



PDF Download
3746027.3754813.pdf
05 January 2026
Total Citations: 0
Total Downloads: 60

Latest updates: <https://dl.acm.org/doi/10.1145/3746027.3754813>

RESEARCH-ARTICLE

ResearchPulse: Building Method-Experiment Chains through Multi-Document Scientific Inference

QI CHEN, University of Chinese Academy of Sciences, Beijing, China

JINGXUAN WEI, University of Chinese Academy of Sciences, Beijing, China

ZHUOYA YAO, University of Chinese Academy of Sciences, Beijing, China

HAIGUANG WANG, University of Chinese Academy of Sciences, Beijing, China

GAOWEI WU, University of Chinese Academy of Sciences, Beijing, China

BIHUI YU, University of Chinese Academy of Sciences, Beijing, China

[View all](#)

Open Access Support provided by:

Shanghai Artificial Intelligence Laboratory

University of Chinese Academy of Sciences

Zhejiang University

Published: 27 October 2025

[Citation in BibTeX format](#)

MM '25: The 33rd ACM International
Conference on Multimedia
October 27 - 31, 2025
Dublin, Ireland

Conference Sponsors:
SIGMM

ResearchPulse: Building Method–Experiment Chains through Multi-Document Scientific Inference

Qi Chen*
University of Chinese Academy of
Sciences
Beijing, China

Jingxuan Wei*[†]
University of Chinese Academy of
Sciences
Beijing, China
weijingxuan20@mails.ucas.edu.cn

Zhuoya Yao
University of Chinese Academy of
Sciences
Beijing, China

Haiguang Wang
University of Chinese Academy of
Sciences
Beijing, China

Gaowei Wu
University of Chinese Academy of
Sciences
Beijing, China

Bihui Yu
University of Chinese Academy of
Sciences
Beijing, China

Siyuan Li
Zhejiang University
Hangzhou, China

Cheng Tan[†]
Shanghai Artificial Intelligence
Laboratory
Shanghai, China
chengtan9907@gmail.com

Abstract

Understanding how scientific ideas evolve requires more than summarizing individual papers—it demands structured, cross-document reasoning over thematically related research. In this work, we formalize multi-document scientific inference, a new task that extracts and aligns motivation, methodology, and experimental results across related papers to reconstruct research development chains. This task introduces key challenges, including temporally aligning loosely structured methods and standardizing heterogeneous experimental tables. We present **ResearchPulse**, an agent-based framework that integrates instruction planning, scientific content extraction, and structured visualization. It consists of three coordinated agents: a Plan Agent for task decomposition, a Mmap-Agent that constructs motivation–method mind maps, and a Lchart-Agent that synthesizes experimental line charts. To support this task, we introduce **ResearchPulse-Bench**, a citation-aware benchmark of annotated paper clusters. Experiments show that our system, despite using 7B-scale agents, consistently outperforms strong baselines like GPT-4o in semantic alignment, structural consistency, and visual fidelity. The dataset are available in <https://huggingface.co/datasets/ResearchPulse/ResearchPulse-Bench>

CCS Concepts

• **Applied computing** → **Document management and text processing**; • **Computing methodologies** → *Artificial intelligence*.

*Both authors contributed equally to this research.

[†]Corresponding author



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

MM '25, Dublin, Ireland

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2035-2/2025/10

<https://doi.org/10.1145/3746027.3754813>

Keywords

Multi-document scientific inference, agent-based research analysis, method-tracking, experimental result alignment, research trajectory visualization

ACM Reference Format:

Qi Chen, Jingxuan Wei, Zhuoya Yao, Haiguang Wang, Gaowei Wu, Bihui Yu, Siyuan Li, and Cheng Tan. 2025. ResearchPulse: Building Method–Experiment Chains through Multi-Document Scientific Inference. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25)*, October 27–31, 2025, Dublin, Ireland. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3746027.3754813>

1 Introduction

Scientific research is cumulative and comparative. To truly understand how a field evolves, researchers often need to examine multiple thematically related papers together [1–3], analyze their motivations and methodological innovations, and track how experimental performance changes over time [4–6]. This type of structured, cross-document analysis is essential for identifying trends, gaps, and breakthroughs in fast-developing domains like artificial intelligence. However, current summarization models [7–10] and automatic survey generators [11–14] focus either on compressing individual papers or generating high-level overviews, lacking the granularity needed to uncover methodological trajectories and benchmark shifts. Such limitations hinder the ability to track how cutting-edge techniques evolve, converge, or diverge across papers over time. Meanwhile, emerging deep research agents [15–20] offer promising automation but are either domain-agnostic or lack structural precision.

We introduce a new task, multi-document scientific inference, which involves extracting and aligning key research elements—such as motivation, methodology, and experimental results—across thematically related papers to reconstruct structured development chains. As illustrated in Figure 1, this task differs fundamentally

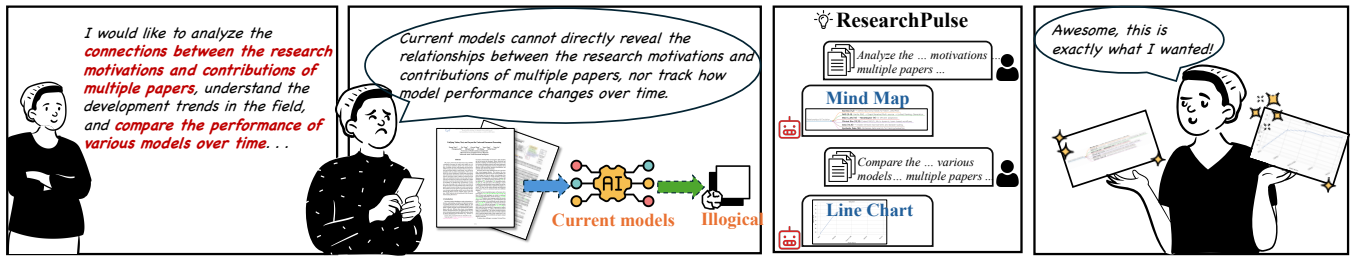


Figure 1: ResearchPulse transforms scientific papers into mind maps and line charts for structured research tracking.

from traditional summarization or survey generation, as it requires not only semantic abstraction but also structural parsing and scientific reasoning across documents. This process introduces several core challenges: (1) **identifying and temporally aligning motivation–method pairs from diverse, loosely structured textual sources**, and (2) **extracting heterogeneous experimental tables and converting them into unified, interpretable visual trends**. These challenges make multi-document scientific inference a structurally grounded and underexplored problem, distinct from prior efforts in academic summarization or auto-survey pipelines.

