledge in knowledge base are shown above.

2.4 Datapoint and Knowledge Base

After above manual verification, we do content integrate and get 8 high-quality datasets for the 8 domains, correspondingly with 8 knowledge base. The detailed statistics of benchmark is in Table 1.

Datapoint Format. The content of datapoints of every domain is as following formula:

$$\mathcal{D} = \{q_i, s_i, \{k_j^{(a)}\}_{j=1}^{A_i}, \{k_j^{(t)}\}_{j=1}^{T_i}, e_i\}_{i=1}^{N} \quad (1)$$

where $\mathcal{D}$ is the dataset for one domain, N is data number, q_i is one requirement, s_i is the goldn solution, $k_j^{(a)}$ is an analytical knowledge used for q_i and A_i is the total number, $k_j^{(t)}$ is a technical knowledge used for q_i and T_i is the total number.

Knowledge Base. In order to get the referential knowledge base for each engineering domain, we collect all the $k_j^{(a)}$ and $k_j^{(t)}$ within the same domain into a large corpus, as shown in following:

$$\mathcal{K} = \cup[\{k_j^{(a)}\}_{j=1}^{A_i}, \{k_j^{(t)}\}_{j=1}^{T_i}] = \{k_i\}_{i=1}^{M} \quad (2)$$

where $\mathcal{K}$ is the knowledge base for one domain, and M is the number of knowledge in $\mathcal{K}$.

Evaluation Formulating. There are two ways to using SolutionBench for evaluation. The first one is that given a requirement q and expect an reliable solution $\hat{s}$, as shown in following formula:

$$\hat{s} = \mathcal{F}(q) \quad (3)$$

And the second one is RAG setting, which extra provides the relevant knowledge base for retrieval and augmentation, as shown in following formula:

$$\hat{s} = \mathcal{F}(q, \mathcal{K}) \quad (4)$$

Since the completion of above tasks requires various engineering expertise, which is prone to hallucination issues in regular-sized LLMs (Jiang et al., 2023), we mainly focus on the RAG setting in this paper. At the same time, we also test some powerful deep reasoning LLMs in experiments without using RAG, the details are in Section 4.

3 SolutionRAG

To further advance research in complex engineering solution design, we propose SolutionRAG, a system that can generate reliable solutions through tree-based exploration and bi-point thinking. Specially, as illustrated in Figure 3, since the improvement process from a suboptimal solution to a reliable one is flexible and lacks a fixed reasoning pattern, SolutionRAG performs tree-base exploration to find the most effective improvement process for each input requirement. Moreover, due to the multiple real-world constraints within the requirements, the system cannot directly guarantee the generated solutions satisfy all constraints. Therefore, SolutionRAG employs a bi-point thinking approach, alternating between solution design and review, gradually enhancing the solution's completeness and reliability. Finally, to balance inference performance and efficiency, SolutionRAG employs node evaluation to prune the tree, ensuring that the inference process follows the most promising solutions and the most helpful reviewed comments.

3.1 Bi-point Thinking Tree

To explore the optimal process for solution improvement during inference and ensure the output solutions meet all constraints in the requirements, SolutionRAG performs inference based on a bi-point thinking tree, which consists of alternating connected solution nodes and comment nodes.

Solution Node. The content within a solution nodes is the solution designed for the given requirement, which is expected to be a complete and feasible solution meeting all constraints specified in the requirement. The solution nodes at the shallower levels of the tree typically have a lower degree of reliability for the given requirement, while those at deeper levels have a higher degree of reliability. For convenience, we use $s_j^{(i)}$ represents the j-th solution node at the i-th layer of the tree.

Comment Node. The content within a comment node is the comment obtained from reviewing a

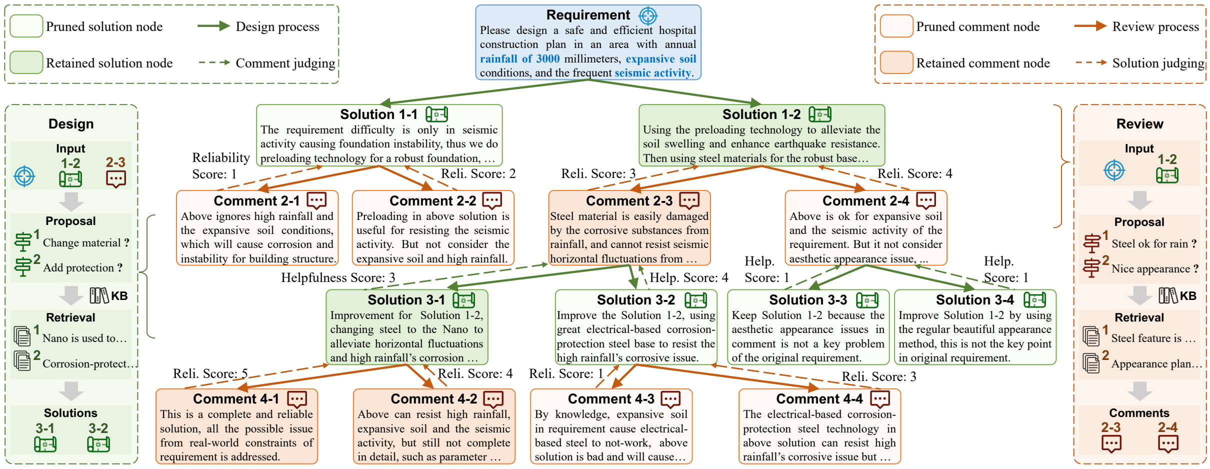

Figure 3: Illustration of SolutionRAG, we set the child number of each node as 2 for easy presentation above. SolutionRAG uses tree-based exploration to find optimal solution improvement process, bi-point thinking to guarantee generated solutions satisfy all constraints, and a pruning mechanism to balance efficiency and performance.

particular solution, which indicates the aspects in which the solution still has deficiencies with respect to the given requirement. For convenience of description, we use $c_j^{(i+1)}$ represents the j-th comment node at the $(i+1)$-th layer of the tree.

Tree Structure. The above-mentioned solution nodes and comment nodes are alternately connected to form a bi-point thinking tree, where the child nodes of a solution node are comment nodes, and the child nodes of a comment node are solution nodes, as shown in the following formula:

$$s_j^{(i)} \rightarrow \{c_h^{(i+1)}\}_{h=1}^{H} \tag{5}$$

$$c_j^{(i+1)} \rightarrow \{s_h^{(i+2)}\}_{h=1}^{H} \tag{6}$$

where H is the number of child node in tree. The content of the root node of the tree is the requirement q, so i is at least one in above formula.

3.2 Solution Improvement via Tree Growth

In this section, we introduce how SolutionRAG specifically achieves continuous improvement of solutions through the growth process of aforementioned bi-point thinking tree, including the node expansion and node evaluation process.

Node Expansion. During the growth process of the bi-point thinking tree, there are two types of node expansion actions, one is based on the requirement or comment node to design new solution nodes, and the other is based on reviewing the solution node to create new comment nodes.

(1) ***Design***. Given the requirement q and a specific comment $c_j^{(i+1)}$ as input (if i is at least one), the design process generate H proposals $\{p_h\}_{h=1}^{H}$ through random sampling using a LLM, representing H different directions for designing:

$$\{p_h\}_{h=1}^{H} = \text{LLM}(q, c_j^{(i+1)}) \tag{7}$$

Then, small-scale relevant knowledge $\mathcal{K}_h$ is retrieved from the knowledge base $\mathcal{K}$ for each p_h:

$$\mathcal{K}_h = \text{Retrieval}(p_h, \mathcal{K}) = \{k_r\}_{r=1}^{R} \tag{8}$$

Finally, q, $c_j^{(i+1)}$, $\mathcal{K}_h$, and the history solution $s_z^{(i)}$ are concatenated as input, allowing the LLM to output a more refined new solution:

$$s_h^{(i+2)} = \text{LLM}(q, s_z^{(i)}, c_j^{(i+1)}, \mathcal{K}_h) \tag{9}$$

Thus, we obtain H new solutions $\{s_h^{(i+2)}\}_{h=1}^{H}$ refined based on the comment $c_j^{(i+1)}$. Note that during the generation of solution nodes in the first layer, there are no previous solutions or comments, so we initialize $s_z^{(i)}$ and $c_j^{(i+1)}$ as empty text.

