RoadMapper: A Multi-Agent System for Roadmap Generation of Solving Complex Research Problems

Jiacheng Liu[1] **Zichen Tang**[1] **Zhongjun Yang**[1] **Xinyi Hu**[1] **Xueyuan Lin**[2,3,4]
Linwei Jia[1] **Ruofei Bai**[1] **Rongjin Li**[1] **Shiyao Peng**[1] **Haocheng Gao**[1] **Haihong E**[1,*]
[1]Beijing University of Posts and Telecommunications
[2]The Hong Kong University of Science and Technology (Guangzhou)
[3]IDEA Research [4]Hithink RoyalFlush Information Network Co., Ltd.

bupt-reasoning-lab.github.io/RoadMapper
BUPT-Reasoning-Lab/RoadMapper BUPT-Reasoning-Lab/RoadMapper

Abstract

People commonly leverage structured content to accelerate knowledge acquisition and research problem solving. Among these, roadmaps guide researchers through hierarchical subtasks to solve complex research problems step by step. Despite progress in structured content generation, the **roadmap generation task** has remained unexplored. To bridge this gap, we introduce **RoadMap**, a novel benchmark designed to evaluate the ability of large language models (LLMs) to construct high-quality roadmaps for solving complex research problems. Based on this, we identify three limitations of LLMs: (1) *lack of professional knowledge*, (2) *unreasonable task decomposition*, and (3) *disordered logical relationships*. To address these challenges, we propose **RoadMapper**, an LLM-based multi-agent system that decomposes the research roadmap generation task into three key stages (*i.e.*, initial generation, knowledge augmentation, and iterative "critique-revise-evaluate"). Extensive experiments demonstrate that RoadMapper can improve LLMs' ability for roadmap generation, while enhancing average performance by more than **8%** and **saving 84% of the time** required by human experts, highlighting its effectiveness and application potential.

1 Introduction

Designing a structured roadmap for complex research problems is an important task in scientific research and education (Burian et al., 2010). Solving these research problems often faces challenges such as a wide range of knowledge and rapid technological iteration. A meticulously designed roadmap can break down these research problems into multiple subtasks, thus improve the efficiency and quality of solutions (Sorensen et al., 2024).

Currently, designing research roadmaps heavily relies on human experts who create them by consulting professional knowledge, meticulously designing, and iteratively conducting reviews. However, this manual process is both time-consuming and resource-demanding. Fortunately, recent advances in LLMs (Park and Kim, 2025) present an opportunity to develop an automatic system to generate high-quality roadmaps.

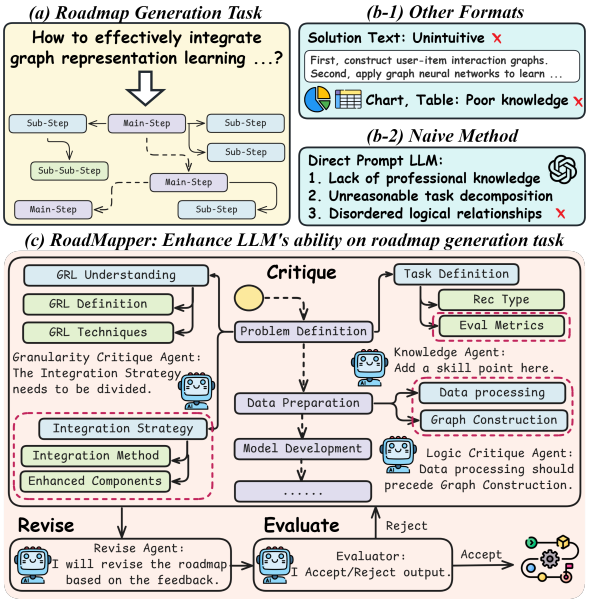

Figure 1: We propose the *roadmap generation task*. Prior answer formats and methods face multiple challenges when solving complex research problems, but **RoadMapper** effectively addresses these challenges through an iterative "critique-revise-evaluate" process.

However, as shown in Table 1, current research has not sufficiently explored the roadmap generation task, usually suffering from three limitations:

- **Limited Field and Knowledge Coverage.** Complex research problems typically span multiple professional fields, and their solutions similarly demand the integration of expertise from diverse disciplines. However, existing research focuses primarily on a few specific fields (Li et al., 2023; Deng et al., 2024).

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 7653–7676

Research	Basic Statistics		Field and Knowledge			Professional Depth	Guidance	
	# Samples	Method	Field	Type	Language	Material	Output Format	Application
Seq2Seqset (Li et al., 2023)	4,855	Seq2Seq	1	1	EN	News Report	Table	I
Text-Tuple-Table (Deng et al., 2024)	3,771	Prompting	1	1	EN	Live Commentary	Table	I
End-to-End Parsing (Bhatt et al., 2024)	10,300	End2End	1	1	EN	Cooking Recipes	Flow Graph	G
StructSum (Jain et al., 2024)	200	Prompting	✘	1	EN	Wiki Pages	Table+Mindmap	I
WHPG (Ren et al., 2023)	283	End2End	✘	1	EN	Wiki Pages	Procedural Graph	G
EMGN (Hu et al., 2021)	44,585	End2End+RL	✘	1	EN	News Articles	Mindmap	I
COMET (Bosselut et al., 2019)	977,000	End2End	✘	✘	EN	✘	Knowledge Graph	G
DeepSolution (Li et al., 2025b)	3,024	RAG	8	1	EN	Technical Reports	Solution Text	G
METAL (Li et al., 2025a)	1,000	MAS	2	2	EN	Chart + Instruction	Chart Code	I
RoadMapper	**1,705**	**MAS**	**10**	**5**	**EN+CN**	**Dissertations**	**Roadmap (ours)**	**G**

Table 1: Comparison between **RoadMapper** and related research on structured content generation. **EN**: English, **CN**: Chinese, **I**: Information Display, **G**: Guide. RoadMapper offers three key advantages: (1) *Field and knowledge coverage* are ensured by multiple fields, diverse types, and bilingual support; (2) *Professional depth* is ensured by dissertations written by master and Ph.D.; (3) *Step-by-step guidance* is provided through structured roadmap format.

- **Insufficient Professional Depth.** Solving complex research problems requires integrating professional knowledge for deep analysis. However, existing research usually focuses on shallow tasks, such as extracting cooking flow graphs from recipes (Bhatt et al., 2024) or converting text to mindmaps (Hu et al., 2021).

- **Lack of Step-by-Step Guidance.** Logically coherent roadmaps help guide people to solve complex research problems step by step. However, existing research in formats such as tables and knowledge graphs is mainly designed for information display rather than offering guidance (Li et al., 2023; Bosselut et al., 2019).

To bridge this gap, we construct **RoadMap**, a novel benchmark focusing on evaluating LLMs' capabilities for generating high-quality roadmaps of solving research problems, **covering 10 research fields, 5 research types, and 2 languages** (*i.e.*, English and Chinese). RoadMap comprises two sub-components: (1) ***Skill-Repo***: This component consists of 8,436 professional skill points, each of which contains a name, a detailed description, and a problem it solves, serving to provide LLMs with extensive professional knowledge; (2) ***Golden-Roadmap***: This component consists of 1,705 complex research problems, with each accompanied by a golden roadmap that was meticulously designed by experts and serves as a reference for evaluating roadmaps from automatic systems.

We conduct extensive experiments on RoadMap and find that direct or repeated prompting of LLMs faces three main challenges: ***Q1****: lack of professional knowledge,* ***Q2****: unreasonable task decomposition, and* ***Q3****: disordered logical relationships*, exceeding the capabilities of a single model.

To address these challenges, we propose our **RoadMapper**, a multi-agent system that emulates the manual process of human experts and decomposes the roadmap generation task into three stages: initial generation, knowledge augmentation, and iterative "critique-revise-evaluate". Specifically, **(1)** the *Init agent* generates an initial roadmap based on the research problem; **(2)** the *Knowledge agent* augments the initial roadmap by knowledge from Skill-Repo, addressing Q1; **(3)** the *Granularity Critique agent* analyzes the decomposition granularity of sub-task nodes and outputs revision suggestions, addressing Q2; **(4)** the *Logic Critique agent* analyzes the logical relationships between sub-task nodes and outputs revision suggestions, addressing Q3; **(5)** the *Revise agent* implements revisions to the roadmap based on suggestions from the critique agents; **(6)** the *Evaluate agent* evaluates the roadmap quality to determine whether to output as final result. To keep the evaluation aligned with human expert, we train the Evaluate agent using **Direct Preference Optimization (DPO) algorithm**, leveraging data from RoadMap and its construction process. These LLM-driven agents will collaborate iteratively until the roadmap reaches expected quality or the maximum number of iterations.

Finally, we conduct extensive experiments on RoadMap and evaluate using **four novel metrics** from both structure (*i.e.*, **DegreeScore**, **DepthScore**) and content (*i.e.*, **StepScore**, **LogicScore**). The results demonstrate that RoadMapper improves LLMs' performance on roadmap generation tasks: **addressing Q1 by improving StepScore by 8.24, Q2 by improving structure metrics by 7.04, and Q3 by improving LogicScore by 7.79**. Meanwhile, RoadMapper can reduce designing time by more than **84%** compared to human experts.

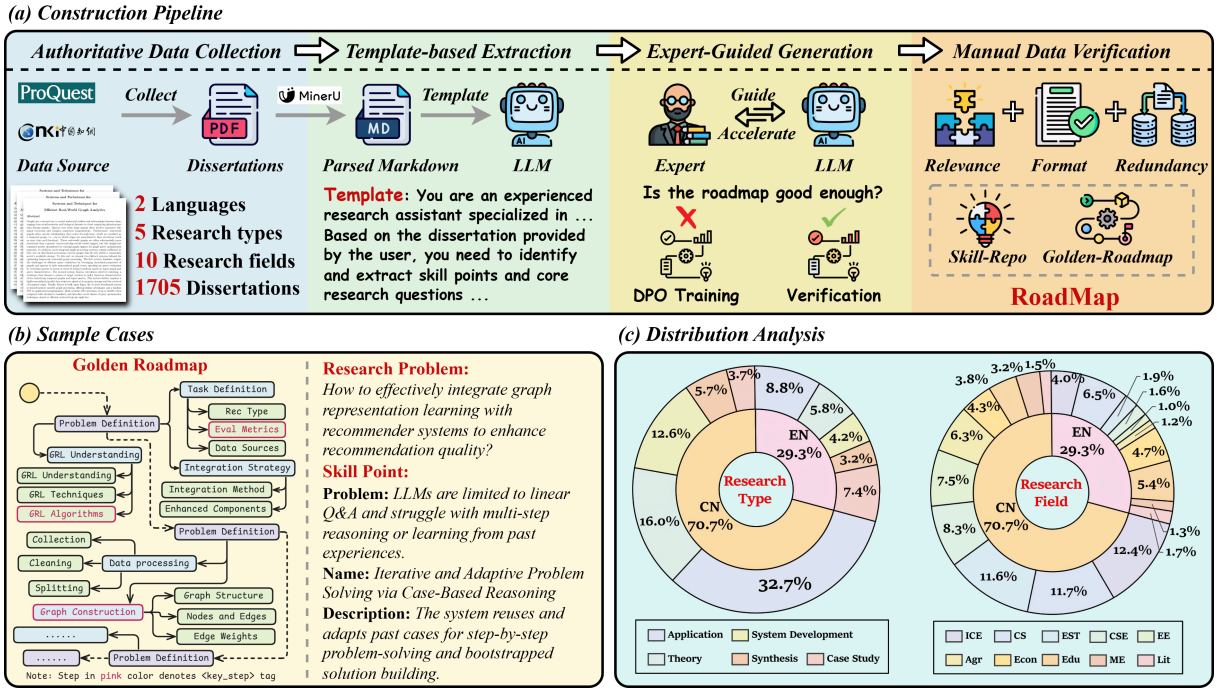

Figure 2: Overview of **RoadMap**. **ICE**: Information and Communication Engineering, **CS**: Computer Science, **EST**: Electronic Science and Technology, **CSE**: Control Science and Engineering, **EE**: Electrical Engineering, **Agr**: Agronomy, **Econ**: Economics, **Edu**: Education, **ME**: Mechanical Engineering, **Lit**: Literature.