To support this new task, we introduce ResearchPulse-Bench, a benchmark dataset constructed from citation-aware paper clusters curated from *arXiv* and *OpenReview*. Each cluster contains semantically related papers aligned by topic and publication timeline. We annotate core scientific elements—motivations, methods, experimental tables, evaluation metrics, and citation links—using model-assisted extraction techniques, followed by rigorous human validation to ensure structural integrity and factual correctness. This enables high-quality supervision for both Method-Tracking and Experimental-Analysis tasks, supporting model training and standardized evaluation.

Building on this benchmark, we propose ResearchPulse, an end-to-end agent system for structured scientific inference. The system begins with a *Plan Agent* that interprets user instructions and coordinates the workflow. For method-tracking, a *Mmap-Agent* extracts motivation–method pairs from related papers and organizes them into temporally aligned mind maps. For experimental-analysis, a *Lchart-Agent* identifies experimental tables, model names, and evaluation metrics, and synthesizes benchmark trajectories as line charts. Each component operates in a modular yet coordinated manner, enabling ResearchPulse to transform loosely connected papers into coherent, interpretable research chains and support dynamic tracking of scientific progress.

Our contributions are as follows:

- We formally define the task of multi-document scientific inference, which focuses on extracting and aligning motivation, method, and experiment elements from thematically related papers to reconstruct structured research development chains.
- We introduce **ResearchPulse-Bench**, a citation-aware benchmark dataset consisting of annotated document clusters and reference outputs for method-tracking and experimental-analysis.

- We propose **ResearchPulse**, a modular agent system that integrates instruction planning, scientific content extraction, and structure-aware visualization into a unified pipeline for tracking methodological evolution and experimental trends in AI research.

2 Related Work

2.1 Scientific Document Summarization

Scientific document summarization aims to condense long and structurally complex academic texts while preserving core semantic content. Early approaches like TextRank [21] and LexRank [22] rely on surface-level lexical features and often fail to capture discourse-level dependencies [23]. Neural and transformer-based models such as BERTSum [7] and LoBART [8] introduce deep semantic modeling but remain limited by context length and generation inconsistencies. Later methods like HEGEL [9] and HAESum [10] improve global-local relation modeling through hypergraph and hierarchical attention, though they often depend on noisy tools like LDA or KeyBERT [24]. Structure-aware models—such as dependency-based discourse parsers [25] and sentence compression with anaphora constraints [26]—seek better coherence at the cost of increased system complexity. Document expansion [27] introduces external context to improve coverage but risks topic drift. Models like BoookScore [28] and top-down inference frameworks [29] target book-length summarization but still struggle with scientific logic modeling. Overall, while single-document methods evolve in modeling semantics and structure, they often overlook the underlying scientific logic—motivation, method, experiment—and lack mechanisms for linking content across documents.

Multi-document summarization requires integrating redundant or conflicting information from multiple sources. Hybrid frameworks such as Hi-MAP [30], SKT5SciSumm [31], and REFLECT [32] combine extraction with generation but often flatten document-specific context. Graph-based systems like CeRA [33] and cross-document information graphs [34] offer improved factual grounding but mainly focus on entity-event relations. Other methods highlight diversity or specificity—e.g., DisentangleSum [35] and DIVERSESUMM [36]—while large-scale pipelines like GPT-based summarizers [37] use recursive generation and clustering to scale. Despite these advances, few approaches address the extraction or alignment of structured scientific reasoning. Evaluation studies confirm this limitation, showing that many benchmarks fail to assess true cross-document synthesis [35, 36]. These gaps underscore the

need for systems that can extract, align, and reason over structured scientific content across thematically related works.

2.2 Automatic Survey Generation

Automatic survey generation seeks to streamline the labor-intensive process of writing literature surveys by leveraging large language models (LLMs). Traditional multi-document summarization methods, designed for small input sets and superficial summaries, fall short for this task. Recent works propose more targeted pipelines. For instance, BigSurvey introduces a large-scale dataset and the CAST model, combining sentence classification and sparse-transformer-based generation for structured and abstracted summarization [11]. However, such outputs often lack critical synthesis. To enhance organization, CHIME uses hierarchical generation with LLMs to build topic trees and incorporates expert feedback for structural refinement [12]. AutoSurvey adopts a four-stage pipeline—retrieval, outline generation, section-wise writing, and optimization—where multiple LLMs co-generate and critique survey drafts, improving efficiency but still facing depth and grounding issues [13]. SurveyX further refines this process through attribute-based preprocessing and information templates, generating attribute forests and novel evaluation metrics for factual consistency and coverage [14]. Other systems like STORM and ChatCite focus on improving cognitive scaffolding via multi-perspective questioning and human-in-the-loop comparison workflows [38, 39]. While these approaches mark significant progress in scaling academic synthesis via LLMs, they primarily focus on high-level thematic abstraction, often overlooking fine-grained research trajectories such as methodological evolution or experimental performance shifts. In contrast, our work aims to uncover structural research progressions by aligning methodology and experimental results across thematically linked papers within a specific research direction—thereby enabling more granular knowledge discovery than generic survey generation pipelines.

2.3 Automated Deep Research Systems

Recent advancements in large language models (LLMs) have enabled deep research systems that go beyond summarization to support iterative scientific inquiry through retrieval, reasoning, and hypothesis generation. For example, DOLPHIN[15] introduces a closed-loop framework combining idea generation, experimental validation, and feedback refinement, with features like literature retrieval and error-aware debugging. ResearchAgent[16] leverages citation graphs and cross-domain knowledge to enhance idea generation, and uses multi-agent LLM-based peer review for evaluating novelty and clarity. Other systems focus on targeted capabilities such as information seeking. DeepSeek-R1[40] and R1-Searcher[41] apply reinforcement learning to prompt external search when knowledge gaps arise, using reward-driven prompting and retrieval-aware rollouts to improve factual grounding. Broader agentic reasoning frameworks [17] coordinate multiple agents—like memory agents, search agents, and code executors—to support multi-hop inference, task decomposition, and synthesis across modalities. Commercial platforms have also adopted this paradigm. OpenAI[42] and Gemini[43] embed tool use, asynchronous planning, and adaptive web exploration into general-purpose research agents. However,

these systems often lack domain-specific precision for structured scientific analysis.