(2) ***Review***. Similar to the previous process, the review process also consists of three steps: First, proposals $\{p_h\}_{h=1}^{H}$ are generated based on q and $s_j^{(i)}$, representing H distinct review directions. Next, knowledge $\mathcal{K}_h$ is retrieved for each p_h. Finally, comments $\{c_h^{(i+1)}\}_{h=1}^{H}$ are generated for $s_j^{(i)}$ based on q, $s_j^{(i)}$, and $\mathcal{K}_h$. The maximum depth of the bi-point thinking tree, denoted as L, is a hyperparameter. Prompts for this part are Appendix C.

Node Evaluation. As described in above node expansion part, the number of nodes becomes enormous as the tree grows, leading to significant time consumption during inference. To this end, during tree growth, we do prune by each node score from its child nodes, meaning whether $s_j^{(i)}$ is an reliable solution based on $\{c_h^{(i+1)}\}_{h=1}^{H}$ and whether $c_j^{(i+1)}$ is a helpful comment for solution improvement based on $\{s_h^{(i+2)}\}_{h=1}^{H}$. Specifically, for judging $s_j^{(i)}$, we put $s_j^{(i)}$, $c_h^{(i+1)}$, and a suffix u_s together as the LLM input, and get the *reliability score* $\mathcal{J}_h(s_j^{(i)})$ by calculating LLM-predicted average logits of u_s:

$$\mathcal{J}_h(s_j^{(i)}) = \text{Logits}(u_s|s_j^{(i)}, c_h^{(i+1)}) \tag{10}$$

where u_s is "According to the comment, above solution is reliable". And then get final score $\mathcal{J}(s_j^{(i)})$ for $s_j^{(i)}$ by average all $\{\mathcal{J}_h(s_j^{(i)})\}_{h=1}^{H}$. Similarly, for judging $c_j^{(i+1)}$, we use $s_z^{(i)}$, $c_j^{(i+1)}$, $s_h^{(i+2)}$, and u_c as input, and get the *helpfulness score* $\mathcal{J}_h(c_j^{(i+1)})$ by calculating LLM-predicted average logits:

$$\mathcal{J}_h(c_j^{(i+1)}) = \text{Logits}(u_c|s_z^{(i)}, c_j^{(i+1)}, s_h^{(i+2)}) \tag{11}$$

where u_c is "Comparing the new solution and old solution, the comment is helpful", and get $\mathcal{J}(c_j^{(i+1)})$ after same averaging process.

During the tree growth, for each layer we only keep the W highest-scoring nodes, aiming to keep the inference process always focus on the most promising solutions and the most helpful reviewed comments, thus achieving a balance between efficiency and performance. The nodes that are used in final inference process are called *retained nodes*, while those not-used are *pruned nodes*.

4 Experiments

4.1 Experimental Settings

Evaluation Metrics. Since expected system output in SolutionBench are solutions that may have various textual expressions, rule-based metrics are difficult to provide a score that aligns with human habits (Xu et al., 2023; Mayfield et al., 2024). To this end, we follow metrics of previous Long-form QA evaluation methods (Tan et al., 2024; Wang et al., 2024a), using GPT-4o[2] as score evaluator to compute two scores, (1) **Analytical Score**: We integrate the expert-designed solution, analytical knowledge used by experts, and the explanation as reference, and then let evaluator judge whether the system-designed solution, like the expert-designed one, uses the correct analytical knowledge to adequately analysis the complex constraints in requirements. (2) **Technical Score**: Similarly, we integrate the expert-designed solution, technical knowledge used by experts, and the explanation as reference, and then let evaluator judge whether the system-designed solution, like the expert-designed one, uses the correct technical knowledge to tackle the complex constraints in requirements. Both analytical score and technical score are range from 0 to 100. Prompts for this part are in Appendix D.

Selected Baselines. In order to comprehensively evaluate the abilities of various types of systems in solving complex engineering solution design tasks, we extensively select multiple types of methods as baselines in the experiments. Specifically, (1) **Deep Reasoning Models**: This type includes models like o1-2024-12-17 (OpenAI, 2024b), GLM-Zero-Preview (Zhipu, 2024), and QwQ-32B-Preview (Qwen, 2024), which possess strong long-chain reasoning capabilities, but do not utilize external knowledge like RAG. (2) **Single-round RAG Methods**: These methods perform only one round of retrieval and generation, where Naive-RAG (Lewis et al., 2020) does not process the retrieval results, while Rerank-RAG (Li et al., 2023) uses an additional model to re-rank the retrieval results. (3) **Multi-round RAG Methods**: These methods conduct multiple rounds of RAG, iteratively performing tasks such as question rewriting, retrieval, filtering, and generating intermediate answer. We choose 3 accredited methods, which are Self-RAG (Asai et al., 2024), GenGround (Shi et al., 2024), and RQ-RAG (Chan et al., 2024).

Implementation Details. For deep reasoning models in baselines, we directly use official API for experiments. For the single-round and multi-round RAG methods, we follow their official process. For SolutionRAG, we set maximum tree depth L as 5, number of child per node H as 2, and number of retained node W in pruning as 1. To ensure fair comparison, we adopt the following same implementation setting for SolutionRAG and all RAG-based methods in baselines: base model is Qwen2.5-7B-Instruct (Team, 2024), retrieval model is NV-Embed-v2 (Lee et al., 2025), and the number of retrieval results R is setting as 10. For convenience, in all RAG-based experiments, we deploy the base model as API by vLLM[3].

[2] https://openai.com/index/hello-gpt-4o/

[3] https://pypi.org/project/vllm/

Method	Env.		Min.		Tra.		Aer.		Tel.		Arc.		Wat.		Far.	
	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS
Deep Reasoning Models																
o1-2024-12-17 (OpenAI, 2024b)	60.5	48.3	51.9	37.5	57.3	44.7	57.8	47.6	63.5	52.3	61.2	52.0	59.9	50.4	62.9	52.2
GLM-Zero-Preview (Zhipu, 2024)	47.0	30.6	43.2	22.2	45.2	27.0	42.3	25.7	45.1	31.7	47.7	32.4	47.3	30.8	51.4	36.6
QwQ-32B-Preview (Qwen, 2024)	54.3	38.7	48.0	27.9	47.2	29.3	47.4	31.9	52.2	35.9	51.3	35.6	49.2	33.0	53.4	37.0
Single-round RAG Methods																
Naïve-RAG (Lewis et al., 2020)	64.8	62.2	57.2	40.1	62.7	54.9	67.7	65.4	67.4	66.8	66.2	63.3	**66.0**	57.5	65.7	63.0
Rerank-RAG (Li et al., 2023)	62.7	60.7	53.4	38.4	60.0	49.7	65.6	65.2	66.1	63.4	66.4	62.8	64.1	55.4	64.0	59.7
Multi-round RAG Methods																
Self-RAG (Asai et al., 2024)	64.2	63.6	56.1	41.6	62.9	56.5	68.8	69.9	67.6	66.9	66.7	65.9	64.8	58.6	65.1	61.1
GenGround (Shi et al., 2024)	54.8	46.1	53.0	33.3	54.7	37.2	55.7	46.0	58.3	50.7	60.1	50.7	60.4	48.9	59.8	52.7
RQ-RAG (Chan et al., 2024)	53.5	44.4	48.9	28.7	53.8	38.8	55.0	46.1	57.9	44.6	56.3	46.9	54.3	39.8	57.2	45.2
Tree-based Exploration and Bi-point Thinking																
SolutionRAG (Ours)	**66.4**	**67.9**	**59.7**	**50.5**	**64.1**	**58.5**	**69.9**	**72.7**	**68.8**	**69.0**	**67.9**	**68.0**	**66.0**	**60.7**	**66.9**	**65.2**

Table 2: Main experimental results on SolutionBench with eight engineering domains, the AS is the analytical score and TS is the technical score. The table shows that previous methods perform poorly for complex engineering solution design. In contrast, our SolutionRAG is able to output more complete and reliable solutions.