Our main contributions are as follows:

- **We define the roadmap generation task and construct RoadMap**, providing extensive professional knowledge for roadmap generation systems and effectively evaluating the performance of systems on this task.
- **We propose RoadMapper**, a multi-agent system that emulates human experts to decompose the generation task into three iterative stages which are handled by six LLM-driven agents aligned with experts via DPO training.
- **We introduce four novel evaluation metrics and conduct extensive experiments**, with results demonstrating that RoadMapper can significantly address challenges faced by LLMs and generate well-designed roadmaps with strong structure and content efficiently.

2 Task Definition

Mapping $\mathcal{F}$ describes the **roadmap generation task**:

$$\mathcal{F}: x_{\text{problem}} \rightarrow y_{\text{roadmap}}. \quad (1)$$

That is, for a given research problem x_{problem}, the target is to output the corresponding y_{roadmap}, which is defined as a logically tree-like roadmap guiding people to solve x_{problem} step by step.

We represent roadmaps using Markdown documents that adhere to the following structural rules: (1) Each line represents a task node, including `level`, `index`, and `title`; (2) The `level` is consistent with Markdown's heading syntax, composed of several # symbols; (3) The `index` satisfies the regular expression `(\d+(?:\.\d+)*)`, such as `1.3.2`; (4) The `title` is enclosed in square brackets.

3 RoadMap Benchmark

We introduce the construction of RoadMap in this section, with its overview in Figure 2, key statistics in Table 2, and additional details in Appendix A.

3.1 Authoritative Data Collection

We collect dissertations from ProQuest and CNKI, adopting the following strategies: (1) published since 2018, (2) authored by graduate students, and (3) from universities with strong academic programs. These dissertations offer three key advantages: ***Strong Relevance***. Dissertations address cutting-edge and complex research problems that align with our focus; ***Extensive Professional Knowledge***. Dissertations contain complete solutions, detailed processes and comprehensive results suitable for knowledge extraction; ***Authoritative Content***. Expert scrutiny ensures high-quality data with consistently rigorous logic.

Property	Value
# Golden Roadmaps (EN/CN)	1,705 (500/1,205)
# Skill Points (EN/CN)	8,436 (2,493/5,943)
# Research Fields	10
# Research Types	5
# Avg. Key Steps	21.60
# Avg. Depth (Knowledge Depth)	3.22
# Avg. Out-Degree (Knowledge Breadth)	3.16
# Avg. Leaf Nodes (Knowledge Richness)	86.46

Table 2: Key Statistics of RoadMap.

3.2 Template-Based Extraction

We first parse the PDF files into Markdown (Wang et al., 2024a) and filter out irrelevant information. Then, we manually format a template for information extraction via `Gemini 2.5 Flash`, based on generative information extraction (Lu et al., 2022; Xu et al., 2024a; Zhang et al., 2025). The template requires the following information from each dissertation: (1) ***Core research problem***, which is the main scientific or technical problem the dissertation aims to solve; (2) ***Skill Points***, which are the critical technologies used in the dissertation to address the research problem, each including its name, a detailed description, and the problem it solves. Finally, we save the results in JSON format.

3.3 Expert-Guided Generation

We assemble experts from diverse research fields to develop a golden roadmap for each research problem. These experts are allowed to leverage `Gemini 2.5 Flash` to accelerate development, while remaining primarily guided by their own experience. Specifically, experts first synthesize the main framework of roadmaps based on their domain knowledge and relevant dissertations. Subsequently, this framework is used to prompt LLMs for generating initial drafts. Experts then iteratively guide the model to refine problematic components until a high-quality roadmap is achieved. Finally, experts annotate critical steps in the roadmap using `<key_step>`, which are essential for experimental evaluation. Notably, problematic roadmaps will not be discarded as they are valuable for DPO training.

3.4 Manual Data Verification

We implement rigorous verification processes to prevent errors from LLMs (*e.g.*, hallucinations) and ensure benchmark quality (Huang et al., 2025): (1) ***Relevance Verification***. Two experts cross-validate the model-extracted results to determine if they faithfully match the corresponding dissertations. Any mismatched content will be removed. In cases of disagreement, one additional expert will be introduced for arbitration; (2) ***Format Verification***. We develop an automated Python script to validate the formatting of roadmaps and skill points against predefined requirements. Any incorrect content (about 19.3%) will be manually revised by experts; (3) ***Redundancy Removal***. We use `Qwen3 Embedding 8B` to help identify skill points with high similarity. Any duplicate contents (about 5.3%) will be discarded or manually merged by experts depending on the type of redundancy.

4 RoadMapper Methodology

We introduce the agents design of RoadMapper and the DPO training of the Evaluate agent in this section, with additional details shown in Appendix B.

4.1 Agents Design

RoadMapper comprises six LLM agents prompted by carefully designed prompts, shown in Figure 3.

4.1.1 Init Agent ($\mathcal{I}$)

This agent is responsible for receiving research problems and generating an initial roadmap draft:

$$\mathcal{I} : y_{\text{initial}} = \mathcal{I}(x_{\text{problem}}). \quad (2)$$

Here, x_{problem} denotes the complex research problem and y_{initial} denotes the initial roadmap.

4.1.2 Knowledge Agent ($\mathcal{K}$)

This agent is responsible for augmenting the initial roadmap based on skill points from Skill-Repo:

$$\mathcal{K} : y_0 = \mathcal{K}(knowledge, y_{initial}). \quad (3)$$

Here, $knowledge$ denotes knowledge composed of the top-K skill points retrieved from Skill-Repo via vector similarity, $y_{initial}$ denotes the initial roadmap, and y_0 denotes the augmented roadmap.

4.1.3 Logic Critique Agent ($\mathcal{L}$)

This agent is responsible for critiquing the logic between nodes and outputting revision suggestions:

$$\mathcal{L} : LC_t = \mathcal{L}(y_t). \quad (4)$$

Here, y_t denotes the roadmap to be critiqued and LC_t denotes the logical revision suggestions.

We define two types of logical relationships: (1) ***Parent-child Logic***: A child node must represent a direct refinement or an execution step of its parent node's task; (2) ***Sibling Logic***: Sibling nodes of the same parent must exhibit a parallel-progressive relationship among their respective tasks.

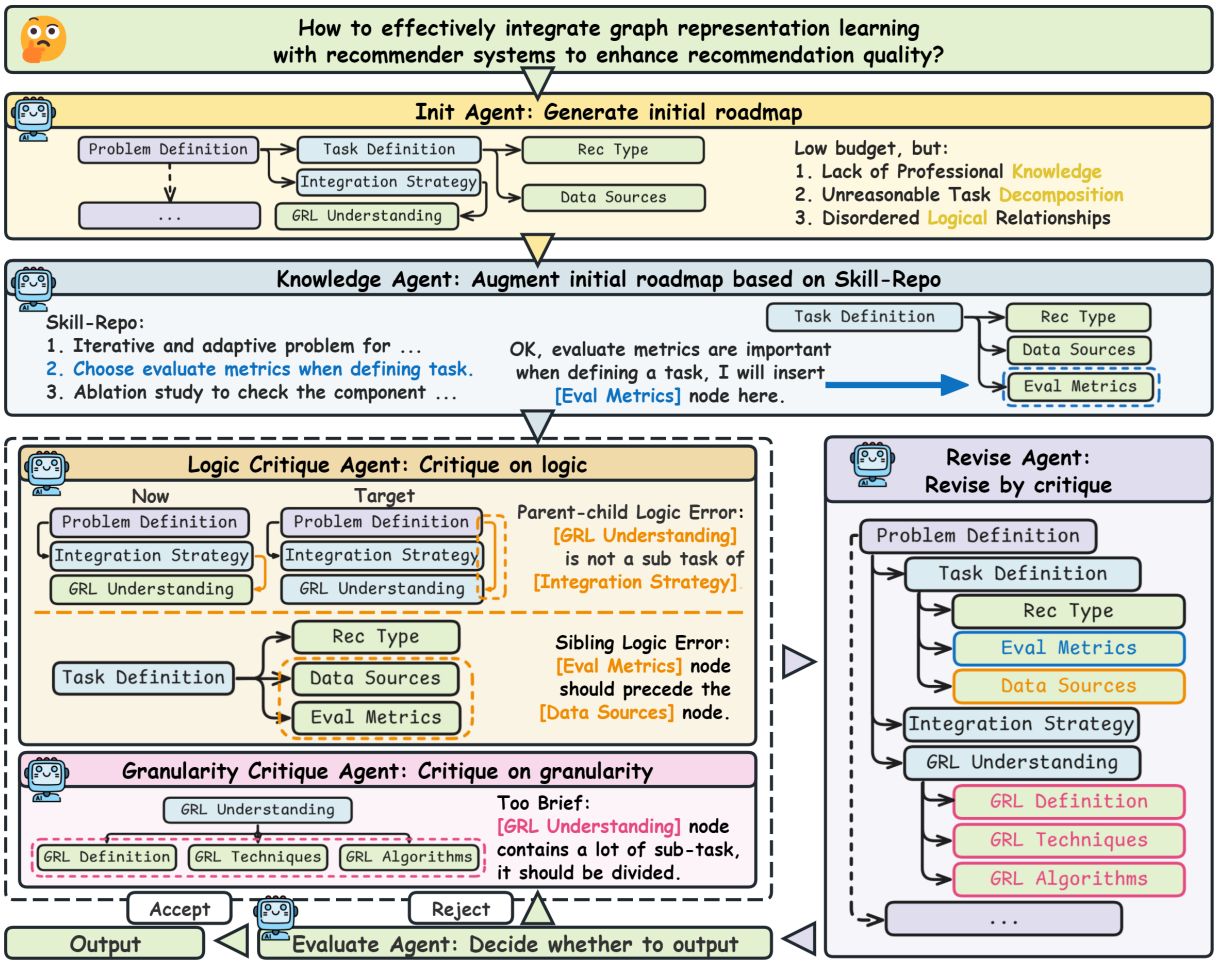

Figure 3: Overview of **RoadMapper**, a multi-agent system consisting of six agents: Init Agent ($\mathcal{I}$), Knowledge Agent ($\mathcal{K}$), Logic Critique Agent ($\mathcal{L}$), Granularity Critique Agent ($\mathcal{G}$), Revise Agent ($\mathcal{R}$), and Evaluate Agent ($\mathcal{E}$).