In contrast, our work introduces a focused agent system for structural knowledge discovery within a specific research domain. Rather than generating broad research plans, our system extracts and aligns motivation, methods, and results across related papers—capturing how ideas evolve, techniques recur, and performance shifts over time. This enables fine-grained, structured insights that bridge the gap between general AI agents and the rigorous demands of scientific research.

3 Method

3.1 Task Definition

We define the objective of **ResearchPulse** as an agent-based system for scientific document understanding, designed to analyze and organize structural research progress through two complementary tasks: (1) *Method-Tracking*, and (2) *Experimental-Analysis*. Each task enables fine-grained scientific inference over related papers via multi-agent coordination and tool-assisted document processing.

Let $C = \{D_1, D_2, \dots, D_N\}$ denote a collection of N related scientific documents, typically belonging to the same research thread or citation lineage. Each document D_i contains multiple sections including introduction, methodology, experiments, and references.

Method-Tracking Task. Given a document set C and a user instruction $\mathcal{I}_{\text{method}}$, the goal is to extract for each D_i its core motivation m_i and methodological description r_i from the abstract and introduction. The system then temporally aligns the extracted tuples $\{(m_i, r_i)\}_{i=1}^N$ based on the publication timestamp t_i of each D_i , and generates a structured representation $\mathcal{M}_{\text{chain}}$ in a hierarchical markdown format. This content is further processed into a mind-map–style visualization.

Experimental-Analysis Task. Given the same input document set C and a user instruction \mathcal{I}_{exp} , the goal is to extract from each D_i its main experimental table T_i , associated model names \mathcal{M}_i , evaluation metrics \mathcal{E}_i , and cited baseline years \mathcal{Y}_i . The system organizes the metric values over time and produces a structured summary $\mathcal{E}_{\text{chain}}$ reflecting comparative performance trends. The output is rendered as a line chart via auto-generated Python code.

3.2 ResearchPulse

ResearchPulse is implemented as a modular, instruction-driven agent system designed to perform structured scientific inference across related documents. Given a document set $C = \{D_1, D_2, \dots, D_N\}$ and a user-issued instruction \mathcal{I} , the system decomposes the task into two coordinated subtasks—Method Tracking and Experimental Analysis—each handled by a specialized sub-agent. The overall pipeline is illustrated in Figure 2.

3.2.1 Plan Agent. The *Plan Agent* acts as the central controller of ResearchPulse. It receives user instruction \mathcal{I} and identifies the corresponding intent category: $\mathcal{I}_{\text{method}}$ for method-tracking or \mathcal{I}_{exp} for experimental comparison. Based on this classification, it dynamically dispatches subtasks to the appropriate downstream agent (Mmap-Agent or Lchart-Agent).

To support end-to-end automation, the Plan Agent invokes an Extraction module, which parses each document D_i in C into its

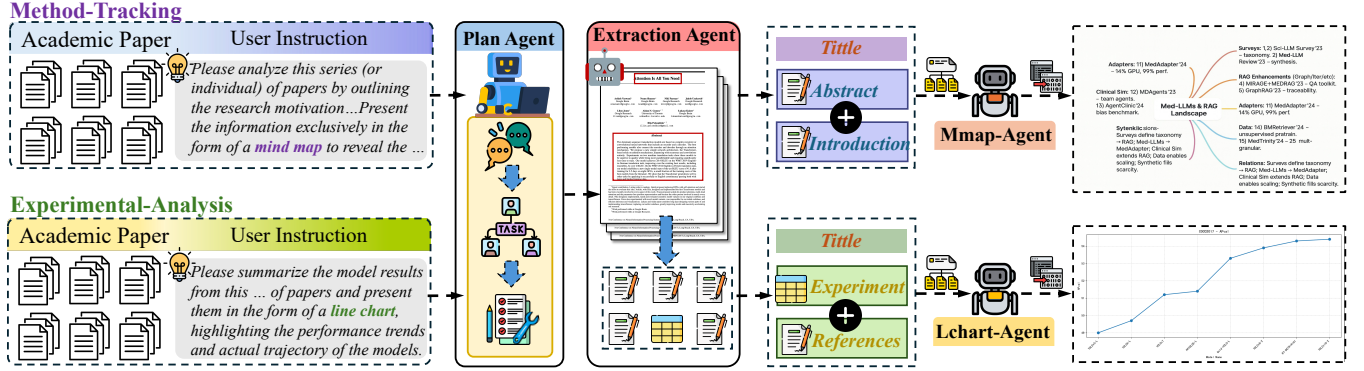


Figure 2: The ResearchPulse pipeline, consisting of three main agents: Plan Agent, Mmap-Agent, and Lchart-Agent.

structured components—including abstract, introduction, experiment tables, and references—producing normalized content X_i :

$$X_i = \text{Extract}(D_i), \quad \forall i \in \{1, \dots, N\}. \quad (1)$$

The agent then routes each X_i with the appropriate instruction $\mathcal{I}_{\text{method}}$ or \mathcal{I}_{exp} to the respective downstream processing agent. This decouples instruction planning from content processing, enabling modular coordination across diverse scientific tasks.

3.2.2 Mmap-Agent. The Mmap-Agent is responsible for identifying the research motivation and methodology for each paper in the corpus and constructing a time-aligned representation of scientific evolution. Specifically, for each X_i , the agent extracts a tuple (m_i, r_i) representing the motivation and method, respectively. These are typically sourced from the abstract and introduction sections.