Method	Env.		Min.		Tra.		Aer.		Tel.		Arc.		Wat.		Far.		Overall	
	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS
SolutionRAG	**66.4**	**67.9**	**59.7**	**50.5**	**64.1**	**58.5**	**69.9**	**72.7**	**68.8**	**69.0**	**67.9**	**68.0**	**66.0**	**60.7**	**66.9**	**65.2**	**66.2**	**64.1**
w/o tree structure	63.5	66.5	57.3	46.2	63.1	57.4	60.8	68.4	60.9	63.7	66.2	67.2	65.6	59.9	64.2	63.9	62.7	61.7
w/o bi-point thinking	62.8	64.7	55.6	47.3	61.5	55.7	63.2	68.3	62.6	64.8	67.5	67.3	64.4	59.1	65.2	64.7	62.9	61.5

Table 3: Ablation results for tree-based exploration and bi-point thinking. The table shows that both mechanisms have obviously positive effects for SolutionRAG and exhibit a similar level of importance in the overall.

4.2 Overall Results

Results compared with baselines are shown in the Table 2, there are two main conclusions:

Previous methods fail to effectively address the complex engineering solution design. The table shows that, on one hand, deep reasoning models without RAG perform poorly across all eight domains in SolutionBench. For example, GLM-Zero-Preview achieves an analytical score of only 42.3 in the aerospace domain. On the other hand, RAG-based methods achieve some better performance but still remain at relatively low levels. For instance, Naive-RAG obtains a technical score of only 40.1 in the mining engineering domain, and Self-RAG achieves a technical score of just 63.6 in the environmental engineering domain.

In contrast, SolutionRAG is an effective system for complex engineering solution design tasks. The table shows that SolutionRAG achieves SOTA performance across all of eight domains in the benchmark, demonstrating a significant improvement over baseline methods. For example, in the mining domain, SolutionRAG improves the technical score by 10.4 compared to Naive-RAG and by 8.9 compared to Self-RAG. These experimental results confirm that SolutionRAG can effectively handle complex solution design tasks in various real-world engineering scenarios.

4.3 Ablation Results

Since tree-based exploration and bi-point thinking are two key mechanisms in SolutionRAG, we conduct two ablation experiments, results are shown in Table 3, where "w/o tree structure" is that each node generates only one child, resulting in a single-chain inference pattern, and "w/o bi-point thinking" is that the tree does not include reviewing and all nodes are solutions, leading to a uni-point thinking inference pattern. There are two main conclusions:

Both of the tree-based exploration and bi-point thinking have positive effects. The table shows that removing either mechanism leads to a significant decline in performance, indicating that these two mechanisms are indeed central to solving complex engineering solution design tasks.

Tree-based exploration and bi-point thinking exhibit a similar level of importance. The table shows that after removing these two mechanisms, overall performance decline is quite similar, indicating these two mechanisms hold a comparable level of importance in SolutionRAG.

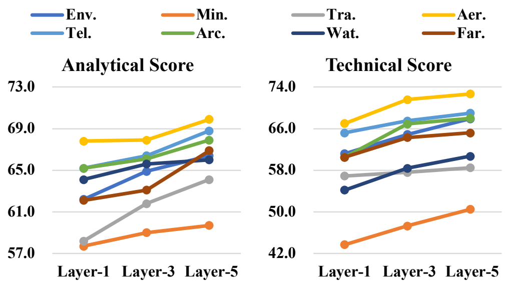

Figure 4: Performance changes during the tree growth. The figure shows that scores become higher as the tree grows, proving SolutionRAG can indeed improve the solution scores as inference being deep.

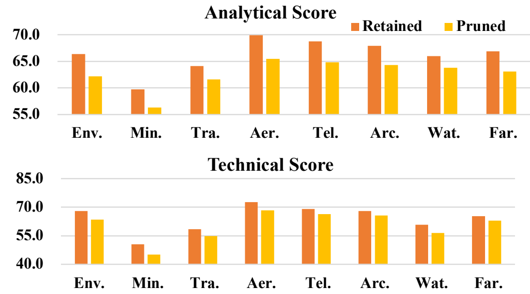

Figure 5: Effectiveness of node evaluation mechanism. The figure shows that scores in retained nodes are higher than in pruned nodes, thus the node evaluation is an effective method for judging and pruning in SolutionRAG.

4.4 Detailed Analysis

In order to further validate the SolutionRAG, we do some detailed analysis, including performance changeing during the tree growth process and effectiveness of the node evaluation in SolutionRAG.

Performance during Tree Growth. To examine whether the solutions actually improve as the tree depth increases in SolutionRAG inference, we score the solutions from the layer-1, 3, and 5 of the tree. The experimental results are shown in Figure 4, performance gradually improves from the shallow layer to the deep layer, which proves that *SolutionRAG can indeed improve the solution as inference process being deep*.

Effectiveness of Node Evaluation. To examine whether node evaluation mechanism for pruning the tree is effective, we compare the scores of solutions from the retained nodes with those from the pruned nodes. The results are shown in Figure 5, where the scores of solutions from the retained nodes are significantly higher than pruned nodes, which proves that *node evaluation is an effective mechanism for judging and pruning*.

5 Related Work

Complex QA Tasks. Recent works in the RAG field mainly focused on knowledge-based question answering tasks that require some level of reasoning. (1) **Multi-hop QA.** The question is a combination of multiple sub-questions, and the expected answer is an entity fragment from relevant knowledge documents (Yang et al., 2018; Ho et al., 2020; Zhu et al., 2024; Wu et al., 2024). (2) **Long-form QA.** The question is an open-ended and comprehensive question, and the expected answer is a text paragraph formed by integrating knowledge fragments from relevant documents (Fan et al., 2019; Stelmakh et al., 2022; Tan et al., 2024; Qi et al., 2024). Compared to above two tasks, questions of complex engineering solution design are with multiple real-world constraints. And the expected answer is a solution needing flexible improvement process, rather than an entity fragment or simply integrated paragraph. Therefore, complex engineering solution design is a novel and challenging task.

Advanced RAG. Prior advanced RAG systems use a multi-round approach to iteratively perform rewriting, retrieval, reranking, and generating intermediate answers (Asai et al., 2024; Shi et al., 2024; Chan et al., 2024; Wang et al., 2024b; Tran et al., 2024; Yu et al., 2024). Compared to these systems, SolutionRAG is with a bi-point thinking tree, which can respond to challenges of complex engineering solution design. Recently some papers construct RAG systems based on MCTS, achieving better performance through deep thinking (Jiang et al., 2024; Li et al., 2025a; Wu et al., 2025). However, these methods lack a mechanism to ensure that all engineering requirements are met, thus failing to guarantee the reliability of solutions.

6 Conclusion

In this paper, we first construct SolutionBench based on engineering reports across various domains, which can examine the ability of systems on complex engineering solution design. Further, we propose SolutionRAG, which explore the optimal solution-improvement process and gradually generates reliable solutions by a bi-point thinking tree. In experiments, previous methods perform poorly in complex engineering solution design task, while SolutionRAG represents a good improvement over existing approaches. This paper offers a promising direction and can inspire the further research.

7 Limitations

Complex engineering solution design is a task requiring deep research based on professional knowledge, which demands the model has strong capabilities in problem analysis, solution reasoning, and critical thinking. In this paper, due to limited GPU computational resources, we construct the system by the existing capabilities of LLMs, without considering special training. Therefore, a possible direction for future work is to use reinforcement learning to train LLMs, in order to develop more powerful complex engineering solution design systems. Additionally, due to the same limitation in GPU computational resources, we do not extensively explore hyperparameters such as the width and depth of the tree in our experiments. This could be a valuable research topic for future work.

Acknowledgments

We sincerely thank the reviewers for their insightful comments and valuable suggestions. This work was supported by Beijing Municipal Science and Technology Project (Nos. Z231100010323002), Beijing Natural Science Foundation (L243006), the Natural Science Foundation of China (No. 62306303, 62476265) and the Basic Research Program of ISCAS (Grant No. ISCAS-ZD-202401).