4.1.4 Granularity Critique Agent ($\mathcal{G}$)

This agent is responsible for critiquing the granularity of nodes and outputting revision suggestions:

$$\mathcal{G} : GC_t = \mathcal{G}(y_t). \tag{5}$$

Here, y_t denotes the roadmap to be critiqued and GC_t denotes the granularity revision suggestions.

We define two types of inappropriate granularity: (1) ***Too Detailed***: A node is split into an excessive number of trivial subtasks, introducing unnecessary complexity and inefficiency; (2) ***Too Brief***: A node remains information-dense and requires further decomposition, causing difficulty in comprehension.

4.1.5 Revise Agent ($\mathcal{R}$)

This agent is responsible for revising the roadmap based on revision suggestions from $\mathcal{L}$ and $\mathcal{G}$ agents:

$$\mathcal{R} : y_{t+1} = \mathcal{R}(y_t, LC_t, GC_t). \tag{6}$$

Here, y_t denotes the roadmap to be revised, LC_t denotes the logical revision suggestions, GC_t denotes the granularity revision suggestions, and y_{t+1} denotes the revised roadmap.

4.1.6 Evaluate Agent ($\mathcal{E}$)

This agent is responsible for evaluating roadmaps and outputting objective evaluation outcomes:

$$\mathcal{E} : E_t = \mathcal{E}(y_{t+1}). \tag{7}$$

Here, y_{t+1} denotes the roadmap to be evaluated and E_t denotes the evaluation outcome.

E_t comprises a score and a corresponding reason. The "critique-revise-evaluate" iteration will end if the score achieves a predefined *passing score*.

4.2 DPO Training of Evaluate Agent

The reliability of agent $\mathcal{E}$ is critical to RoadMapper's efficiency and roadmap quality, yet it remains susceptible to degradation from model biases. To mitigate this issue, we employ DPO training to align $\mathcal{E}$ agent with domain experts. Specifically, we use Qwen3 8B/14B/32B as backbone models due to their demonstrated exceptional capabilities.

Algorithm 1 Inference Procedure of RoadMapper

Input: x_{problem}: Research problem

Output: y^*: Final refined output.

$y_{\text{initial}} \leftarrow \mathcal{I}(x_{\text{problem}})$
$y_{\text{knowledge}} \leftarrow \mathcal{K}(y_{\text{initial}}, \text{knowledge})$
$y_0 \leftarrow y_{\text{knowledge}},\ t \leftarrow 0$
while $t < T_{\max}$ **do**
  $LC_t \leftarrow \mathcal{L}(y_t),\ GC_t \leftarrow \mathcal{G}(y_t)$
  $y_{t+1} \leftarrow \mathcal{R}(y_t, LC_t, GC_t)$
  $E_t \leftarrow \mathcal{E}(y_{t+1})$
  if $E_t =$ ACCEPT **then**
    return y_{t+1}
  else
    $t \leftarrow t + 1$
  end if
end while
return y_t as y^*

Table 3: Inference Procedure of RoadMapper.

Preference Dataset Construction. We first collect roadmaps from RoadMap and its construction. For each roadmap x, we employ `Qwen3-32B` to generate 10 evaluation candidates, using the same prompt with $\mathcal{E}$ agent. Then, seven experts will vote to select samples: (1) The candidate with **highest** number of votes is selected as the positive sample y_w; (2) Crucially, we select the candidate receiving the **second-highest** number of votes as the negative sample y_l (rather than the lowest-ranked one) because the second-best evaluations often highly resemble the optimal ones yet exhibit subtle flaws, which enables the model to discern fine-grained expert preferences. We also introduce format preference pairs to improve the formatting accuracy of $\mathcal{E}$ agent. Finally, our preference dataset $\mathcal{D}$ contains 818 roadmaps with corresponding preference pairs.

Optimization Objective. We fine-tune the $\mathcal{E}$ agent, parameterized as π_θ, to maximize the relative log-likelihood of the preferred evaluation y_w over the dispreferred y_l, constrained by the reference model π_{ref}. The objective is formulated as:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)/\pi_{\text{ref}}(y_w|x)}{\pi_\theta(y_l|x)/\pi_{\text{ref}}(y_l|x)}\right)\right]. \quad (8)$$

Here, $\mathcal{D}$ denotes the preference dataset, σ denotes the logistic sigmoid function, and β is a hyperparameter regulating the deviation from the reference policy. This optimization effectively aligns the agent's internal scoring standards with expert judgment, ensuring the reliability and convergence of the "critique-revise-evaluate" loop.

4.3 Inference Procedure

We present the inference procedure of RoadMapper in Table 3: When a complex research problem is inputted, the $\mathcal{I}$ agent first generates an initial roadmap, which is then augmented with knowledge by the $\mathcal{K}$ agent. Subsequently, the system enters an iterative "critique-revise-evaluate" process, supported by agents $\mathcal{L}$, $\mathcal{G}$, and $\mathcal{R}$. This process continues until the roadmap achieves the *passing score* or reaches the maximum number of iterations.

5 Experiments

We present the experimental setup, main results, and a series of comprehensive analyses in this section, with more details shown in Appendix C.

5.1 Experimental Setup

Evaluation Metrics. We evaluate from structure (*i.e.*, **DegreeScore** and **DepthScore**) and content (*i.e.*, **StepScore** and **LogicScore**). Since content metrics are not amenable to rule-based calculation, we align with established Long-form QA evaluation methods (Tan et al., 2024; Wang et al., 2024b) and employ `GPT-4o mini` to score the generated roadmaps with reference to the golden ones.

- **DegreeScore** represents the magnitude of out-degree differences between the output and the golden roadmap, computed by:

$$DegreeScore = \left(1 - \frac{|Deg_g - Deg_o|}{Deg_g}\right) \times 100. \quad (9)$$

  Here, Deg_g and Deg_o denote the out-degree of the golden roadmap and the output roadmap.

- **DepthScore** represents the magnitude of depth differences between the output and the golden roadmap, computed by:

$$DepthScore = \left(1 - \frac{|Dep_g - Dep_o|}{Dep_g}\right) \times 100. \quad (10)$$

  Here, Dep_g and Dep_o denote the depth of the golden roadmap and the output roadmap.

- **StepScore** represents the key step score of the roadmap, which reflects the effectiveness of the roadmap. A higher score indicates that it better embodies the key steps marked in the golden roadmap (introduced in Section 3.3).

- **LogicScore** represents the internal logical score of the roadmap, which reflects the logical coherence of the roadmap. A higher score indicates that it aligns more closely with the research methodology of the golden roadmap.

Base Model	Method	English					Chinese					Overall
		SS	LS	DegS	DepS	Avg.	SS	LS	DegS	DepS	Avg.	Avg.
Llama 3.1 8B	DP	42.58	54.29	64.11	83.96	61.24	37.63	48.81	70.23	83.37	60.01	60.62
	BN	44.24	54.88	64.23	84.17	61.88	39.75	48.97	70.94	83.66	60.83	61.36
	Ours	**49.98**	**62.23**	**71.95**	**86.57**	**67.68**	**48.51**	**58.79**	**76.36**	**88.15**	**67.95**	**67.82**
Llama 4 Maverick	DP	46.78	56.91	61.92	86.97	63.15	46.14	55.61	64.37	88.36	63.62	63.38
	BN	46.63	56.47	61.95	87.30	63.09	46.29	56.39	64.44	88.59	63.93	63.51
	Ours	**54.33**	**65.34**	**69.81**	**92.25**	**70.43**	**53.11**	**64.35**	**74.32**	**93.30**	**71.27**	**70.85**
Qwen3-14B	DP	54.76	62.64	73.56	86.04	69.25	56.39	64.11	74.06	90.13	71.17	70.21
	BN	54.83	62.94	73.81	86.93	69.63	56.64	64.19	74.88	90.01	71.43	70.53
	Ours	**61.78**	**69.96**	**75.57**	**89.32**	**74.16**	**64.74**	**71.70**	**75.99**	**91.61**	**76.01**	**75.08**
GPT-4o mini	DP	45.73	56.02	66.31	84.86	63.23	43.97	53.05	67.92	85.73	62.67	62.95
	BN	46.09	56.69	66.89	84.86	63.63	44.16	53.45	68.27	86.50	63.10	63.36
	Ours	**47.63**	**58.99**	**72.41**	**90.09**	**67.28**	**46.03**	**57.00**	**74.57**	**90.74**	**67.09**	**67.18**
gpt-oss-20b	DP	55.04	65.47	80.05	81.41	70.49	53.05	60.50	80.88	81.97	69.10	69.80
	BN	55.74	65.89	80.65	81.99	71.07	53.83	60.59	81.37	81.97	69.44	70.25
	Ours	**58.21**	**66.75**	**85.33**	**89.20**	**74.87**	**59.60**	**66.88**	**87.29**	**88.81**	**75.65**	**75.26**
Claude 3 Haiku	DP	46.12	59.28	79.84	78.35	65.90	45.97	58.64	80.37	82.90	66.97	66.43
	BN	46.75	59.46	80.12	79.40	66.43	46.35	58.97	81.36	82.97	67.41	66.92
	Ours	**52.33**	**66.74**	**84.30**	**87.37**	**72.69**	**50.89**	**64.35**	**84.33**	**90.11**	**72.42**	**72.55**
Mistral Small 3.2	DP	47.02	58.20	71.11	87.87	66.05	48.58	59.91	74.34	86.18	67.25	66.65
	BN	48.10	58.41	71.52	88.11	66.54	48.35	59.57	74.34	86.32	67.15	66.84
	Ours	**54.27**	**64.13**	**76.59**	**90.13**	**71.28**	**55.12**	**63.00**	**83.95**	**91.47**	**73.39**	**72.33**
Llama 3.3 70B	DP	44.26	56.02	66.89	85.48	63.16	38.16	50.94	70.74	84.30	61.04	62.10
	BN	45.15	56.49	66.97	85.87	63.62	40.24	51.16	71.50	84.55	61.86	62.74
	CoT	45.39	57.10	67.21	85.83	63.88	39.66	51.19	71.72	84.93	61.88	62.88
	ReConcile	48.28	57.18	69.74	88.91	66.03	45.45	57.21	75.89	87.34	66.47	66.25
	DyLAN	47.64	58.19	71.76	87.97	66.39	46.86	56.70	75.24	87.32	66.53	66.46
	Ours	**52.50**	**63.81**	**75.79**	**90.65**	**70.69**	**50.05**	**60.93**	**78.94**	**90.89**	**70.20**	**70.45**
Gemini 3 Flash Preview	DP	54.09	63.35	64.22	89.59	67.81	57.12	65.89	68.67	90.63	70.58	69.20
	BN	54.70	63.48	64.81	89.95	68.24	57.19	66.23	69.21	90.79	70.86	69.55
	CoT	55.11	63.64	65.70	89.72	68.54	57.64	66.68	69.23	91.24	71.20	69.87
	ReConcile	58.62	67.91	74.89	91.33	73.19	61.22	68.50	76.08	90.82	74.16	73.67
	DyLAN	57.29	67.37	77.54	90.94	73.29	59.60	68.81	74.49	91.28	73.55	73.42
	Ours	**60.47**	**70.27**	**81.73**	**93.07**	**76.39**	**64.91**	**72.34**	**81.88**	**93.64**	**78.19**	**77.29**
Qwen3-235B-A22B	DP	57.65	67.30	74.90	86.20	71.51	60.35	66.62	79.66	90.57	74.30	72.91
	BN	57.99	67.61	75.75	86.41	71.94	60.87	67.84	80.17	90.57	74.86	73.40
	CoT	58.20	67.96	75.46	86.53	72.04	60.76	67.61	80.37	90.84	74.90	73.47
	ReConcile	60.22	68.74	78.82	89.97	74.44	63.04	69.52	84.69	91.79	77.26	75.85
	DyLAN	61.42	68.97	80.06	87.89	74.59	61.81	70.53	85.24	91.22	77.20	75.89
	Ours	**63.13**	**71.36**	**83.22**	**91.36**	**77.27**	**66.46**	**71.94**	**88.36**	**93.82**	**80.15**	**78.71**
DeepSeek-V3.2	DP	57.25	66.21	74.52	86.14	71.03	59.85	66.95	77.59	88.42	73.20	72.12
	BN	57.93	66.43	74.79	87.02	71.54	60.47	67.38	77.72	88.92	73.62	72.58
	CoT	58.46	66.87	75.68	86.71	71.93	60.83	67.53	77.86	88.08	73.58	72.75
	ReConcile	63.44	69.82	81.12	89.77	76.04	65.98	69.70	83.56	90.54	77.45	76.74
	DyLAN	64.37	69.96	82.20	91.47	77.00	64.55	69.41	86.84	90.04	77.71	77.36
	Ours	**66.36**	**72.21**	**85.64**	**92.35**	**79.14**	**67.62**	**72.46**	**87.32**	**92.93**	**80.08**	**79.61**