Formally, this agent performs:

$$(m_i, r_i) = \mathcal{F}_{\text{Mmap}}(X_i, \mathcal{I}_{\text{method}}), \quad (2)$$

where $\mathcal{F}_{\text{Mmap}}$ is a fine-tuned LLM designed for structural scientific information extraction. The output for all documents is then temporally sorted by publication timestamp t_i to produce a markdown-formatted research chain:

$$\mathcal{M}_{\text{chain}} = \text{Sort}_{t_i} \left(\{(m_i, r_i)\}_{i=1}^N \right), \quad (3)$$

which is subsequently rendered into a mind map visualization.

To supervise this extraction process, we optimize the negative log-likelihood of the predicted motivation-method tuple sequence:

$$\mathcal{L}_{\text{Mmap}} = - \sum_{i=1}^N \log P((m_i, r_i) | X_i, \mathcal{I}_{\text{method}}), \quad (4)$$

where $P(\cdot)$ denotes the conditional probability predicted by the fine-tuned LLM. The training data includes manually annotated pairs of motivations and methods across representative paper series. The loss encourages the agent to capture salient reasoning structures and align them chronologically for subsequent visualization.

3.2.3 Lchart-Agent. The Lchart-Agent is designed to extract experimental results from each paper and generate visual representations of comparative performance. For each document X_i , the agent identifies its main experimental table T_i , model names \mathcal{M}_i , evaluation metrics \mathcal{E}_i , and baseline publication years \mathcal{Y}_i linked via citation resolution.

Formally, the agent conducts:

$$(T_i, \mathcal{M}_i, \mathcal{E}_i, \mathcal{Y}_i) = \mathcal{F}_{\text{Lchart}}(X_i, \mathcal{I}_{\text{exp}}), \quad (5)$$

where $\mathcal{F}_{\text{Lchart}}$ is a separately fine-tuned agent optimized for tabular extraction and metric normalization. The extracted results are aligned across time and datasets to construct a structured trend record:

$$\mathcal{E}_{\text{chain}} = \text{Align} \left(\{(T_i, \mathcal{M}_i, \mathcal{E}_i, \mathcal{Y}_i)\}_{i=1}^N \right), \quad (6)$$

which is visualized as a line chart through Python code automatically generated by the agent:

$$\text{Chart} = \mathcal{V}(\mathcal{E}_{\text{chain}}). \quad (7)$$

The Lchart-Agent is trained to extract structured experimental content with high precision. Its objective is to maximize the correctness and completeness of extracted tuples $(T_i, \mathcal{M}_i, \mathcal{E}_i, \mathcal{Y}_i)$ under supervision. We define the training loss as:

$$\mathcal{L}_{\text{Lchart}} = - \sum_{i=1}^N \log P(T_i, \mathcal{M}_i, \mathcal{E}_i, \mathcal{Y}_i | X_i, \mathcal{I}_{\text{exp}}), \quad (8)$$

where the probability is computed over multi-field outputs. The agent is fine-tuned using a curated dataset of aligned tables and metric annotations, encouraging accurate parsing of experimental results and consistent temporal alignment across documents.

4 ResearchPulse-Bench

We present **ResearchPulse-Bench**, a benchmark designed to support multi-document scientific inference tasks including method-tracking and experimental comparison. Unlike traditional summarization datasets, ResearchPulse-Bench is constructed from real-world citation networks and explicitly captures the structure of research progressions. Each instance contains a series of related scientific papers organized by semantic and temporal proximity, annotated with motivation, methodology, and experimental evidence. The pipeline includes data collection, document parsing, information extraction, and visualization, as shown in Figure 3.

Data Collection. We begin by retrieving a set of seed papers from Google Scholar based on AI-related keywords (e.g., “Machine Learning”, “Computer Vision”, “Natural Language Processing”), citation counts, and publication timeframes (past five years). To ensure data openness, we restrict our collection to papers that are

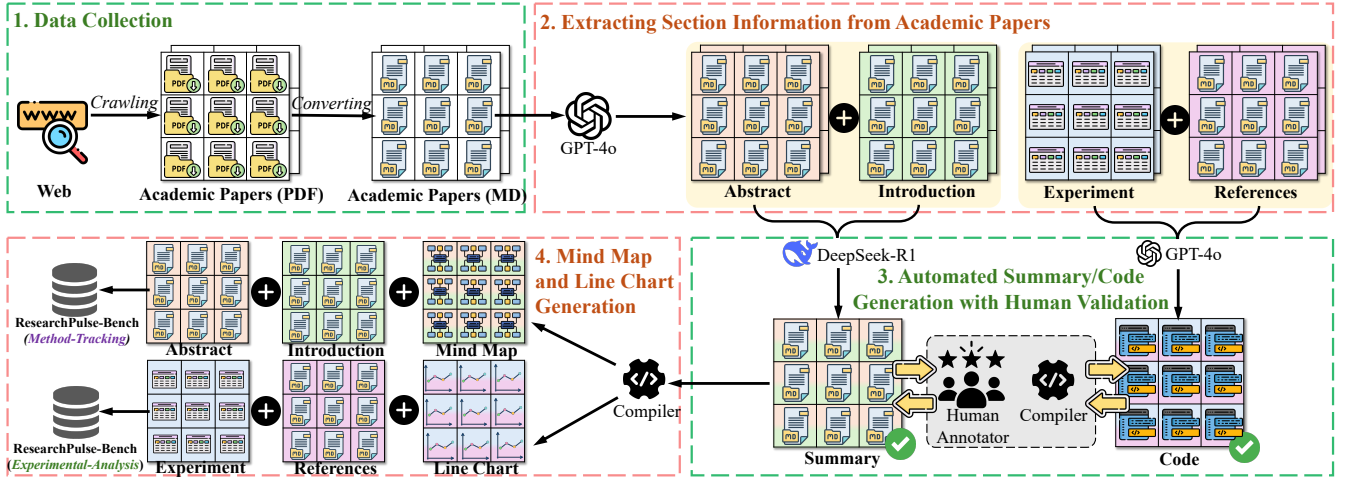


Figure 3: The construction pipeline of ResearchPulse-Bench.

publicly accessible via platforms such as **arXiv** and **OpenReview**, complying with their open-access terms. Each selected paper’s metadata—including title, authorship, publication date, and citation statistics—is extracted using public APIs or permitted crawling mechanisms. We then construct citation graphs by retrieving forward and backward references, enabling us to group semantically and temporally related papers into document clusters for downstream processing.