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. RQ-RAG: Learning to refine queries for retrieval augmented generation. In *First Conference on Language Modeling*.

Olivier L De Weck, Daniel Roos, and Christopher L Magee. 2011. *Engineering systems: Meeting human needs in a complex technological world*. Mit Press.

Waguih ElMaraghy, Hoda ElMaraghy, Tetsuo Tomiyama, and Laszlo Monostori. 2012. Complexity in engineering design and manufacturing. *CIRP annals*, 61(2):793–814.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

David Fortus, Joseph Krajcik, Ralph Charles Dershimer, Ronald W Marx, and Rachel Mamlok-Naaman. 2005. Design-based science and real-world problem-solving. *International Journal of Science Education*, 27(7):855–879.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jinhao Jiang, Jiayi Chen, Junyi Li, Ruiyang Ren, Shijie Wang, Wayne Xin Zhao, Yang Song, and Tao Zhang. 2024. Rag-star: Enhancing deliberative reasoning with retrieval augmented verification and refinement. *Preprint*, arXiv:2412.12881.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

David Jonassen, Johannes Strobel, and Chwee Beng Lee. 2006. Everyday problem solving in engineering: Lessons for engineering educators. *Journal of engineering education*, 95(2):139–151.

Katharina Kalogerakis, Christian Lüthje, and Cornelius Herstatt. 2010. Developing innovations based on analogies: experience from design and engineering consultants. *Journal of Product Innovation Management*, 27(3):418–436.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. Nv-embed: Improved techniques for training llms as generalist embedding models. *Preprint*, arXiv:2405.17428.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025a. Search-o1: Agentic search-enhanced large reasoning models. *Preprint*, arXiv:2501.05366.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *Preprint*, arXiv:2308.03281.

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei

Han, Le Sun, and Yongbin Li. 2025b. StructRAG: Boosting knowledge intensive reasoning of LLMs via inference-time hybrid information structurization. In *The Thirteenth International Conference on Learning Representations*.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. *Preprint*, arXiv:2203.12277.

James Mayfield, Eugene Yang, Dawn Lawrie, Sean MacAvaney, Paul McNamee, Douglas W. Oard, Luca Soldaini, Ian Soboroff, Orion Weller, Efsun Kayi, Kate Sanders, Marc Mason, and Noah Hibbler. 2024. On the evaluation of machine-generated reports. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1904–1915, New York, NY, USA. Association for Computing Machinery.

Madara Ogot and Gul Kremer. 2004. *Engineering design: a practical guide*. Trafford Publishing.

OpenAI. 2024a. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2024b. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Zehan Qi, Rongwu Xu, Zhijiang Guo, Cunxiang Wang, Hao Zhang, and Wei Xu. 2024. $long^2rag$: Evaluating long-context & long-form retrieval-augmented generation with key point recall. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4852–4872, Miami, Florida, USA. Association for Computational Linguistics.

Team Qwen. 2024. Qwq: Reflect deeply on the boundaries of the unknown.

Zhengliang Shi, Shuo Zhang, Weiwei Sun, Shen Gao, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. 2024. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7339–7353, Bangkok, Thailand. Association for Computational Linguistics.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haochen Tan, Zhijiang Guo, Zhan Shi, Lu Xu, Zhili Liu, Yunlong Feng, Xiaoguang Li, Yasheng Wang, Lifeng Shang, Qun Liu, and Linqi Song. 2024. ProxyQA: An alternative framework for evaluating long-form text generation with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6806–6827, Bangkok, Thailand. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hieu Tran, Zonghai Yao, Junda Wang, Yifan Zhang, Zhichao Yang, and Hong Yu. 2024. Rare: Retrieval-augmented reasoning enhancement for large language models. *Preprint*, arXiv:2412.02830.

Minzheng Wang, Longze Chen, Cheng Fu, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. 2024a. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa. *Preprint*, arXiv:2406.17419.

Ruobing Wang, Daren Zha, Shi Yu, Qingfei Zhao, Yuxuan Chen, Yixuan Wang, Shuo Wang, Yukun Yan, Zhenghao Liu, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024b. Retriever-and-memory: Towards adaptive note-enhanced retrieval-augmented generation. *Preprint*, arXiv:2410.08821.

Feijie Wu, Zitao Li, Fei Wei, Yaliang Li, Bolin Ding, and Jing Gao. 2025. Talk to right specialists: Routing and planning in multi-agent system for question answering. *Preprint*, arXiv:2501.07813.

Jian Wu, Linyi Yang, Zhen Wang, Manabu Okumura, and Yue Zhang. 2024. Cofca: A step-wise counterfactual multi-hop qa benchmark. *Preprint*, arXiv:2402.11924.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. 2023. A critical evaluation of evaluations for long-form question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3225–3245, Toronto, Canada. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Preprint*, arXiv:1809.09600.

Tian Yu, Shaolei Zhang, and Yang Feng. 2024. Auto-rag: Autonomous retrieval-augmented generation for large language models. *arXiv preprint arXiv:2411.19443*.

Zikang Zhang, Wangjie You, Tianci Wu, Xinrui Wang, Juntao Li, and Min Zhang. 2025. A survey of generative information extraction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4840–4870, Abu Dhabi, UAE. Association for Computational Linguistics.

Zhipu. 2024. Glm-zero.

Yujia Zhou, Zheng Liu, and Zhicheng Dou. 2024. Boosting the potential of large language models with an intelligent information assistant. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. Fanoutqa: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37.

A List of Engineering Journals

In order to ensure that the data sources used to construct the benchmark are authentic, authoritative, and diverse, we select engineering reports on solution design from authoritative journals in multiple engineering fields as our data sources. If the report is in Chinese, we extract the useful content, then use GPT-4o to translate the content into English and manually verify its accuracy. We list the used engineering journals, including the journal name and ISSN meaning the international standard serial number. The detailed list of used engineering journals is shown in Table 4, 9 and 10.

B Template for Extraction

In order to obtain the necessary content for evaluating and judging systems from engineering reports, we manually format a template. When extracting, we combine the report with this template, input it into GPT-4o, and then organize the output into JSON format and save it. The extracted content includes: real-world complex requirements, expert-authored solutions, analytical knowledge used to interpret the requirements, technical knowledge applied in addressing the requirements, and explanations for the expert's solution design process. The complete template is shown in Figure 6.

C Prompt for Node Expansion

In the growth of the tree, there are two expansion processes: design and review. The review process is divided into two stages: generating proposals based on parent node information and generating comments based on retrieved documents. The design process is also divided into two stages: generating proposals based on parent node information and generating solutions based on retrieved documents. Moreover, the design process based on the root node and the design process based on the comment node use different prompts due to the differences in input information. All the prompts mentioned in this section are shown in Figure 7.

Environment	
Journal Name	**ISSN**
Journal of Environmental Engineering Technology	1674-991X
Environmental Sanitation Engineering	1005-8206
The Administration and Technique of Environmental Monitoring	1006-2009
Environment and Development	2095-672X
Environmental Protection and Technology	1674-0254
Green Environmental Protection Building Materials	1673-6680
Journal of Henan University of Urban Construction	1674-7046
Urban Management and Science & Technology	1008-2271
Science and Technology Square	1671-4792
Construction Materials & Decoration	1673-0038
Intelligent City	2096-1936
Instrument Standardization & Metrology	1672-5611
Northwest Hydropower	1006-2610
Technology & Economics in Petrochemicals	1674-1099
Water Purification Technology	1009-0177
Construction Science and Technology	1671-3915
Urban Geology	2097-3764
Engineering and Construction	1673-5781
Engineering and Technological Research	2096-2789
Scientific and Technological Innovation	2096-4390
Engineering & Test	1674-3407
Inner Mongolia Water Resources	1009-0088
China Cement	1671-8321
Guangdong Chemical Industry	1007-1865
Jiangxi Building Materials	1006-2890
Tianjin Science & Technology	1006-8945
Journal of Zhejiang University of Water Resources and Electric Power	2095-7092
China Municipal Engineering	1004-4655
China Storage & Transport	1005-0434
Mining	
Journal Name	**ISSN**
Coal Engineering	1671-0959
Mining Engineering	1671-8550
Mechanical Management and Development	1003-773X
Coal and Chemical Industry	2095-5979
Colliery Mechanical & Electrical Technology	1001-0874
Modern Mining	1674-6082
China Mine Engineering	1672-609X
Shandong Coal Science and Technology	1005-2801
Jiangxi Coal Science & Technology	1006-2572
Metal Mine	1001-1250
Modern Chemical Research	1672-8114
Petroleum Geology and Engineering	1673-8217
Coal Mine Modernization	1009-0797
Shaanxi Coal	1671-749X
Drilling Engineering	2096-9686
Mineral Resources and Geology	1001-5663
Mine Surveying	1001-358X
Coal	1005-2798
Mining Equipment	2095-1418
Inner Mongolia Coal Economy	1008-0155
Inner Mongolia Petrochemical Industry	1006-7981
Energy and Energy Conservation	2095-0802
China Plant Engineering	1671-0711
Engineering and Construction	1673-5781
Scientific and Technological Innovation	2096-4390
Engineering & Test	1674-3407
Energy Technology and Management	1672-9943
Coal Technology	1008-8725