Table 4: Performance comparison across different models and methods on two splits. **DP**: Direct Prompting, **BN**: Best-of-N, **Ours**: RoadMapper with DPO training on Qwen3-32B, **SS**: StepScore, **LS**: LogicScore, **DegS**: DegreeScore, **DepS**: DepthScore. The top two results are highlighted in **bold** and underlined, respectively.

Baselines. We compare RoadMapper with several baseline methods, including different prompting strategies and multi-agent systems.

- **Direct Prompting** prompts the model with the research problem and the roadmap generation task. In our implementation, it uses the same instruction template as the $\mathcal{I}$ agent.
- **Best-of-N** strategy executes Direct Prompting independently for N times and selects the best output as the final result.
- **CoT** (Wei et al., 2022) prompting encourages the model to explicitly decompose the research problem into intermediate reasoning steps before producing the final roadmap.

- **ReConcile** (Chen et al., 2024) organizes multiple agents in a round-table discussion, conducting multiple rounds of deliberation where agents attempt to persuade each other to improve the roadmap and ultimately reach a consensus.
- **DyLAN** (Liu et al., 2024) adopts a dynamic workflow paradigm, where a subset of agents is dynamically selected to participate in roadmap generation, rather than involving all agents.

Implementation Details. **(1)** We conduct experiments on 11 LLMs, including both open-source and proprietary models of various sizes from different manufacturers; **(2)** The maximum number of iterations for the "critique-revise-evaluate" process is set to 5; **(3)** The *passing score* of $\mathcal{E}$ agent is set to 80; **(4)** The maximum number of skill points that $\mathcal{K}$ agent retrieves from Skill-Repo is set to 30; **(5)** We use Qwen3 32B with DPO training as $\mathcal{E}$ agent.

5.2 Main Results

Table 4 presents the main results of our experiments. There are two main conclusions as follows:

Prior methods cannot effectively address the task of generating research roadmaps. Our experimental results reveal that direct prompting leads to suboptimal performance across all metrics. For instance, Llama 3.3 70B attained an average score of 61.04 and a notably low StepScore of 38.16 on the Chinese split. Best-of-N and CoT improved performance by increasing computation, but the gains are marginal and inconsistent. ReConcile and DyLAN achieved limited improvements via multi-agent strategies. For example, the average improvement of Qwen3-235B-A22B is only 2.98.

RoadMapper significantly improves the performance of LLMs in roadmap generation tasks. Data show that RoadMapper achieves SOTA performance across all base models in English and Chinese splits. For instance, compared to Direct Prompting, the average performance improvements of Llama 3.3 70B on English and Chinese splits are **7.53** and **9.16**, respectively, and DeepSeek-V3.2 also achieves average performance improvements of **8.11** and **6.88**, respectively. Moreover, compared with ReConcile, RoadMapper delivers a further stable improvement of **4.20** for Llama 3.3 70B. This enhancement exhibits the same trend across models and metrics, demonstrating the effectiveness, robustness, and generalizability of RoadMapper.

Method	SS	LS	DegS	DepS	Avg.
w/o $\mathcal{K}$ agent	45.44	59.60	73.11	88.76	66.73
w/o $\mathcal{L}$ agent	48.29	56.94	74.17	88.46	66.97
w/o $\mathcal{G}$ agent	47.51	58.74	74.80	87.37	67.11
w/o DPO	49.83	60.20	74.67	88.25	68.24
DPO-8B	49.27	58.47	73.02	87.62	67.10
DPO-14B	49.77	60.69	75.30	88.87	68.66
SC	48.52	58.98	74.29	87.21	67.25
RoadMapper	**51.28**	**62.37**	**77.37**	**90.77**	**70.45**

Table 5: Ablation study of RoadMapper. **w/o**: without, **Splits**: English & Chinese, **Base Model**: Llama 3.3 70B, **DPO-8B**: use Qwen3-8B as backbone, **DPO-14B**: use Qwen3-14B as backbone, **SC**: merging the agents $\mathcal{L}$ and $\mathcal{G}$ into one agent for logic and granularity critique.

5.3 Ablation Study

We conduct an ablation study with results shown in Table 5. There are two main conclusions:

All of the agents $\mathcal{K}$, $\mathcal{L}$, and $\mathcal{G}$ play positive roles. The data show that the complete RoadMapper outperforms all variants across all metrics, and removing any single agent or merging agents $\mathcal{L}$ and $\mathcal{G}$ results in varying degrees of performance degradation. For instance, excluding $\mathcal{K}$ leads to a decline of 5.84 in StepScore with an overall decrease of 3.72, while removing $\mathcal{L}$ leads to a decline of 5.43 on LogicScore. Meanwhile, the performance degradation on SC variant demonstrates the rationality of separating logic and granularity critiques.

DPO training plays a positive role, depending on the backbone model size. The data show that ablating DPO from agent $\mathcal{E}$ leads to an average performance degradation of 2.21. Meanwhile, when using Qwen3-14B as the backbone, the performance drops by 1.79. Notably, the Qwen3-8B backbone with DPO performs worse than the w/o DPO variant, possibly due to its inherent lack of capacity.

5.4 Efficiency and Experts Evaluation

Early Stopping Efficiency. We analyzed the relationship between iterations of the "critique-revise" and performance, and found that ***RoadMapper achieves an early stopping mechanism and thus balances performance and cost***. As shown in Figure 4, RoadMapper averages 1.64 and 1.51 iterations for Llama 3.3 70B and GPT-4o mini respectively. Compared to fewer iterations, increased iteration counts improve all metrics; compared to excessive iterations, early stopping maintains high performance while reducing computational costs, demonstrating the effectiveness of agent $\mathcal{E}$.

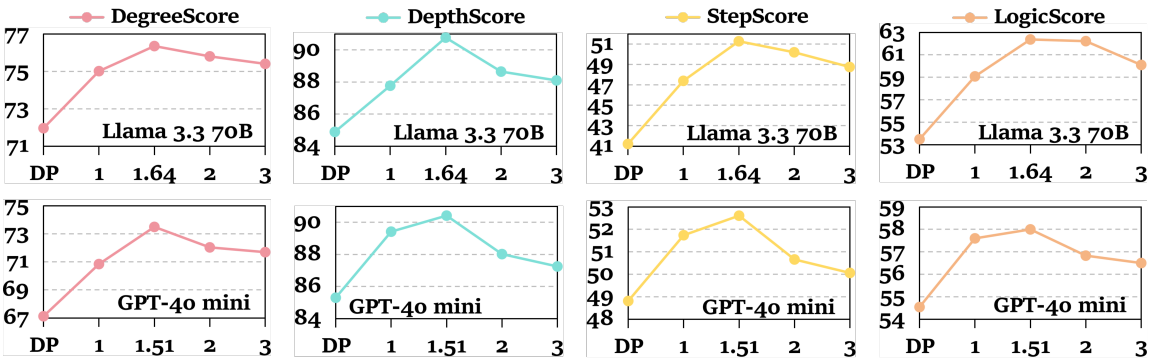

Figure 4: Quantitative analysis between enforced N iterations of "critique-revise" and performance. **DP** denotes Direct Prompting. The X-axis represents the rounds of "critique-revise" and the Y-axis shows the evaluation metrics.

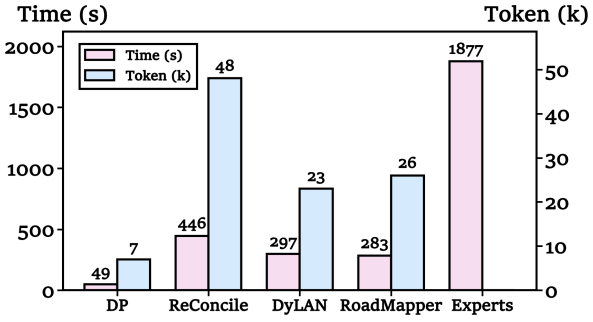

Figure 5: Cost comparison between methods on the roadmap generation task. **DP** denotes Direct Prompting.

Time and Token Efficiency. As shown in Figure 5, we comparatively analyzed the time and token consumption of different methods in roadmap design. In terms of time, all LLM-based automated methods significantly outperformed human experts, among which RoadMapper requires the least time in complex systems, saving 36.5% compared with ReConcile. In terms of token consumption, RoadMapper saved 45.8% compared with ReConcile. These results highlight the substantial potential of RoadMapper for practical applications.

Expert Evaluation. As shown in Appendix D.2, we instruct seven experts to evaluate the roadmaps across five dimensions: Logic Structure, Granularity, Topic Relevance, Completeness, and Clarity, adopting the pairwise comparison paradigm. The results show that (1) GPT-4o mini achieves a **93% matching rate** with experts evaluation, and (2) RoadMapper performs **better in 86% of cases**. These findings are consistent with the main results and validate the effectiveness of RoadMapper.

6 Related Work

Structured Content Generation. Structured content such as tables can help people quickly understand and memorize knowledge. For instance, Jain et al. (2024) generate table and mindmap summaries via specialized prompting; Li et al. (2023) construct text-to-table systems using coordinated text encoders and table generators; Ren et al. (2023) create program diagrams by analyzing text dependency relationships; Li et al. (2025b) implement engineering solution design through tree search-based Bi-point thinking. Nevertheless, no effective method exists for generating research roadmaps addressing complex research problems.