Data Processing. The collected papers are first parsed using PDF-to-markdown conversion tools to extract structured content. Section segmentation is performed to isolate the abstract, introduction, experiment, and reference sections. For semantic clustering, we encode each abstract with Sentence-BERT to generate vector representations, then apply K-Means to group papers by topic proximity. Each resulting cluster is labeled using high-frequency terms and reviewed manually. Within each cluster, we use DeepSeek-R1 or GPT-4o [44] to extract fine-grained elements including motivations, methods, and experimental results. These outputs are stored in markdown format and further rendered as mind map diagrams and temporal trend charts to facilitate structured inspection of research motivations, methodologies, and experimental trajectories. This processed data is subsequently used for visual generation, as illustrated in Figure 3.

Human Inspection. To ensure dataset reliability, we conduct a multi-stage human inspection. Annotators review the semantic clustering results to verify topical coherence and identify misassigned papers. For method-related content, we check whether motivations and methodologies are logically and chronologically consistent. For experiment-based summaries, we verify if extracted tables match their originals, whether citation years are correctly resolved, and whether metric values are consistent with ground truth. All visual outputs—mind maps and line charts—are reviewed for structural completeness and semantic fidelity. Annotator feedback is used to adjust clustering granularity and refine LLM prompts, forming a closed-loop quality assurance process.

Table 1: Token statistics for the Method-Tracking and Experimental-Analysis tasks.

Statistic	Method-Tracking		Experimental-Analysis	
	train	test	train	test
Total Samples				
Sample Count	1958	491	1550	320
Query Length (tokens)				
Minimum	55	78	301	3026
Maximum	22326	24031	46877	32971
Average	1161.16	1210.78	14402.73	16545.45
Answer Length (tokens)				
Minimum	63	141	331	407
Maximum	1722	2036	3081	2020
Average	344.54	355.6	885.98	928

4.1 Data Analysis

Dataset Scale and Token Statistics. ResearchPulse-Bench contains 100 citation-aware document clusters, each curated for either the Method-Tracking or Experimental-Analysis task. Table 1 summarizes the number of samples as well as the input/output token lengths. Method-Tracking queries are relatively concise, averaging around 1.1K tokens per input and 350 tokens per output. Experimental-Analysis queries are significantly longer (averaging over 14K tokens) and require structured extraction of tabular data, with outputs around 900 tokens. This range reflects the dual challenge of focused scientific summarization and long-context reasoning.

Cluster Distribution and Examples. Table 2 shows the distribution of document clusters across train and test splits. Each cluster forms a temporally ordered and semantically coherent series of papers. On average, clusters contain around 20 documents, with the largest containing over 30. This structure supports fine-grained modeling of research trajectories over time. Representative examples for both

Method-Tracking

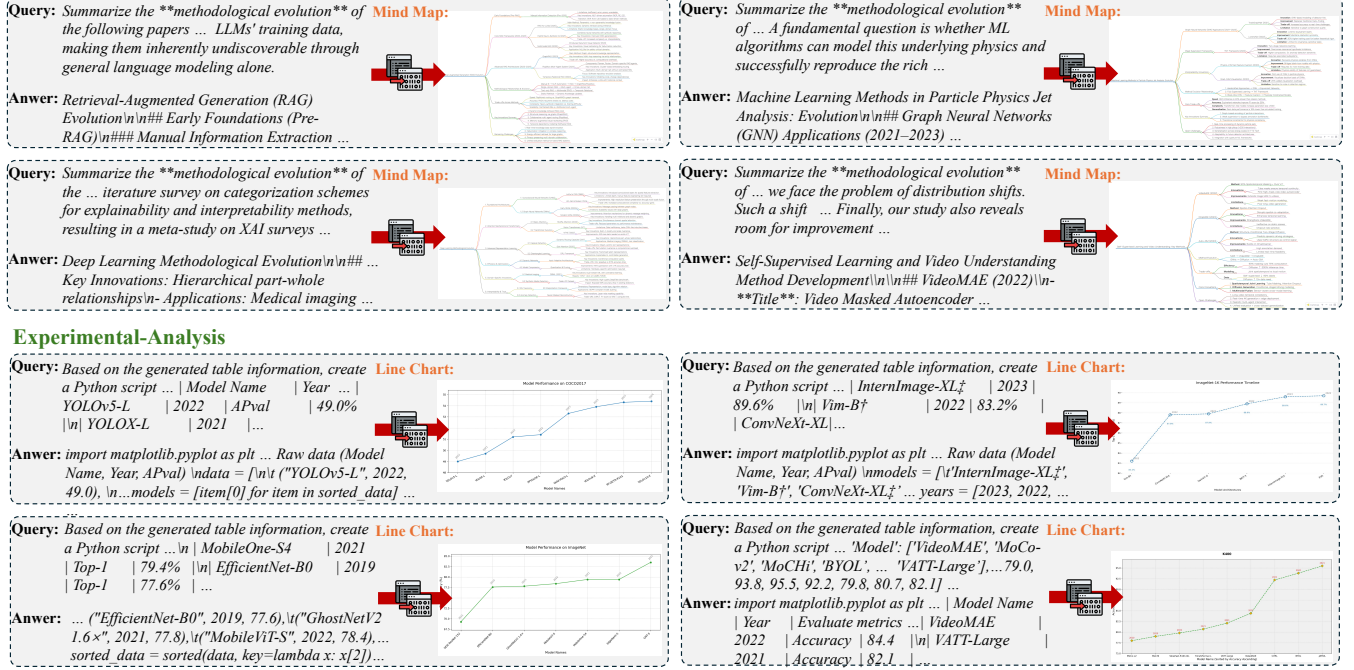


Figure 4: Representative examples from Method-Tracking (top) and Experimental-Analysis (bottom) tasks.

Table 2: Distribution of paper clusters.

Series Count	Train	Test	Total
Total	80	20	100
Minimum Papers	6	8	14
Maximum Papers	33	31	38
Average Papers	19.19	19.9	24.93

tasks are shown in Figure 4, including a mind map from method-tracking and a line chart from experimental-analysis. These outputs highlight the benchmark's diversity and its support for downstream tasks such as trend forecasting and methodological comparison.