Table 4: List of the engineering journals used for construction the benchmark. The information for environment domain and mining domain is shown above, and information for other domains is in Table 9 and 10.

D Prompt for Scores Calculation

In order to evaluate the solutions provided by the system, we follow the methods from previous Long-form QA evaluation (Tan et al., 2024; Wang et al., 2024a; Li et al., 2025b), and use a LLM-based scoring method. Specifically, for a given solution generated by the system, we calculate two scores: (1) Analytical score, which uses the golden solution, explanation, and corresponding analytical knowledge as references, allowing GPT-4o to assess whether the system's solution sufficiently consider the challenges posed by the complex constraints in the requirements. (2) Technical score, which uses the golden solution, explanation, and corresponding technical knowledge as references, allowing GPT-4o to evaluate whether the system's solution correctly apply the appropriate technologies to address the complex constraints in the requirements. Both analytical score and technical score are range from 0 to 100. The used prompts for score calculation are shown in Figure 8.

E Experiment Results on More LLMs

In order to further demonstrate the effectiveness of SolutionRAG, we conduct experiments using various LLMs as base models. The experimental results show that SolutionRAG still achieves the best performance, proving its strong generalization capability across different base models. Detailed experimental results are shown in Table 5, 6, 7, 8.

```
The above is a solution for a complex task. You need to extract the following content from the input document, with the following requirements:
1. Extraction must strictly follow the template.
2. Extraction must strictly be based on the document content; no additional information may be added.
3. The extracted content must be detailed, including specific values and professional terms.

{
   "title": "", # The title of the document
   "requirement": "Under these conditions of..., complete the task of...", # A detailed and comprehensive description of the task in the document, covering all the conditions mentioned in the above analysis
   "solution": "This paper proposes the solution of..., specifically, first, considering the conditions of..., and the challenges faced, using... technologies, through... achieved...; secondly,...", # Based on the explanation, a detailed and comprehensive introduction to the solution in the document
   ”analytical knowledge": [ # Extract the analysis of the task from the document, with specific values and professional terms
      {
         "idx": "analysis0",
         "condition": "Condition is...", # The description of task conditions in the document
         "challenge": "Under this condition, conducting... will face challenges such as..., which may lead to... consequences" # Based on the document's analysis, the challenges and potential negative consequences encountered in the task due to this condition
      },
      ...
   ],
   "technical knowledge": [ # Extract the solutions for the task from the document, with specific values and professional terms
      {
         "idx": "technology0",
         "name": "...technology", # A key technology in the solution in the document
         "detail": "This technology applies to..., and can achieve..." # The scope and effects of the technology mentioned in the document, with specific values and professional terms
      },
      ...
   ],
   "explanation": [ # Find the reasoning from the task to the solution in the document, corresponding to the analysis. For each analysis, an explanation must be provided; one analysis may correspond to one or more technologies
      {
         "idx": "explanation0",
         "content": "In response to analysis0, considering the conditions of... and the challenges that may arise, the use of technology0... achieves...", # Find the explanation in the document that describes how the mentioned technologies solve the challenges from the analysis
      },
      ...
   ],
}

Example:
{
   "title": "Analysis of Difficulties in High-Temperature and Ultra-Deep Geothermal Well Drilling and Construction Plan Design in Datong Area",
   "requirement": "Under the conditions of complex strata, high-temperature and high-pressure environments, and ultra-deep high geostress, complete the task of drilling a geothermal exploration well to a depth of 4000 meters, requiring full-core sampling, a core recovery rate of ≥70%, and meeting well inclination and logging quality requirements.",
   "solution": "This paper proposes a construction solution for complex strata, high-temperature high-pressure, and ultra-deep high geostress conditions. Specifically, first, considering the complexity of the strata, which may lead to hole collapse and stuck pipe, the use of high-temperature drilling fluid technology and wireline coring technology, through optimizing tools and mud performance, achieved efficient drilling and core recovery; secondly, to address the high-temperature and high-pressure environment that may lead to blowout risks, high-temperature drilling fluid and well control technology were employed, ensuring construction safety through sealing agent formulations and blowout preventer configuration; finally, to address the wall instability issues caused by ultra-deep high geostress, the automatic drilling system and cementing technology were used, ensuring wall stability and progress through stable drilling pressure and high-performance cementing."
   "analytical knowledge": [
      {
         "idx": "analysis0",
         "condition": "The condition is complex strata, including Quaternary and Neogene loose strata, and deep rocks of the Precambrian Jining group, possibly with fractured rock layers.",
         "challenge": "Under this condition, drilling will face challenges such as hole collapse, excessive wear on drilling tools, low rock drillability, and stuck pipe, which may lead to low construction efficiency, tool damage, or hole instability."
      },
      {
         "idx": "analysis1",
         "condition": "The condition is a high-temperature and high-pressure environment, with temperatures exceeding 150°C at depths above 1500 meters, and even higher temperatures at greater depths.",
         "challenge": "Under this condition, drilling will face challenges such as high temperature resistance requirements for drilling fluids and tools, and a high risk of blowouts, which may lead to blowout accidents and drilling fluid failure."
      },
      {
         "idx": "analysis2",
         "condition": "The condition is ultra-deep hole with high stress, with depths exceeding 3000 meters, potentially causing high geostress issues.",
         "challenge": "Under this condition, drilling will face challenges such as wall instability, collapse, necking, and losses, which may lead to deformation or interruption of the drilling process."
      }
   ],
   "technical knowledge": [
      {
         "idx": "technology0",
         "name": "High-Temperature Drilling Fluid Technology",
         "detail": "This technology applies to high-temperature strata, using high-temperature-resistant water-based drilling fluids with a resistance of 240°C and 220°C sealing agents, which can operate stably at 236°C, ensuring hole wall stability and drilling fluid performance."
      },
      {
         "idx": "technology1",
         "name": "Wireline Coring Drilling Technology",
         "detail": "This technology is suitable for deep rock drilling, achieving stable and efficient core sampling in hard rocks by reducing tripping time and improving drilling efficiency."
      },
      {
         "idx": "technology2",
         "name": "Automatic Drilling System",
         "detail": "Suitable for deep hole construction, it modifies traditional drilling rigs to achieve stepless speed regulation and stable control of drilling pressure, meeting the requirement of 1.2 meters/hour drilling speed for deep wells."
      },
      {
         "idx": "technology3",
         "name": "Well Control and Cementing Technology",
         "detail": "Uses 244.5 mm casing and a high-temperature-resistant cement system, with a blowout preventer that has a sealing pressure of 14 MPa, effectively preventing blowout accidents in high-temperature and high-pressure wells."
      }
   ],
   "explanation": [
      {
         "idx": "explanation0",
         "content": "In response to analysis0, considering that complex strata may lead to hole collapse and stuck pipe, the use of technology1 and technology0, through efficient tool design and optimized mud performance, achieved stable drilling and high core recovery rates."
      },
      {
         "idx": "explanation1",
         "content": "In response to analysis1, considering that high-temperature and high-pressure conditions may trigger blowouts and drilling fluid failure, the use of technology0 and technology3, through high-temperature-resistant formulations and blowout preventer configurations, ensured construction safety and continuity."
      },
      {
         "idx": "explanation2",
         "content": "In response to analysis2, considering that ultra-deep high geostress may lead to wall collapse and necking, the use of technology2 and technology3, through stable drilling pressure and high-performance cementing, resolved the complex issues caused by geostress."
      }
   ],
}
```