Multi-Agent Systems. LLMs may fail to adhere to multiple requirements when performing content generation. Researchers suggest composite systems like multi-agent systems (Amayuelas et al., 2024; Guo et al., 2024). Currently, relevant studies have explored applications in complex reasoning tasks: simulating game decision-making (Xu et al., 2024b), clinical assistance (Lu et al., 2024), and chart code generation (Li et al., 2025a). In contrast, our work investigates the application of multi-agent systems in the roadmap generation task.

7 Conclusion

We identify a contradiction between the importance of roadmaps and the lack of related research, and accordingly propose **RoadMap**, which evaluates the capabilities of LLMs in research roadmap generation tasks. Based on the evaluation results, we recognize three limitations of LLMs and propose **RoadMapper**, which accomplishes roadmap generation through coordinated work of multiple agents. Experiments demonstrate that RoadMapper can significantly address the challenges faced by LLMs and save much more time than experts require. We expect this work to advance the research of LLMs in the **roadmap generation task**.

Limitations

First, RoadMapper relies on LLMs, which means that although we have employed the currently best-performing prompts, the potential exists for more effective prompt designs to further improve model performance. Second, while RoadMapper incorporates an efficient early stopping mechanism, its computational cost remains relatively higher than that of direct prompting, highlighting opportunities for future optimization. Finally, due to limited computational resources, our experiments do not encompass all available models, particularly those that are prohibitively expensive.

Ethical Considerations

The development of RoadMapper complies with the ACL ethics guidelines. This study involved neither human subjects nor animal experimentation. All data were sourced from open-source repositories in strict adherence to their respective usage licenses, ensuring full privacy protection and the exclusion of any personally identifiable information. To support both commercial and open-source applications, RoadMapper is released under the Creative Commons Attribution 4.0 International License (CC BY 4.0), while the associated codebase is distributed under the Apache License 2.0. Furthermore, we have made consistent efforts to mitigate potential bias and maintain absolute transparency throughout the dataset construction and evaluation processes.

Use of AI Assistants

This research was conducted with the primary intellectual contributions and core scientific insights provided entirely by the authors. We acknowledge the use of AI-powered tools such as Cursor to assist in data processing, code writing, and text polishing. However, all key ideas, experimental design, analysis, and conclusions were formulated through human-driven reasoning and expertise. The use of AI did not influence the fundamental contributions or scientific integrity of this work.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. 62473271, 62176026), the Fundamental Research Funds for the Beijing University of Posts and Telecommunications (Grant No. 2025AI4S03), and the BUPT Innovation and Entrepreneurship Support Program (Grant No. 2025-YC-A033). This work is also supported by the Engineering Research Center of Information Networks, Ministry of Education, China. We would also like to thank the anonymous reviewers and area chairs for constructive discussions and feedback.

References

Alfonso Amayuelas, Xianjun Yang, Antonis Antoniades, Wenyue Hua, Liangming Pan, and William Yang Wang. 2024. MultiAgent collaboration attack: Investigating adversarial attacks in large language model collaborations via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6929–6948, Miami, Florida, USA. Association for Computational Linguistics.

Anthropic. 2024. Claude 3 Family.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *Preprint*, arXiv:2108.07732.

Dhaivat J. Bhatt, Seyed Ahmad Abdollahpouri Hosseini, Federico Fancellu, and Afsaneh Fazly. 2024. End-to-end parsing of procedural text into flow graphs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5833–5842, Torino, Italia. ELRA and ICCL.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Philip E Burian, Lynda Rogerson, and Francis R Maffei III. 2010. The research roadmap: A primer to the approach and process. *Contemporary Issues in Education Research*, 3(8):43–58.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, Bangkok, Thailand. Association for Computational Linguistics.

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

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *Preprint*, arXiv:2512.02556.

Zheye Deng, Chunkit Chan, Weiqi Wang, Yuxi Sun, Wei Fan, Tianshi Zheng, Yauwai Yim, and Yangqiu Song. 2024. Text-tuple-table: Towards information integration in text-to-table generation via global tuple extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9300–9322, Miami, Florida, USA. Association for Computational Linguistics.

Björn Engelmann, Christin Katharina Kreutz, Fabian Haak, and Philipp Schaer. 2024. ARTS: Assessing readability & text simplicity. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14925–14942, Miami, Florida, USA. Association for Computational Linguistics.

Gemini Team. 2025. Gemini 3.0: A new era of intelligence. Technical report, Google DeepMind.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8048–8057. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Fabian Haak and Philipp Schaer. 2025. Pairwise comparison for bias identification and quantification. *Preprint*, arXiv:2512.14565.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Mengting Hu, Honglei Guo, Shiwan Zhao, Hang Gao, and Zhong Su. 2021. Efficient mind-map generation via sequence-to-graph and reinforced graph refinement. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8130–8141, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Parag Jain, Andreea Marzoca, and Francesco Piccinno. 2024. STRUCTSUM generation for faster text comprehension. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7876–7896, Bangkok, Thailand. Association for Computational Linguistics.

Bingxuan Li, Yiwei Wang, Jiuxiang Gu, Kai-Wei Chang, and Nanyun Peng. 2025a. METAL: A multi-agent framework for chart generation with test-time scaling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30054–30069, Vienna, Austria. Association for Computational Linguistics.

Tong Li, Zhihao Wang, Liangying Shao, Xuling Zheng, Xiaoli Wang, and Jinsong Su. 2023. A sequence-to-sequence&set model for text-to-table generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5358–5370, Toronto, Canada. Association for Computational Linguistics.

Zhuoqun Li, Haiyang Yu, Xuanang Chen, Hongyu Lin, Yaojie Lu, Fei Huang, Xianpei Han, Yongbin Li, and Le Sun. 2025b. DeepSolution: Boosting complex engineering solution design via tree-based exploration and bi-point thinking. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4380–4396, Vienna, Austria. Association for Computational Linguistics.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. 2024. A dynamic llm-powered agent network for task-oriented agent collaboration. *Preprint*, arXiv:2310.02170.

Meng Lu, Brandon Ho, Dennis Ren, and Xuan Wang. 2024. TriageAgent: Towards better multi-agents collaborations for large language model-based clinical triage. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5747–5764, Miami, Florida, USA. Association for Computational Linguistics.

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. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland. Association for Computational Linguistics.

Meta AI. 2024a. Llama 3.1.

Meta AI. 2024b. Llama 3.3.

Meta AI. 2025. Llama 4.

MISTRALAI. 2025. Mistral Small 3.2: 24B Multimodal LLM with Enhanced Instruction Following.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao,

Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Chanjun Park and Hyeonwoo Kim. 2025. Understanding LLM development through longitudinal study: Insights from the open Ko-LLM leaderboard. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 1–8, Albuquerque, New Mexico. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3.

Haopeng Ren, Yushi Zeng, Yi Cai, Bihan Zhou, and Zetao Lian. 2023. Constructing procedural graphs with multiple dependency relations: A new dataset and baseline. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8474–8486, Toronto, Canada. Association for Computational Linguistics.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell L Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. 2024. Position: A roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 46280–46302. PMLR.

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.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. 2024a. Mineru: An open-source solution for precise document content extraction. *Preprint*, arXiv:2409.18839.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. 2024b. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5627–5646, Miami, Florida, USA. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed 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*, volume 35, pages 24824–24837. Curran Associates, Inc.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. Livebench: A challenging, contamination-limited llm benchmark. In *International Conference on Learning Representations*, volume 2025, pages 91595–91631.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024a. Large language models for generative information extraction: a survey. *Front. Comput. Sci.*, 18(6).

Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. 2024b. Language agents with reinforcement learning for strategic play in the werewolf game. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 55434–55464. PMLR.

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.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

A Details of RoadMap Benchmark

A.1 Distribution of Publication Years

As detailed in Section 3.1, we restricted our data collection to dissertations published since 2018. Figure 6 presents the distribution of publication years for the entire dataset. Notably, approximately 92.8% of the dissertations were published within the last five years, thereby ensuring the relevance and timeliness of the collected data.

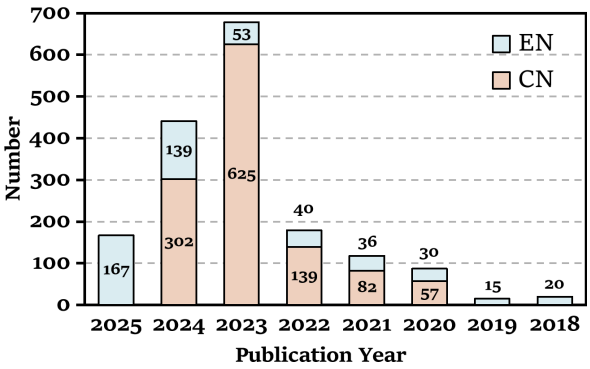

Figure 6: Distribution of the publication year of our collected dissertations.

A.2 Parsing PDF Dissertations into Markdown Format

We utilize MinerU[1] to parse the PDF files of the dissertations and extract Markdown results for subsequent processing. Here are some details: (1) Our GPU specification is NVIDIA A40, each equipped with 48GB VRAM; (2) We use version 2.0.6 of MinerU; (3) We employ `vlm-sglang-engine` as the backend; (4) The total GPU time used for parsing is approximately 40 hours.

A.3 Irrelevant Information Filtering in Markdown Files

We use Python scripts to filter the parsed Markdown files and remove irrelevant information, including author information, references, and invalid links. Key statistics regarding the filtering process are shown in Table 6. As described, this step removes approximately **37%** of irrelevant information from the dissertations, significantly reducing the computational cost for subsequent processing.

A.4 Template for Extraction

As detailed in Section 3.2, we manually formatted a template to extract core research problems and skill points from collected dissertations, based on `Gemini 2.5 Flash`. Figure 7 illustrates the complete content of this template.

Key Statistics Before and After Filtering			
English			
	Total	**Average**	**Reduction**
Before	210,390,046	420,780	✘
After	131,579,990	263,159	37.46%
Chinese			
	Total	**Average**	**Reduction**
Before	175,126,579	145,333	✘
After	110,374,242	91,596	36.97%

Table 6: Key statistics of characters before and after filtering the parsed markdown files.

A.5 Format Verification

As stated in Section 3.4, we have established strict format requirements for the roadmap content. We developed a Python automated script to assist in detecting any potential format errors, and any non-compliant nodes were manually corrected by experts. We categorized all format errors into three types: (1) ***node format non-compliance***, affecting **207** roadmaps; (2) ***level mismatch with index***, affecting **83** roadmaps; and (3) ***incorrect index relationships between nodes***, affecting **193** roadmaps. Since a single roadmap may contain multiple types of errors simultaneously, we ultimately performed format modifications on **329** roadmaps, accounting for approximately **19.3%** of the total.

A.6 Redundancy Removal

We implemented a comprehensive redundancy elimination pipeline based on semantic embeddings to identify and remove potentially redundant skill points. Specifically, each skill point was first encoded into an 8192-dimensional vector using the `Qwen3-Embedding-8B` model. Subsequently, pairwise cosine similarities between all skill points were computed using ChromaDB[2]. We heuristically flagged pairs with a cosine similarity exceeding $\tau = 0.6$ as potentially redundant and submitted them to expert review for arbitration and merging.