4.2 Evaluation Metrics

We employ a comprehensive set of evaluation metrics tailored to the multimodal outputs of ResearchPulse. For the method-tracking task, which produces structured textual summaries of motivations and methodologies, we use BERTScore (Precision, Recall, F1) and METEOR to assess semantic similarity and lexical overlap with references. To capture qualitative aspects such as fluency, factuality, and coherence, we additionally report GPT-Score—a human-aligned rating between 0 and 1 generated by GPT-4o along five dimensions: fluency, relevance, accuracy, creativity, and overall quality.

In the experimental-analysis task, the system outputs visual charts derived from extracted tables. To evaluate these generated figures, we adopt standard image quality metrics including Inception Score (IS), Fréchet Inception Distance (FID), Kernel Inception

Distance (KID), and CLIP-FID for perceptual and distributional alignment. We further assess structural fidelity using SSIM, MS-SSIM, PSNR, LPIPS, and CMMD.

When the system generates Python code to render visualizations, we evaluate functional correctness using the pass@1 metric, which reports the percentage of code generations that execute successfully without runtime errors. Together, these metrics comprehensively evaluate the system's outputs across semantics, visual quality, and functional reliability.

5 Experiment

Setup. ResearchPulse performs method-tracking and experimental-analysis through four core agents: Plan Agent, Mmap-Agent, and Lchart-Agent. The Plan Agent, utilizing Qwen-72B[45], is responsible for effectively distinguishing and interpreting the user's instructions. The Mmap-Agent is based on Qwen2.5-7B [46], while the Lchart-Agent leverages Qwen2.5-Coder-7B [47]. Both agents were fine-tuned for four epochs on a 4×80GB A100 GPU setup to optimize performance for their respective tasks.

Model. In the method-tracking and experimental-analysis tasks, Mmap-Agent and Lchart-Agent are compared with several state-of-the-art models, including both open-source and closed-source approaches. The open-source models include Qwen2.5-7B-Instruct [46], InternLM3-8B-Instruct [48], Llama-3.1-8B-Instruct [49], CodeLlama-7B-hf [50], and Qwen2.5-Coder-7B-Instruct [47]. Closed-source models include GPT-4o [44], Claude-3.7-Sonnet [51], and Gemini-1.5 Pro [52]. These models were selected based on their advanced

Table 3: (a) Main results: Performance comparison of Mmap-Agent with state-of-the-art models. The best result is highlighted in bold. (b) Ablation study: Impact of different module configurations on Mmap-Agent performance.

Model	Size	METEOR↑	BERTScore			GPT-4o Score				
			P↑	R↑	F1↑	Fluency↑	Relevance↑	Accuracy↑	Creativity↑	Quality↑
(a) Main results										
Qwen2.5-7B-Instruct	7B	42.60	88.80	86.23	87.49	92.10	87.14	81.57	78.72	85.10
InternLM3-8B-instruct	8B	46.14	87.46	87.25	87.35	89.45	85.21	80.34	78.43	83.46
Llama-3.1-8B-Instruct	8B	39.47	88.64	85.80	87.19	90.44	84.52	77.57	74.96	82.00
CodeLlama	7B	17.49	81.94	81.79	81.85	80.50	80.10	73.07	73.69	81.59
Qwen2.5-Coder-7B-Instruct	7B	41.03	88.78	85.74	87.22	91.68	86.62	80.71	78.65	84.57
GPT-4o	-	42.66	84.90	86.06	85.48	88.32	82.46	78.27	75.88	83.42
Claude-3.7-Sonnet	-	42.73	88.43	85.34	86.85	90.63	85.99	80.32	78.40	84.01
Gemini-1.5 Pro	-	43.25	88.75	83.42	85.99	90.26	85.48	79.95	77.14	83.32
Mmap-Agent	7B	46.14	90.90	89.89	90.39	92.16	87.62	82.29	79.54	85.59
(b) Ablation study										
w/o GPT-4o	7B	41.78	87.23	87.93	86.91	91.29	86.49	81.45	73.88	83.98
w/o Compiler	7B	44.13	88.19	84.67	83.49	90.18	83.73	80.18	76.11	81.69
w/o GPT-4o & Compiler	7B	42.23	86.11	83.99	82.83	90.05	82.99	79.64	75.32	82.71

performance in similar tasks and serve as key benchmarks for evaluating Mmap-Agent and Lchart-Agent.

5.1 Method-Tracking

Main Results. Mmap-Agent achieves superior performance across multiple evaluation metrics, as demonstrated in Table 3. It leads in BERTScore F1 (90.39) and METEOR (46.14), showcasing its exceptional ability to preserve semantic accuracy and generate fluent, contextually appropriate text. In GPT-4o-based evaluations, Mmap-Agent surpasses other models with an overall score, excelling in fluency (92.16), relevance (87.62), accuracy (82.29), creativity (79.54) and quality (85.59). While baseline models like InternLM3-8B-Instruct and Claude-3.7-Sonnet demonstrate strong performance in specific areas, such as METEOR or fluency, they fall short in providing balanced performance across all dimensions. InternLM3, for instance, achieves the highest METEOR score (46.14), but its BERTScore F1 (87.35) and GPT-4o results are lower, indicating weaker semantic alignment and generation quality. Similarly, Claude-3.7-Sonnet excels in fluency (90.63) and relevance (85.99), but underperforms in accuracy (80.32) and creativity (78.40). In contrast, Mmap-Agent’s well-rounded performance across fluency, relevance, accuracy, and creativity highlights its robustness in both linguistic fluency and semantic integrity. These results validate Mmap-Agent’s strength in method-level reasoning and generation tasks, where it consistently outperforms other models by generating coherent, precise, and diverse text.

Ablation Study. To evaluate the impact of key components on Mmap-Agent’s performance, we conducted ablation studies by systematically removing critical modules. As illustrated in Table 3, the removal of the GPT-4o module resulted in notable declines in fluency and creativity, with BERTScore F1 dropping to 86.91. When the Compiler module was excluded, there was a marked reduction

in precision and relevance, further lowering BERTScore F1 to 83.49. Removing both modules simultaneously caused an additional performance drop, with BERTScore F1 falling to 82.83. These results emphasize the importance of both modules in maintaining Mmap-Agent’s overall performance, particularly in ensuring high fluency, precision, and relevance in generated text.