Figure 6: Template used to extract useful content from original engineering reports, aiming to capture real-world complex requirements, expert-authored solutions, analytical knowledge used to interpret the requirements, technical knowledge applied in addressing the requirements, and explanations for the expert's solution design process.

```
prompt_for_get_solution_proposal_from_root = """<Instruction>:
In order to solve the following question, I need to search for relevant knowledge in external knowledge bases. Please generate a proposal based on the question and tell me what areas of knowledge I should search for.

<Question>:
{requirement}

<Proposal>:

"""



prompt_for_get_comment_proposal = """<Instruction>:
For the following question, there is currently a candidate solution. To evaluate this candidate solution, I need to search for relevant knowledge in external knowledge bases. Please generate a proposal based on the question and the candidate solution, and tell me what areas of knowledge I should search for.

<Question>:
{requirement}

<Candidate Solution>:
{solution}

<Proposal>:"""



prompt_for_get_solution_proposal = """<Instruction>:
For the following question, there is a candidate solution as well as an expert's evaluation of this candidate solution. In order to redesign a better solution, I need to search for relevant knowledge in external knowledge bases. Please generate a proposal based on the question and the candidate solution, and tell me what areas of knowledge I should search for.

<Question>:
{requirement}

<Candidate Solution>:
{old_solution}

<Critique for Candidate Solution>:
{comment}

<Proposal>:"""


prompt_for_get_solution_for_root = """<Instruction>:
Based on the reference knowledge, design a good solution for the question. Be sure to make full use of reference knowledge to analyze the challenges contained within the question and provide a comprehensive solution.

<Question>:
{requirement}

<Reference>:
{reference}

<Solution>:"""



prompt_for_get_doubt = """<Instruction>:
For the following question, a candidate solution has already been provided. You need to critique the candidate solution based on the reference knowledge. Be sure to make full use of the reference knowledge to identify the shortcomings of the old solution in terms of its analysis of the challenges in the question and its technical implementation.

<Question>:
{requirement}

<Candidate Solution>:
{solution}

<Reference>:
{reference}

<Critique>:"""


prompt_for_get_solution = """<Instruction>:
For the following question, an old solution has already been provided and its shortcomings have been pointed out by human experts. You need to redesign a better solution based on the reference knowledge and the guidance from human experts. Be sure to make full use of the reference knowledge to analyze the challenges contained within the question and provide a comprehensive solution.

<Question>:
{question}

<Candidate Solution>:
{old_solution}

<Critique for Candidate Solution>:
{comment}

<Reference>:
{reference}

<New Solution>:"""
```

Figure 7: Prompts used in node expansion of tree growth, including generating solution proposals and solutions based on the root node, generating comment proposals and comments based on a solution node, and generating solution proposals and solutions based on a comment node.

```
"""<<Task>>
{task}

<<Model-generated solution>>
{solution}

<<Judgement reference>>
## Analysis knowledge:
{analysis_knowledge}
## Technology knowledge:
{technology_knowledge}
## Goldn explanation:
{golden_explanation}
## Goldn solution:
{golden_solution}

<<Instruction>>
The above <<Task>> is a complex requirement in an actual engineering scenario. The above <<Model-generated solution>> is a solution generated by a certain model. You are required to evaluate this solution based on the <<Judgement reference>> annotated by a human expert. The <<Judgement reference>> consists of the following components:
(a) Analysis knowledge: A deep analysis of the various restrictive factors present in this complex requirement.
(b) Technology knowledge: A detailed explanation of the various technologies that must be used to solve this complex requirement.
(c) Golden explanation: An explanation of how to use these technologies to overcome various challenges.
(d) Golden solution: The standard solution provided by human experts.
Your evaluation of the <<Model-generated solution>> must fully consider the <<Judgement reference>>. The specific evaluation requirements are as follows.

<<Requirements>>
1. You need to evaluate the solution above from 2 dimensions. The range for each of the three scores is an integer between 0 and 100, where the minimum score is 0 and the maximum score is 100.
2. The scoring details for the 2 dimensions are as follows:
(2.1) Analysis Score: Refer to the aforementioned Analysis knowledge, Golden explanation, and Golden solution to assess whether the <<Model-generated solution>> has thoroughly considered the various restrictive factors in the <<Task>>. Pay special attention to listing each restrictive factor in the Analysis knowledge one by one and evaluating whether the model output has considered these factors. If considered, you need to specify which part of the <<Model-generated solution>> addresses the restrictive factor, and whether this part is sufficiently correct and specific.
(2.1.1) If no factors are considered, score 0.
(2.1.2) If factors are considered but the analysis is not entirely correct, score 11-30 depending on the degree of correctness.
(2.1.3) If factors are considered and the analysis is correct but not specific, score 31-60 depending on the level of specificity.
(2.1.4) If factors are considered, the analysis is correct, and it is specific, score 61-90 based on its similarity to the standard Analysis knowledge.
(2.1.5) If it is fully consistent with the standard Analysis knowledge, score 100.
(2.2) Technology Score: Refer to the aforementioned Technology knowledge, Golden explanation, and Golden solution to evaluate whether the <<Model-generated solution>> has employed appropriate technologies to address the challenges in the <<Task>>. Pay special attention to listing each technology in the Technology knowledge one by one and evaluating whether the model output has used these technologies. If used, you need to specify which part of the <<Model-generated solution>> utilizes the technology, and whether this part is sufficiently correct and specific.
(2.2.1) If no technologies are used, score 0.
(2.2.2) If technologies are used but not entirely correctly, score 11-30 depending on the degree of correctness.
(2.2.3) If technologies are used and correctly applied but not specific, score 31-60 depending on the level of specificity.
(2.2.4) If technologies are used correctly and specifically, score 61-90 based on their similarity to the standard Technology knowledge.
(2.2.5) If it is fully consistent with the standard Technology knowledge, score 100.
3. During the evaluation process, you must first evaluate the solution based on the three dimensions mentioned above and display your reasoning process. After completing your reasoning, you must output the identifier ##Scores## followed by your evaluation results in the form of a dictionary, i.e.: ##Scores## {{"Analysis Score": int, "Technology Score": int}}
4. Note that a longer solution is not necessarily better. The sole basis of your evaluation process is the aforementioned Judgement reference, and your evaluation results must fully take this reference into account.
5. Below is an example you can refer to when completing your evaluation:
## An example of evaluation:
1. For Analysis Score Evaluation. The analysis evaluates whether the solution adequately considers restrictive factors in the task. Let's analyze each factor from the Analysis knowledge and compare it with the model's solution:
For (analysis_0) Uneven soil particle distribution and high groundwater levels: The solution mentions groundwater analysis (in Geotechnical Investigation and Analysis) and dewatering systems (in Water Management). It also discusses preventing water ingress but does not explicitly address piping issues or the specific use of interlocking casing piles to stabilize the foundation pit. While the factors are considered, the analysis lacks detail and specificity regarding solutions such as technology for groundwater cut-off and particle stabilization. Conclusion: Analysis is present and mostly correct but lacks specificity in addressing key risks like piping and groundwater intrusion. Score: 60/100.
For (analysis_1) Deep excavation with moderately weathered limestone: The solution discusses slope stability analysis and phased excavation to control depth, as well as bracing and anchoring systems for stabilization. However, it does not explicitly address challenges such as settlement risks or hard rock excavation in limestone. The application of pre-applied axial force in steel supports is not mentioned, which is critical for ensuring stability. Conclusion: The analysis is partially correct but lacks depth and specificity regarding excavation in moderately weathered limestone and associated risks. Score: 50/100.
For (analysis_2) Sensitive surrounding environment: The solution acknowledges sensitivity in the surrounding environment and mentions real-time monitoring and mitigation measures (e.g., structural impact assessments, emergency response plans). However, it does not provide detailed measures for preventing settlement or damage to underground pipelines or nearby structures, such as dynamic monitoring and quantified safety parameters. Conclusion: The analysis is correct but not specific enough regarding risks to surrounding buildings and underground pipelines. Score: 60/100.
Overall Analysis Score: While the solution considers the main restrictive factors, it does not address all of them comprehensively or with sufficient specificity. Final Analysis Score: 57/100.
2. For Technology Score Evaluation. The evaluation checks whether appropriate technologies were employed. Let's analyze each technology from the Technology knowledge:
For (technology_0) Casing Interlocking Pile Technology: The solution does not explicitly mention casing interlocking pile technology, which is critical for ensuring foundation pit enclosure and managing groundwater effectively. Instead, it suggests general retaining structures like sheet piles and diaphragm walls, which are less specific for high groundwater conditions. Conclusion: This technology is not applied. Score: 0/100.
For (technology_1) Layered Excavation and Steel Support System: The solution mentions phased excavation and the use of steel supports and bracing but does not include details about pre-applied axial forces, which are vital for controlling deformation and ensuring stability. The specificity of this technology's application is lacking. Conclusion: Partially applied. Score: 50/100.
For (technology_2) Precipitation and Grouting Measures: The solution discusses dewatering systems and drainage channels but omits the use of grouting measures, which are essential for controlling groundwater levels and preventing leakage. Grouting was explicitly required in the reference but is absent here. Conclusion: Partially applied. Score: 40/100.
For (technology_3) Dynamic Monitoring and Protection Measures: The solution includes real-time monitoring with sensors and references emergency response measures. However, it does not quantify safety metrics, such as settlement rates, or specify dynamic adjustment strategies. These omissions limit the specificity of this technology's application. Conclusion: Partially applied. Score: 50/100.
Overall Technology Score: The solution employs some relevant technologies but omits critical ones (e.g., casing interlocking piles and grouting measures) and lacks specificity in others. Final Technology Score: 35/100.
Final Scores
##Scores## {{"Analysis Score": 57, "Technology Score": 35}}

<<Judgement>>"""
```