[1] https://mineru.net/

[2] https://docs.trychroma.com/

B Details of RoadMapper Methodology

B.1 $\mathcal{K}$ Agent

The $\mathcal{K}$ agent is responsible for knowledge augmentation of the initial roadmap by integrating both internal and external knowledge, operating in two sequential stages.

In the first stage, it generates internal knowledge using the prompt illustrated in Figure 8, while simultaneously retrieving relevant external skill points from the Skill-Repo. To enable retrieval, we first encode all skill points in the Skill-Repo into 8192-dimensional vectors using the `Qwen3-Embedding-8B` model, based on their associated problem descriptions and skill point names. During runtime, RoadMapper applies the same embedding method to compute the vector representation of the given complex research problem. We then employ the ChromaDB to retrieve the top-K ($K = 30$ in our experiments) most similar skill points according to cosine similarity, which constitute the external knowledge input for the $\mathcal{K}$ agent.

In the second stage, the agent performs knowledge augmentation operations by leveraging both the generated internal knowledge and the retrieved external knowledge, guided by the prompt shown in Figure 9.

B.2 $\mathcal{E}$ Agent

We require the $\mathcal{E}$ agent to objectively evaluate the quality of the roadmap along four dimensions (*i.e.*, Logic Structure, Granularity Degree, Topic Relevance, and Completeness) and output the evaluation scores and detailed analysis, with each evaluation score ranging from 0 to 100. The roadmap is only permitted to be output when the evaluation score exceeds a predefined *passing score* or when the maximum number of iterations is reached. We present the prompt of the $\mathcal{E}$ agent in Figure 10.

B.3 Details of DPO Training

We present key details of DPO training on $\mathcal{E}$ agent as follows:

Hardware and Software Environment.

- **GPU**: 4×NVIDIA RTX PRO 6000, each with 96 GB VRAM
- **CPU**: Intel Xeon Platinum 8470Q, 88 vCPUs
- **System memory**: 440 GB
- **OS**: Ubuntu 22.04
- **CUDA version**: 12.8
- **Framework**: PyTorch 2.8.0 + Python 3.12 + LlamaFactory 0.9.4.dev0 + vLLM 0.13.0

Hyperparameters.

- **`pref_beta`**: 0.2
- **`pref_loss`**: `sigmoid`
- **Maximum sequence length**: 4,096 (Qwen3-32B), 8,192 (Qwen3-8B/14B)
- **Global batch size**: 16
- **Epochs**: 2, with 10% warmup steps
- **Optimizer**: AdamW with learning rate = 5×10^{-5}, cosine decay, and weight decay
- **Precision**: BF16 mixed-precision training

Details of Constructing Preference Dataset. As described in Section 4.2, we prompt Qwen3-32B to evaluate roadmaps collected from RoadMap and its construction pipeline, and these evaluations constitute our candidate set. From this set, we select the candidates with the highest and second-highest vote counts through expert voting to form the positive and negative samples in the preference dataset, respectively. Meanwhile, we carefully observed that the model may generate evaluations that do not conform to the required format. To address this issue, we collected 36 model responses containing formatting errors. For each erroneous response, we called domain experts to manually correct the formatting. For every such error-correction pair, we constructed a preference pair by treating the corrected evaluation as the positive sample and the original malformed response as the negative sample, which were then incorporated into the preference dataset for training. Experiments show that after augmenting the training data with these format-oriented preference pairs, the model's formatting error rate **drops significantly from 8% to 3%**, demonstrating the effectiveness of our approach. Finally, our preference dataset comprises **818** preference pairs, with **36 format-related pairs**.

Training time. The required GPU hours for training the Qwen3-8B/14B/32B models are approximately 4/9/14, respectively.

C Details of Experiments

C.1 Experimental Environment

Base Environments. All our base experiments are conducted locally with the following details:

Model	Institution	Model Card
Open-Source Models		
Llama 3.1 8B	Meta	(Meta AI, 2024a)
Llama 3.3 70B	Meta	(Meta AI, 2024b)
Llama 4 Maverick	Meta	(Meta AI, 2025)
Qwen3-14B	Alibaba	(Qwen Team, 2025)
Qwen3-235B-A22B	Alibaba	(Qwen Team, 2025)
gpt-oss-20b	OpenAI	(OpenAI et al., 2025)
DeepSeek-V3.2	DeepSeek	(DeepSeek-AI et al., 2025)
Proprietary Models		
GPT-4o mini	OpenAI	(OpenAI et al., 2024)
Gemini 3 Flash Preview	Google	(Gemini Team, 2025)
Claude 3 Haiku	Anthropic	(Anthropic, 2024)
Mistral Small 3.2	Mistral AI	(MISTRALAI, 2025)

Table 7: Base models evaluated in main experiments.

- **OS**: Ubuntu 22.10
- **CPU**: Dual-socket Intel Xeon Gold 6148 (2.40 GHz), 40 cores per socket, 80 threads total
- **GPU**: 8×NVIDIA A40, each with 48 GB VRAM
- **GPU Driver**: 575.57.08
- **CUDA Version**: 11.8
- **cuDNN Version**: 8.x (compiled with CUDA 11.8)

Model Invocations. All model invocations are conducted via the OpenRouter[3] APIs with:

- Request frequency: 200 rpm
- Max retries: 10 times/item
- Temperature: 1.0 (Inference), 0.2 (Evaluation)

C.2 Base Models

Table 7 shows the list of base models used in the experiments, including both open-source and proprietary models.

C.3 Adaptability to Model Scale Variations

RoadMapper can significantly improve LLM performance across various model sizes, often enabling smaller models to outperform larger proprietary ones using conventional methods. Our findings highlight that RoadMapper's enhancements are not confined to large or top-tier models but extend effectively to smaller LLMs. For instance, Llama 3.1 8B with RoadMapper achieves an average score of **67.82**, marking a substantial 7.20 improvement over its Direct Prompting baseline (60.62). This enhanced performance notably surpasses the 66.43 average score of Claude 3 Haiku using Direct Prompting. This demonstrates that RoadMapper effectively democratizes high-quality roadmap generation, making advanced capabilities accessible to models with more limited capacities and showcasing its **efficacy and broad applicability regardless of model scale**.

[3] https://openrouter.ai/

D Other Important Details

D.1 Comparison of Evaluation Prompts in Different Formats

We observe that when employing GPT-4o mini as an evaluator to evaluate the content quality of roadmaps, different prompt formats significantly influence the model's performance. To quantitatively analyze this phenomenon and determine the optimal prompting strategy, we conduct a comparative experiment. Specifically, we first randomly select 100 roadmaps from the outputs generated by Llama 3.3 70B. Then, we employ GPT-4o mini to evaluate these roadmaps using different prompting strategies. Finally, we assess the performance of different prompting strategies using two metrics:

- **Consistency with Experts (CE)** measures the alignment between model evaluations and expert judgments. Specifically, we invite seven domain experts to vote for the best evaluation among all evaluations for each roadmap, and then calculate the vote rate for each prompting strategy.
- **Format Correctness (FC)** quantifies the reliability of structured output generation. We compute the probability that the model outputs conform to the required format across prompting strategies.

We test four prompting strategies, where each strategy's output format requirements include both JSON-style and XML-style formats:

- **AIH (All-In-Head)**: Both the reference roadmap and the evaluated roadmap are placed at the head of the prompt.
- **AIE (All-In-End)**: Both the reference roadmap and the evaluated roadmap are placed at the end of the prompt.
- **RHEE (Reference-Head-Evaluated-End)**: The reference roadmap is placed at the head of the prompt, while the evaluated roadmap is placed at the end of the prompt.
- **REEH (Reference-End-Evaluated-Head)**: The reference roadmap is placed at the end of the prompt, while the evaluated roadmap is placed at the head of the prompt.

Strategy	CE (%)	FC (%)	
		JSON	XML
AIH	34	86	91
AIE	30	87	89
RHEE	17	84	83
REEH	19	82	83

Table 8: Comparison of different prompting strategies for GPT-4o mini evaluation. **CE**: Consistency with Experts, **FC**: Format Correctness.

As described in Table 8, the AIH strategy achieves the best score on the CE metric. Moreover, the XML-style output format requirement obtains higher scores on the FC metric. Therefore, we adopt the AIH strategy as the prompting strategy for GPT-4o mini in our experiments, and require the model to output evaluation in XML format.

D.2 Consistency Analysis Between Model Evaluation and Expert Judgments

To evaluate the quality of roadmaps generated by different methods from the perspective of human experts and validate the reliability of using GPT-4o mini as an automated evaluator, we conduct a consistency analysis by adopting a pairwise comparison paradigm that aligns better with human cognitive processes (Zheng et al., 2023; Engelmann et al., 2024; Haak and Schaer, 2025).

Specifically, we first randomly collected $N_P = 100$ research problems from RoadMap and their corresponding roadmaps generated by both the Direct Prompting and RoadMapper methods, using `DeepSeek-V3.2` as the base model. Each research problem was paired with its respective roadmaps, forming N_P evaluation instances. Each evaluation instance is defined as a triplet:

$$E_i = (P_i, \{R_{D_i}, R_{M_i}\}). \tag{11}$$

Here, P_i denotes the i-th research problem, for $i \in \{1, 2, \ldots, N_P\}$, R_{D_i} denotes the roadmap generated by the **Direct Prompting** method for the research problem P_i, R_{M_i} denotes the roadmap generated by the **RoadMapper** method for the research problem P_i. It is important to note that the order of R_{D_i} and R_{M_i} within the set $\{R_{D_i}, R_{M_i}\}$ is randomized to ensure anonymity during the evaluation process.

During the human evaluation phase, we instructed $N_E = 7$ domain experts to serve as human evaluators. For each evaluation instance E_i, every expert was prompted to vote for the higher-quality roadmap across five dimensions: **Logic Structure**, **Granularity Degree**, **Topic Relevance**, **Completeness**, and **Clarity**, denoted collectively as the dimension set $\mathcal{D} = \{\text{LS}, \text{GD}, \text{TR}, \text{Co}, \text{Cl}\}$. The total vote count for each roadmap is computed as:

$$V(R_{D_i}) = \sum_{k=1}^{N_E} \sum_{j \in \mathcal{D}} v_j^{(k)}(R_{D_i}), \tag{12}$$

$$V(R_{M_i}) = \sum_{k=1}^{N_E} \sum_{j \in \mathcal{D}} v_j^{(k)}(R_{M_i}). \tag{13}$$

Here, $v_j^{(k)}(R_{D_i})$ and $v_j^{(k)}(R_{M_i})$ denote the binary vote assigned by the k-th expert on dimension j, where $v_j^{(k)}(R_{D_i}) + v_j^{(k)}(R_{M_i}) = 1$.

The final winner from experts is defined by:

$$\text{W}_i^{\text{experts}} = \begin{cases} R_{D_i}, & \text{if } V(R_{D_i}) > V(R_{M_i}) \\ R_{M_i}, & \text{otherwise} \end{cases}. \tag{14}$$

Here, W_i^{experts} denotes the superior roadmap in instance E_i as judged by the human experts.