5.2 Experimental-Analysis

Main Results. As shown in Table 4, Lchart-Agent outperforms most models across key metrics, achieving a Pass@1 score of 97.50, surpassing GPT-4o (96.25) and Claude-3.7-Sonnet (91.56). It also achieves the lowest FID of 6.73, demonstrating superior image quality. Additionally, Lchart-Agent excels in perceptual metrics like LPIPS (8.49), CMMD (4.06), and SSIM (55.56), indicating high visual and semantic consistency. However, other models show strengths in specific areas: Claude-3.7-Sonnet leads in CLIP-FID with 0.08, suggesting better semantic alignment, while GPT-4o outperforms in KID with 1.14, indicating sharper images. Despite these areas of strength for other models, Lchart-Agent maintains a competitive edge overall, consistently excelling in generating high-quality, semantically consistent images across multiple dimensions.

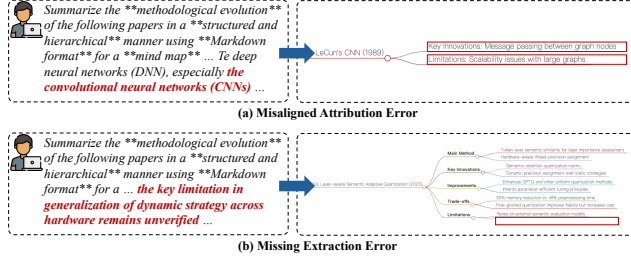
Ablation Study. As shown in Table 4, removing the GPT-4o module resulted in a decrease in IS to 2.32 and an increase in FID to 7.13, indicating the module’s crucial role in maintaining high-quality image generation. Similarly, excluding the Compiler module led to a reduction in performance, with Pass@1 dropping to 90.63 and FID rising to 8.02. Notably, the Compiler module also plays a key role in ensuring code executability, as evidenced by its direct impact on the Pass@1 score. When both modules were removed together, Pass@1 further declined to 90.00 and FID increased to 8.33, reinforcing the importance of these components in optimizing the model’s overall performance.

Table 4: (a) Main results: Performance comparison of Lchart-Agent with state-of-the-art models. The best result is highlighted in bold. (b) Ablation study: Impact of different module configurations on Lchart-Agent performance.

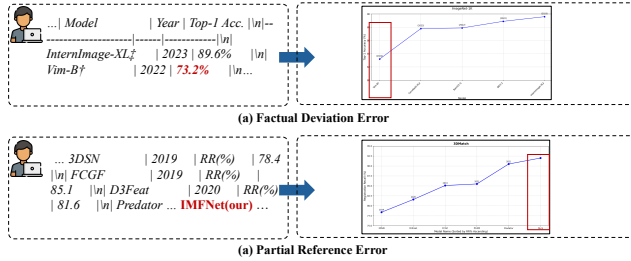
Model	Size	Pass@1↑	IS↑	FID↓	KID↓	CLIP-FID↓	LPIPS↓	CMMD↓	SSIM↑	PSNR↑	MS-SSIM↑
(a) Main results											
Qwen2.5-7B-Instruct	7B	65.00	0.18	12.73	4.10	2.09	24.35	9.85	29.13	7.93	26.51
InternLM3-8B-instruct	8B	68.13	0.57	29.91	2.08	4.42	10.39	15.22	22.97	5.14	17.91
Llama-3.1-8B-Instruct	8B	58.75	0.24	36.39	2.14	1.64	13.47	6.70	24.71	6.55	15.17
CodeLlama-7B-hf	7B	55.00	0.09	30.16	5.33	0.23	30.32	10.63	20.83	9.18	10.58
Qwen2.5-Coder-7B-Instruct	7B	73.13	0.77	10.69	2.15	5.50	16.71	12.02	32.33	6.37	21.18
GPT-4o	-	96.25	2.40	7.15	1.14	1.18	9.31	4.09	53.76	12.14	23.70
Claude-3.7-Sonnet	-	91.56	2.28	8.56	3.65	0.08	9.81	7.24	52.25	11.90	36.34
Gemini-1.5 Pro	-	90.31	1.10	8.68	4.33	5.75	10.40	5.47	42.33	9.57	23.25
Lchart-Agent	7B	97.50	2.65	6.73	1.17	1.10	8.49	4.06	55.56	12.49	36.55
(b) Ablation study											
w/o GPT-4o	7B	97.50	2.32	7.13	1.25	1.22	9.13	5.08	52.77	10.76	33.52
w/o Compiler	7B	90.63	2.26	8.02	1.37	1.14	9.22	6.16	51.67	11.53	26.85
w/o GPT-4o & Compiler	7B	90.00	1.98	8.33	1.56	1.31	9.38	7.32	48.79	9.84	22.91

5.3 Error Analysis

Method-Tracking



Experimental-Analysis

**Figure 5: The error examples of Mmap-Agent and Lchart-Agent.**

We conduct an in-depth analysis of typical failure cases across the Method-Tracking and Experimental-Analysis tasks, as illustrated in Figure 5. In the Method-Tracking task, a prominent issue is the misaligned attribution error, where the model assigns key innovations to incorrect sources—for example, attributing “message passing between graph nodes” to LeCun’s CNN (1989), despite it being a defining feature of later graph-based neural networks

such as GCNs or GATs. This indicates a misunderstanding of foundational methodological distinctions. Another frequent error is missing extraction, where the model overlooks essential components of the described method. In the case of layer-aware semantic adaptive quantization, the system fails to capture innovations such as dynamic precision assignment and semantic-aware evaluation metrics, resulting in incomplete or superficial summarization. For the Experimental-Analysis task, we observe factual deviation errors, such as incorrect reporting of Top-1 accuracy or misinterpretation of data trends from visualized results, which can undermine the credibility of the analysis. Furthermore, partial reference errors are evident when models mention results (e.g., IMFNet) without sufficient contextual grounding or comparative clarity. These observations underscore the necessity of more robust reasoning and alignment mechanisms to ensure faithful extraction and accurate interpretation of complex scientific content.