Figure 8: Prompts for calculating analytical score and technical score, which uses the golden solution, explanation, and corresponding analytical and technical knowledge as references, allowing GPT-4o to assess whether the system's solution sufficiently consider the challenges posed by the complex constraints and apply the appropriate technologies to address the complex constraints in the requirements.

Method	**Env.**		**Min.**		**Tra.**		**Aer.**		**Tel.**		**Arc.**		**Wat.**		**Far.**	
	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS
Single-round RAG Methods																
Naïve-RAG (Lewis et al., 2020)	59.7	56.9	55.6	37.8	60.8	56.8	68.6	66.7	69.4	67.5	68.4	63.5	64.6	56.8	62.6	59.2
Rerank-RAG (Li et al., 2023)	61.2	58.0	54.2	37.9	60.6	56.2	68.5	65.0	66.4	67.8	63.1	59.2	62.2	62.7	68.2	56.6
Multi-round RAG Methods																
Self-RAG (Asai et al., 2024)	60.3	62.0	56.8	40.3	61.1	56.5	65.4	64.2	70.5	71.4	69.2	64.8	60.8	59.5	66.6	64.4
GenGround (Shi et al., 2024)	43.8	36.5	38.2	21.5	52.6	41.8	53.5	39.6	62.5	48.3	53.8	45.8	54.9	47.2	55.8	42.7
RQ-RAG (Chan et al., 2024)	53.3	43.8	45.4	21.7	54.0	45.0	53.3	43.5	59.3	47.6	56.2	48.1	57.8	45.6	62.2	50.6
Tree-based Exploration and Bi-point Thinking																
SolutionRAG (Ours)	**63.0**	**63.1**	**58.0**	**42.4**	**64.2**	**62.4**	**70.8**	**68.2**	**72.2**	**73.0**	**71.0**	**71.3**	**65.9**	**65.2**	**70.2**	**65.6**

Table 5: Main experimental results based on Mistral-7B-Instruct-v0.3.

Method	**Env.**		**Min.**		**Tra.**		**Aer.**		**Tel.**		**Arc.**		**Wat.**		**Far.**	
	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS
Single-round RAG Methods																
Naïve-RAG (Lewis et al., 2020)	54.3	57.0	51.2	41.1	56.4	52.1	65.9	68.1	65.2	59.8	61.6	59.9	61.3	54.4	66.2	65.0
Rerank-RAG (Li et al., 2023)	51.7	52.3	48.4	31.9	54.6	51.6	62.3	64.8	63.3	63.4	59.8	53.1	60.9	49.6	65.9	59.4
Multi-round RAG Methods																
Self-RAG (Asai et al., 2024)	55.8	56.5	49.2	32.2	57.0	51.5	64.0	66.7	63.6	63.3	65.1	60.8	59.9	54.5	65.8	65.9
GenGround (Shi et al., 2024)	48.9	43.0	33.2	26.9	46.7	38.2	51.6	45.3	46.8	34.6	49.0	37.2	50.1	39.4	51.0	42.8
RQ-RAG (Chan et al., 2024)	46.1	39.7	41.8	17.9	51.1	40.4	47.7	33.8	51.6	38.0	54.3	43.3	48.2	36.2	55.4	46.6
Tree-based Exploration and Bi-point Thinking																
SolutionRAG (Ours)	**56.9**	**57.6**	**53.9**	**42.0**	**58.6**	**53.0**	**68.8**	**70.2**	**65.6**	**65.1**	**66.2**	**61.8**	**62.6**	**56.8**	**67.3**	**68.8**

Table 6: Main experimental results based on Llama-3.1-8B-Instruct.

Method	**Env.**		**Min.**		**Tra.**		**Aer.**		**Tel.**		**Arc.**		**Wat.**		**Far.**	
	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS
Single-round RAG Methods																
Naïve-RAG (Lewis et al., 2020)	61.2	58.0	56.4	36.0	62.4	54.7	67.0	64.2	62.6	65.6	66.2	62.1	63.6	54.0	64.4	61.5
Rerank-RAG (Li et al., 2023)	62.1	60.2	52.8	34.4	59.2	48.3	64.2	65.1	63.4	61.3	62.2	57.0	64.7	53.7	57.9	59.6
Multi-round RAG Methods																
Self-RAG (Asai et al., 2024)	63.0	**64.2**	54.1	35.0	62.9	55.4	68.6	69.2	64.5	65.4	64.5	64.7	63.9	56.6	61.0	58.6
GenGround (Shi et al., 2024)	50.1	37.5	50.6	29.9	53.3	36.0	54.8	45.3	56.5	48.3	57.8	50.2	58.2	46.5	58.1	51.4
RQ-RAG (Chan et al., 2024)	52.7	43.7	47.9	25.5	51.6	37.1	54.1	44.9	57.1	43.5	54.2	44.6	49.9	38.9	58.0	44.3
Tree-based Exploration and Bi-point Thinking																
SolutionRAG (Ours)	**65.6**	**64.2**	**58.6**	**44.6**	**63.6**	**57.7**	**69.8**	**71.5**	**66.1**	**67.8**	**67.6**	**66.9**	**64.9**	**59.8**	**66.9**	**64.3**

Table 7: Main experimental results based on Qwen2.5-3B-Instruct.