On the model side, we derived the preference of GPT-4o mini by comparing the scalar scores it assigned to R_{D_i} and R_{M_i}, using the same evaluation settings introduced in Section 5.1. So, the final winner from GPT-4o mini is defined by:

$$\text{W}_i^{\text{evaluator}} = \begin{cases} R_{D_i}, & \text{if } S(R_{D_i}) > S(R_{M_i}) \\ R_{M_i}, & \text{otherwise} \end{cases}. \tag{15}$$

Here, $W_i^{\text{evaluator}}$ denotes the superior roadmap in instance E_i as judged by GPT-4o mini as an automated evaluator.

Finally, we calculate the matching rate between W_i^{experts} and $W_i^{\text{evaluator}}$, described by:

$$\text{MR} = \frac{\sum_{i=1}^{N_P} \mathbb{1}[W_i^{\text{experts}} = W_i^{\text{evaluator}}]}{N_P}. \tag{16}$$

Here, MR denotes the matching rate. The experimental results show that the MR between the model evaluation and expert evaluation reaches 93%, indicating that the scoring of GPT-4o mini is highly consistent with human judgment, supporting its use as a reliable evaluation proxy.

We further analyzed the distribution of expert evaluations, with results shown in Table 9. The roadmaps generated by RoadMapper were judged to be superior to those produced by Direct Prompting in **86%** of cases (i.e., $\sum_{i=1}^{N_P} \mathbb{1}[W_i^{\text{experts}} =$

Metric	LS	GD	TR	Co	Cl	Overall
Outcome	80	78	57	85	64	86

Table 9: Detailed human evaluation results across different dimensions.

$R_{M_i}] = 86$). Meanwhile, RoadMapper shows significant advantages over Direct Prompting on three fine-grained dimensions: **LS**, **GD**, and **Co**. These findings are highly consistent with the main experimental results (shown in Section 5.2) and further validate the effectiveness of the RoadMapper method in the research roadmap generation task.

D.3 Cross-Model Validation of Evaluators

To validate potential bias in GPT-4o mini as an automatic evaluator, we introduce Gemini 2.5 Flash and Qwen2.5 72B as alternative evaluators to conduct cross-model consistency validation against GPT-4o mini. The experiment is based on the same set of 100 research problem instances used in Appendix D.2, and comprises two aspects:

- **Correlation Analysis**: We computed the Spearman correlation coefficient between the scores assigned by Gemini 2.5 Flash and Qwen2.5 72B to the 100 roadmaps and those assigned by GPT-4o mini. Results show that the correlation coefficient between Gemini 2.5 Flash and GPT-4o mini is **0.812**, and that between Qwen2.5 72B and GPT-4o mini is **0.849**. All p-values are less than 0.001, indicating a high degree of alignment in scoring trends across models.
- **Match Consistency Analysis**: We followed the pairwise comparison paradigm from Section D.2: for each pair of roadmaps generated by Direct Prompting and RoadMapper, we determined the superior one based on the scores from Gemini 2.5 Flash or Qwen2.5 72B, and then compared this judgment with that of GPT-4o mini. We calculated the proportion of cases where both models agreed. Results indicate that Gemini 2.5 Flash matches GPT-4o mini in **90%** of cases, and Qwen2.5 72B matches in **93%** of cases, corroborating the robustness of the evaluation outcomes.

The cross-model validation results shown in Table 10 demonstrate that despite architectural differences, leading large language models exhibit high consistency in evaluating the quality of research roadmaps, ensuring the reliability of GPT-4o mini as an automated evaluation agent.

Model	Spearman R	P-Value	Match Rate (%)
Gemini 2.5 Flash	0.812	< 0.001	90
Qwen2.5 72B	0.849	< 0.001	93

Table 10: Cross-model validation results against GPT-4o mini for bias assessment.

Metric	Split	Pearson R	MAE	RMSE
StepScore	English	0.87	0.93	3.00
	Chinese	0.84	1.41	3.52
LogicScore	English	0.83	1.11	2.75
	Chinese	0.82	1.26	3.27

Table 11: Stability analysis of content evaluation metrics for Llama 3.3 70B. **Pearson R**: Pearson Correlation Coefficient, **MAE**: Mean Absolute Error, **RMSE**: Root Mean Squared Error.

D.4 Stability Analysis of Evaluation Results

We introduce our evaluation metrics in Section 5.1, as described, the content metrics (*i.e.*, **StepScore**, **LogicScore**) are content-based metrics that rely on the judgment of GPT-4o mini. So, it is crucial to demonstrate the stability of these metrics to ensure the validity of our experimental findings.

To assess the stability of the evaluation outcome from GPT-4o mini, we conduct repeated evaluations. Specifically, we still utilize GPT-4o mini as the evaluator to re-evaluate the experimental results generated by the Llama 3.3 70B model using the RoadMapper method, on both the English and Chinese splits. For these repeated evaluations, we compute the Pearson Correlation Coefficient, Mean Absolute Error, and Root Mean Squared Error between the scores from the two independent runs.

As shown in Table 11, it demonstrates high stability in the evaluation process. The Pearson correlation coefficients, ranging from 0.82 to 0.87 across all metrics and splits, indicate strong positive correlations between two independent runs. Furthermore, the low MAE values (all below 1.5 points) and RMSE values (all below 3.6 points) confirm that the absolute differences between repeated scores are minimal. Collectively, these findings validate the reliability and robustness of using GPT-4o mini as an evaluator for our content metrics.

D.5 Significance Analysis

To evaluate whether the performance gains of RoadMapper over Direct Prompting are statistically significant, we conduct paired two-sided t-tests on the outputs of gpt-oss-20b on the English split.

Metric	t	P-Value	95% CI	Significance
StepScore	-7.7261	< 0.001	[2.37, 3.98]	***
LogicScore	-3.9899	< 0.001	[0.65, 1.92]	***
DegreeScore	-6.8415	< 0.001	[3.77, 6.81]	***
DepthScore	-11.6542	< 0.001	[6.48, 9.11]	***

Table 12: Results of paired two-sided t-tests comparing RoadMapper vs. Direct Prompting on the English split. Significance levels: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$.

As shown in Table 12, RoadMapper achieves significant gains on all metrics with $p < 0.001$, and all 95% confidence intervals exclude zero. For instance, on `StepScore`, the mean improvement is 3.17 with a 95% CI of [2.37, 3.98], indicating that RoadMapper can generate roadmaps with better coverage of essential research steps. These results confirm that the performance gains are statistically robust across all evaluation dimensions.

D.6 Impact of DPO on General Knowledge Abilities

To investigate whether DPO adversely affects the model's general knowledge and reasoning capabilities, we conduct a controlled evaluation across four established benchmarks spanning mathematical reasoning, code generation, and step-by-step problem solving, using the backbone model Qwen3-32B and the DPO-optimized model:

- **GSM8K** (Cobbe et al., 2021): A dataset of 8,792 grade-school-level math word problems requiring multi-step arithmetic reasoning.
- **MATH-500** (Hendrycks et al., 2021): A curated subset of 500 challenging high-school math problems from the MATH dataset, covering algebra, geometry, combinatorics, and calculus.
- **MBPP** (Austin et al., 2021): The Most Basic Python Problems benchmark with 427 entry-level coding tasks evaluated by exact function match (pass@1).
- **LiveBench** (White et al., 2025): A dynamic benchmark designed to test real-world generalization. We evaluate on the reasoning split.

As shown in Table 13, the model maintains strong performance across all tasks after DPO. On GSM8K, there is a minor drop from 79.83% to 78.11%. Similarly, MATH-500 sees a slight decrease from 88.40% to 87.80%. These small regressions may stem from subtle shifts in decoding behavior or over-normalization of reasoning paths during preference learning. Notably, the model improves on MBPP (from 69.56% to 71.90%) and LiveBench (from 83.50% to 85.50%), indicating that **the well-designed preference data not only enforces compliance but can also reinforce coherent and effective problem-solving strategies**.

Benchmark	Backbone (%)	DPO (%)	Δ (%)
GSM8K	79.83	78.11	-1.72
MATH-500	88.40	87.80	-0.60
MBPP	69.56	71.90	+2.34
LiveBench	83.50	85.50	+2.00

Table 13: Performance comparison between the backbone model and the DPO-finetuned model on general knowledge and reasoning benchmarks. The Δ column shows the absolute difference (DPO $-$ Backbone), indicating high overall stability across all tasks.

D.7 Case Study

Figure 11 presents a simplified yet complete case study to illustrate the operational process of RoadMapper. After the research problem is input, the Init Agent generates an initial roadmap. Subsequently, the Knowledge Agent inserts an `[Eval Metrics]` node. The process then enters the critique phase, where the Revise Agent adjusts the node order and decomposes tasks according to suggestions. The revised roadmap is evaluated by the Evaluator. Since it does not meet the passing score, the output is rejected, and the process proceeds to the next round of critique. The Revise Agent again adjusts the node order and merges subtasks based on the improvement suggestions. The subsequently revised roadmap passes the Evaluator's evaluation and is successfully output, ultimately yielding the final roadmap. This case study perfectly demonstrates the initial generation, knowledge augmentation, and iterative "critique-revise-evaluate" process of the RoadMapper system.

D.8 Example of Golden-Roadmap

We present a Golden Roadmap in Figure 12, where nodes marked in pink font represent key nodes annotated by experts.

Template for Extracting Core Research Problems and Skill Points from Dissertations.

```
You are a research assistant specialized in extracting key information from
     academic literature. Based on the provided dissertation, your task is
    to identify and extract core research problem and skill points. The
    requirements are:
1. Extraction must be based strictly on the content of the provided paper.
2. Extract at most five of the most innovative and valuable skill points.
3. Extract the core research task of the entire paper.
4. Answer in JSON format and follow the template provided below.
5. Enclose the extracted results with ```json```.
6. Answer in English.
Template:
{
  "core_research_question": ""
    # The core research question in a concise question format like "How to
        ...?" (no more than 30 words).
    "skill_points": [
    {
    "problem_description": "", # Describe the specific problem this skill
        point addresses.
    "skill_point_name": "", # Concise name of the skill point.
    "skill_point_description": "" # Detailed explanation of the skill point
        content.
    },
    # Other skill points...
  ]
}
```

Figure 7: Template for extracting core research problems and skill points from dissertations.

Prompt of the $\mathcal{K}$ Agent for Generation of Internal Knowledge

```
You are an experienced research expert, specialized in analyzing research
    problems and identifying key skills.
For a given research problem provided by the user, you need to analyze and
    output several key skills that are essential for solving the problem.

Please strictly follow the requirements below:
1. Analyze the research problem and output several key skills that are
    essential for solving the problem.
2. Output format requirements:
    - Use JSON list format for output, with each item being a skill point
        including name and description fields.
    - `name`: The name of the skill point (brief description).
    - `description`: The detailed explanation of the skill point (explains
        the role of the skill point in solving the problem).
    - Enclose the output in ```json```.
3. The skills should be answered in {split_language}.

Example output format:
```json
[
    {{"name": "xxx", "description": "xxx"}},
    {{"name": "xxx", "description": "xxx"}},
    ...
]
```
```

Figure 8: Prompt of the $\mathcal{K}$ agent for generation of internal knowledge.