Method	**Env.**		**Min.**		**Tra.**		**Aer.**		**Tel.**		**Arc.**		**Wat.**		**Far.**	
	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS	AS	TS
Single-round RAG Methods																
Naïve-RAG (Lewis et al., 2020)	49.0	41.5	47.0	25.3	54.7	46.6	55.5	52.7	58.4	55.9	56.2	50.8	53.5	43.5	61.3	51.6
Rerank-RAG (Li et al., 2023)	50.6	43.5	50.2	31.2	50.0	41.2	53.9	49.9	59.5	54.8	50.6	40.8	54.6	49.1	60.6	52.8
Multi-round RAG Methods																
Self-RAG (Asai et al., 2024)	49.6	**44.2**	48.3	30.4	51.5	45.6	53.7	47.5	57.7	53.8	56.4	51.9	55.7	48.8	57.8	53.0
GenGround (Shi et al., 2024)	23.4	12.4	22.5	16.0	34.7	32.2	32.1	22.6	41.1	36.0	26.5	19.5	35.0	30.6	42.6	43.8
RQ-RAG (Chan et al., 2024)	42.8	29.6	46.0	25.0	46.5	33.1	41.6	34.6	45.4	30.9	46.3	35.4	50.1	38.2	49.3	37.7
Tree-based Exploration and Bi-point Thinking																
SolutionRAG (Ours)	**51.6**	**46.2**	**50.6**	**33.2**	**58.1**	**49.6**	**57.0**	**56.7**	**61.5**	**57.7**	**58.1**	**53.3**	**56.2**	**50.4**	**62.1**	**54.2**

Table 8: Main experimental results based on Qwen2.5-0.5B-Instruct.

Transportation	
Journal Name	ISSN
Railway Construction Technology	1009-4539
Northern Communications	1673-6052
China Municipal Engineering	1004-4655
Highway	0451-0712
Urban Roads Bridges & Flood Control	1009-7716
Technology Innovation and Application	2095-2945
Marine Equipment/Materials & Marketing	1006-6969
Engineering and Construction	1673-5781
Port Operation	1000-8969
Structural Engineers	1005-0159
China Highway	1006-3897
Engineering and Technological Research	2096-2789
Construction Machinery Technology & Management	1004-0005
TranspoWorld	1006-8872
Railway Investigation and Surveying	1672-7479
Transport Construction & Management	1673-8098
Guangdong Water Resources and Hydropower	1008-0112
Western China Communications Science & Technology	1673-4874
Jiangsu Science and Technology Information	1004-7530
Value Engineering	1006-4311
Hoisting and Conveying Machinery	1001-0785
Jiangxi Building Materials	1006-2890
Scientific and Technological Innovation	2096-4390
Transport Business China	1673-3681
Sichuan Cement	0451-0712

Aerospace	
Journal Name	ISSN
Spacecraft Engineering	1673-8748
Aeronautical Manufacturing Technology	1671-833X
Aviation Maintenance & Engineering	1672-0989
Journal of Ordnance Equipment Engineering	2096-2304
Aeroengine	2096-2304
Space International	2096-2304
Avionics Technology	1006-141X
System Simulation Technology	1673-1964
Journal of Civil Aviation	2096-4994
Safety & EMC	1005-9776
Internal Combustion Engine & Parts	1674-957X
Aeronautical Computing Technique	1671-654X
Meteorological Science and Technology	1671-6345
Journal of Astronautics	1000-1328
Communications Technology	1002-0802
Laser & Optoelectronics Progress	1006-4125
Engineering & Test	1674-3407
Chinese Space Science and Technology	1000-758X
Ship Electronic Engineering	1672-9730
China Science and Technology Information	1672-9730
Journal of Deep Space Exploration	2096-9287
China Educational Technology & Equipment	1671-489X
Micromotors	1671-489X
Spacecraft Recovery & Remote Sensing	1009-8518
Journal of Chengdu Aeronautic Polytechnic	1671-4024

Telecom	
Journal Name	ISSN
Systems Engineering and Electronics	1001-506X
Electronic Technology & Software Engineering	2095-5650
Video Engineering	1002-8692
Telecom Engineering Technics and Standardization	1008-5599
Radio & Television Network	2096-806X
Study on Optical Communications	1005-8788
Electronics Quality	1003-0107
Radio & Television Information	1007-1997
Changjiang Information & Communications	2096-9759
Automation in Petro-Chemical Industry	1007-7324
Telecommunications Science	1000-0801
Computer Knowledge and Technology	1009-3044
Journal of Electronics & Information Technology	1009-5896
Laser & Optoelectronics Progress	1006-4125
China Digital Cable TV	1007-7022
Radio Engineering	1003-3106
Journal of Beijing Electronic Science and Technology Institute	1672-464X
Laser Journal	0253-2743
Designing Techniques of Posts and Telecommunications	1007-3043
Wireless Internet Science and Technology	1672-6944
Journal of University of South China(Science and Technology)	1673-0062
Audio Engineering	1002-8684
Automation Application	1674-778X
Chinese Journal of Lasers	0258-7025
Journal of Smart Agriculture	2096-9902

Table 9: List of the engineering journals used for construction the benchmark.

Architecture	
Journal Name	ISSN
Building Technology Development	1001-523X
Building Structure	1002-848X
Construction & Design for Engineering	1007-9467
Modern Paint & Finishing	1007-9548
Architecture Technology	1000-4726
Theoretical Research in Urban Construction	2095-2104
Urban Architecture Space	2097-1141
Art and Design	1008-2832
Architecture & Culture	1672-4909
Journal of Yangzhou Polytechnic College	1008-3693
Heating Ventilating & Air Conditioning	1002-8501
Construction Machinery & Maintenance	1006-2114
China Science and Technology Information	1001-8972
Construction Machinery and Equipment	1000-1212
Journal of Municipal Technology	1009-7767
Jiangxi Building Materials	1006-2890
Urban Roads Bridges & Flood Control	1009-7716
Fujian Construction Science & Technology	1006-3943
Sichuan Cement	1007-6344
Engineering and Technological Research	2096-2789
Journal of North China Institute of Science and Technology	1672-7169
Tianjin Construction Science and Technology	1008-3197
World Forestry Research	1001-4241
Jiangsu Building Materials	1004-5538
Shanghai Construction Science & Technology	1005-6637

Water Resource	
Journal Name	ISSN
Design of Water Resources & Hydroelectric Engineering	1007-6980
Hydro Science and Cold Zone Engineering	2096-5419
Journal of Water Resources and Architectural Engineering	1672-1144
Mechanical & Electrical Technique of Hydropower Station	1672-5387
Yangtze River	1001-4179
Port & Waterway Engineering	1002-4972
Technical Supervision in Water Resources	1008-1305
Small Hydro Power	1007-7642
Pearl River	1001-9235
Water Conservancy Construction and Management	2097-0528
Water Conservancy Science and Technology and Economy	1006-7175
Water Resources Planning and Design	1672-2469
Construction Quality	1671-3702
Henan Water Resources and South-to-North Water Diversion	1673-8853
Engineering and Construction	1673-5781
Technology and Market	1006-8554
Beijing Water	1673-4637
Port Engineering Technology	2097-3519
Water Resources & Hydropower of Northeast China	1002-0624
Mechanical and Electrical Information	1671-0797
Maritime Safety	2097-1745
Gansu Water Resources and Hydropower Technology	2095-0144
Water Power	0559-9342
Shanxi Water Resources	1004-7042
Haihe Water Resources	1004-7328

Farming	
Journal Name	ISSN
Modern Agricultural Science and Technology	1007-5739
Farm Machinery	1000-9868
Cereal & Feed Industry	1003-6202
Journal of Agricultural Mechanization Research	1003-188X
Forestry Machinery & Woodworking Equipment	2095-2953
Transactions of the Chinese Society of Agricultural Engineering	1002-6819
Forest Research	1001-1498
Times Agricultural Machinery	2095-980X
Protection Forest Science and Technology	1005-5215
Journal of Beijing University of Agriculture	1002-3186
Contemporary Horticulture	1006-4958
China Southern Agricultural Machinery	1672-3872
Forest Inventory and Planning	1671-3168
Agricultural Machinery Using & Maintenance	2097-4515
Journal of Green Science and Technology	1674-9944
China Forest Products Industry	1001-5299
Forestry Machinery & Woodworking Equipment	2095-2953
The Food Industry	1004-471X
Journal of Hebei Forestry Science and Technology	1002-3356
Electrical Automation	1000-3886
Journal of Library and Information Science	2096-1162
Forest Science and Technology	2097-0285
Chinese Journal of Ecology	1000-4890
Popular Standardization	1007-1350
Management & Technology of SME	1673-1069

Table 10: List of the engineering journals used for construction the benchmark.