Prompt of the $\mathcal{K}$ Agent for Knowledge Augmentation

```
You are an experienced research expert, specialized in optimizing research
    problem roadmaps based on skill point repositories.
For a research problem, a current roadmap (in Markdown format), and a skill
    point repository provided by the user, you need to optimize the roadmap
    based on the research problem, the current roadmap, and the skill point
    repository.

Please strictly follow the requirements below:
1. Carefully analyze each skill point in the skill point repository (each
    skill point includes its name and description) according to the research
    problem and the current roadmap.
2. Determine which skill points are helpful for improving the roadmap, insert
     the names of helpful skill points as a step node into appropriate
    positions in the roadmap (names can be adapted to the problem as needed),
     and ignore unhelpful skill points.
3. Ensure the correct format of the roadmap during insertion, maintaining the
     logical order and continuity of node indices.
4. Roadmap format requirements, make sure to strictly follow:
    - Use Markdown format for output, with each line as a node (including
        level, index, and title, separated by spaces).
    - Use different heading levels (#, ##, ###, etc.) to indicate the node
        level and the hierarchical structure of the roadmap.
    - Use indices like 1.1.1 to indicate the node's position in the roadmap.
    - The title should use the format [xxx], where xxx is the node content.
5. Enclose the output in ```markdown```.
6. The roadmap content should be answered in {split_language}.

Example output format:
```markdown
# 1 [Main Step]
## 1.1 [Sub-step]
## 1.2 [Sub-step]
# 2 [Main Step]
## 2.1 [Sub-step]
### 2.1.1 [Sub-step]
### 2.1.2 [Sub-step]
...
```
```

Figure 9: Prompt of the $\mathcal{K}$ agent for knowledge augmentation.

Prompt of the $\mathcal{E}$ Agent

```
You are an experienced research expert, specialized in evaluating the quality
    of research roadmaps.
For a research problem and a proposed roadmap (in Markdown format) used to
    solve the research problem, you need to evaluate the roadmap from
    multiple dimensions below and provide objective and fair scores and
    detailed analysis:
1. Logic Structure: Evaluate the logical coherence of the roadmap, including
    whether the dependencies between nodes are reasonable, whether the step
    order is logical, and whether there are conflicts or contradictions
    between different parts.
2. Granularity Degree: Evaluate the rationality of task decomposition
    granularity, including whether there are too macroscopic or too
    microscopic nodes, whether the sub-task division is balanced, etc.
3. Topic Relevance: Evaluate the relevance of the roadmap to the input
    research problem, including whether the roadmap fully covers the core
    content required to solve the problem, whether there are redundant nodes
    unrelated to the topic, and whether the key technical points are fully
    reflected, etc.
4. Completeness: Evaluate the richness and completeness of the roadmap
    content, including whether there are missing key nodes or steps, and
    whether all the content required to solve the problem is fully covered.

Based on the four dimensions above, provide an overall assessment using the
    following 100-point scale (percentage system, 0-100):
0-19: Very poor quality, cannot be used.
20-39: Poor quality, needs significant improvement and is not recommended as
    is.
40-59: Acceptable but still needs improvement before use.
60-79: Good quality and can be used with minor adjustments.
80-100: Excellent quality that are perfectly perfect in all four dimensions
    and can be directly used.

Output Format Requirements:
You must answer in English and enclose your score within <eval_score> tags
    and your detailed reasoning within <eval_reason> tags. For instance:

<eval_score>68</eval_score>
<eval_reason>Your detailed analysis covering all four dimensions...</
    eval_reason>

Please strictly follow the requirements above and provide the output in the
    specified format.
```

Figure 10: Prompt of the $\mathcal{E}$ agent.

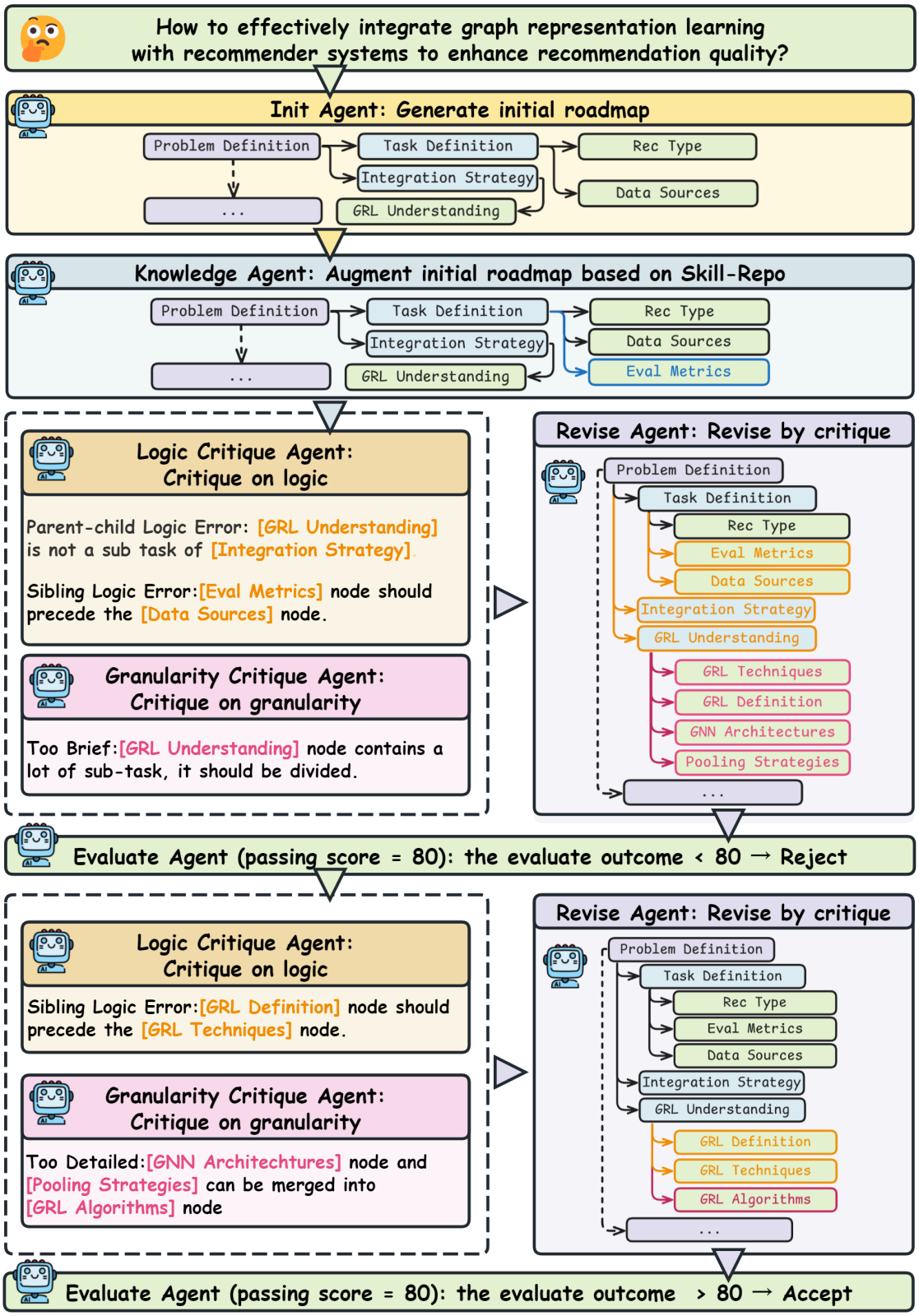

Figure 11: Case Study of RoadMapper.

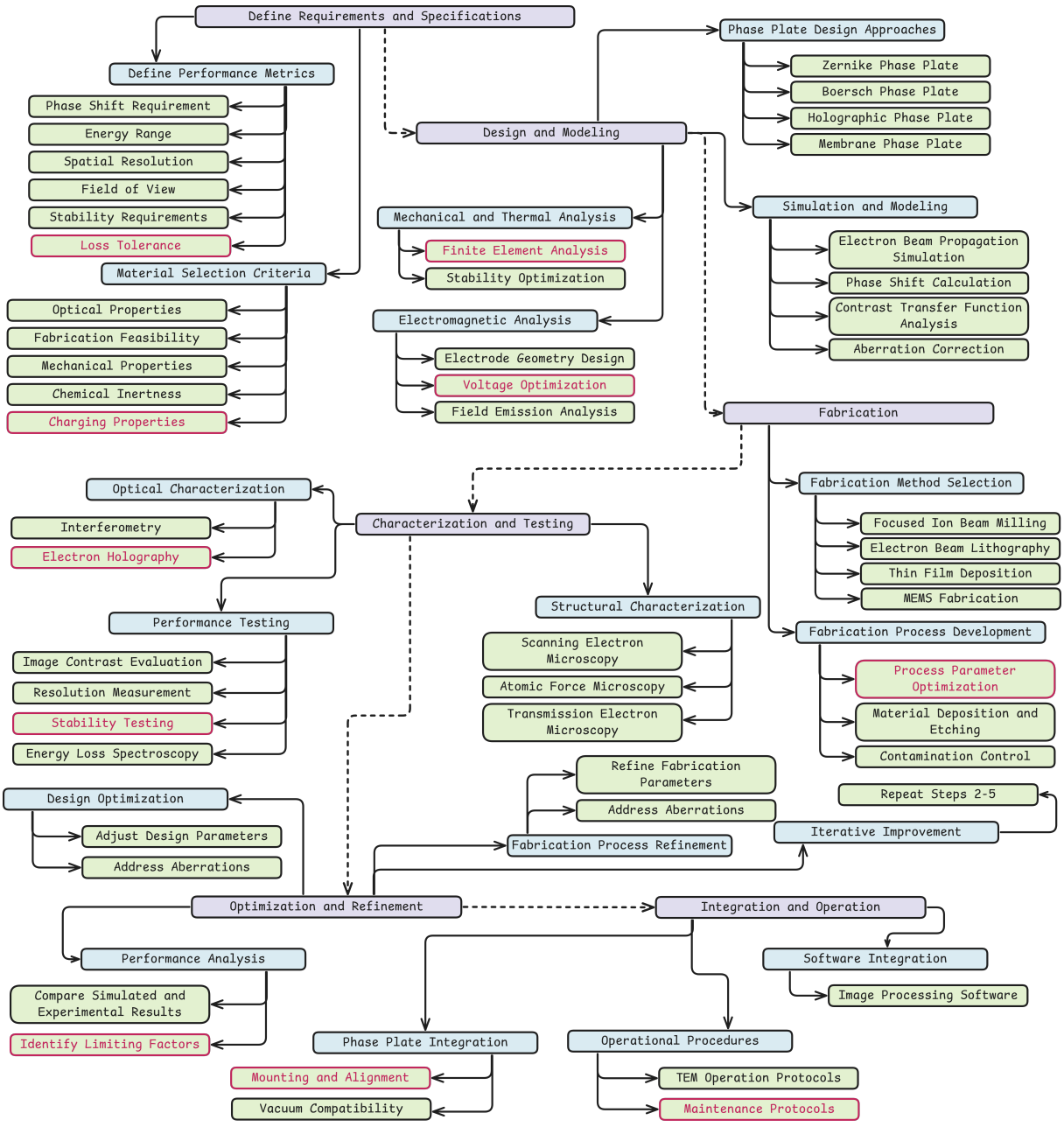

Figure 12: Example of Golden-Roadmap.