6 Conclusion

We present **ResearchPulse**, a modular agent system designed to perform multi-document scientific inference by extracting, aligning, and visualizing key research elements across thematically related papers. The system supports two complementary tasks: *Method-Tracking*, which constructs motivation–method chains rendered as mind maps, and *Experimental-Analysis*, which synthesizes benchmark trajectories from structured experimental results. To enable high-quality supervision, we introduce **ResearchPulse-Bench**, a citation-aware benchmark comprising 100 annotated paper clusters curated from arXiv and OpenReview. Experiments show that our agents, despite operating at 7B scale, achieve state-of-the-art performance in semantic alignment and multimodal output quality, outperforming both open- and closed-source baselines. Our work offers a new paradigm for tracking research evolution and structuring scientific knowledge at scale.

References

- [1] Jiangbin Zheng, Yile Wang, Cheng Tan, Siyuan Li, Ge Wang, Jun Xia, Yidong Chen, and Stan Z Li. Cvt-slr: Contrastive visual-textual transformation for sign language recognition with variational alignment. In *CVPR*, pages 23141–23150, 2023.
- [2] Cheng Tan, Zhangyang Gao, Siyuan Li, and Stan Z Li. Simvpv2: Towards simple yet powerful spatiotemporal predictive learning. *IEEE Transactions on Multimedia*, 2025.
- [3] Cheng Tan, Siyuan Li, Zhangyang Gao, Wenfei Guan, Zedong Wang, Zicheng Liu, Lirong Wu, and Stan Z Li. Openstl: A comprehensive benchmark of spatio-temporal predictive learning. *Advances in Neural Information Processing Systems*, 36:69819–69831, 2023.
- [4] Avi Caciularu, Matthew Peters, Jacob Goldberger, Ido Dagan, and Arman Cohan. Peek across: Improving multi-document modeling via cross-document question-answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1989, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [5] Miao Li, Jianzhong Qi, and Jey Han Lau. Compressed heterogeneous graph for abstractive multi-document summarization. In *AAAI*, volume 37, pages 13085–13093, 2023.
- [6] Xiangxiang Zhang, Jingxuan Wei, Donghong Zhong, Qi Chen, Caijun Jia, Cheng Tan, Jinming Gu, Xiaobo Qin, Zhiping Liu, Liang Hu, Tong Sun, Yuchen Wu, Zewei Sun, Chenwei Lou, Hua Zheng, Tianyang Zhan, Changbao Wang, Shuangzhi Wu, Zefa Lin, Chang Guo, Sihang Yuan, Riwei Chen, Shixiong Zhao, Yingping Zhang, Gaowei Wu, Bihui Yu, Jiahui Wu, Zhehui Zhao, Qianqian Liu, Ruofeng Tang, Xingyue Huang, Bing Zhao, Mengyang Zhang, and Youqiang Zhou. Structvrm: Aligning multimodal reasoning with structured and verifiable reward models, 2025.
- [7] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *EMNLP-IJCNLP*, pages 3730–3740, 2019.
- [8] Potsawee Manakul and Mark Gales. Long-span summarization via local attention and content selection. In *ACL*, pages 6026–6041, 2021.
- [9] Haopeng Zhang, Xiao Liu, and Jiawei Zhang. Hegel: Hypergraph transformer for long document summarization. In *EMNLP*, pages 10167–10176, 2022.
- [10] Chenlong Zhao, Xiwen Zhou, Xiaopeng Xie, and Yong Zhang. Hierarchical attention graph for scientific document summarization in global and local level. In *NAACL*, pages 714–726, 2024.
- [11] Shuaiqi Liu, Jiannong Cao, Ruosong Yang, and Zhiyuan Wen. Generating a structured summary of numerous academic papers: Dataset and method. *arXiv preprint arXiv:2302.04580*, 2023.
- [12] Chao-Chun Hsu, Erin Bransom, Jenna Sparks, Bailey Kuehl, Chenhao Tan, David Wadden, Lucy Lu Wang, and Aakanksha Naik. Chime: Llm-assisted hierarchical organization of scientific studies for literature review support. In *ACL*, pages 118–132, 2024.
- [13] Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Qingsong Wen, Wei Ye, et al. Autosurvey: Large language models can automatically write surveys. *NeurIPS*, 37:115119–115145, 2024.
- [14] Xun Liang, Jiawei Yang, Yezhaohui Wang, Chen Tang, Zifan Zheng, Simin Niu, Shichao Song, Hanyu Wang, Bo Tang, Feiyu Xiong, et al. Surveyx: Academic survey automation via large language models. *arXiv preprint arXiv:2502.14776*, 2025.
- [15] Jiakang Yuan, Xiangchao Yan, Botian Shi, Tao Chen, Wanli Ouyang, Bo Zhang, Lei Bai, Yu Qiao, and Bowen Zhou. Dolphin: Closed-loop open-ended auto-research through thinking, practice, and feedback. *arXiv preprint arXiv:2501.03916*, 2025.
- [16] Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. *arXiv preprint arXiv:2404.07738*, 2024.
- [17] Junde Wu, Jiayuan Zhu, and Yuyuan Liu. Agentic reasoning: Reasoning llms with tools for the deep research. *arXiv preprint arXiv:2502.04644*, 2025.
- [18] Cheng Tan, Qi Chen, Jingxuan Wei, Gaowei Wu, Zhangyang Gao, Siyuan Li, Bihui Yu, Ruifeng Guo, and Stan Z Li. Sketchagent: Generating structured diagrams from hand-drawn sketches. *arXiv preprint arXiv:2508.01237*, 2025.
- [19] Jingxuan Wei, Cheng Tan, Qi Chen, Gaowei Wu, Siyuan Li, Zhangyang Gao, Linzhuang Sun, Bihui Yu, and Ruifeng Guo. From words to structured visuals: A benchmark and framework for text-to-diagram generation and editing. In *CVPR*, pages 13315–13325, 2025.
- [20] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025.
- [21] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *EMNLP*, pages 404–411, 2004.
- [22] Günes Erkan and Dragomir R Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.
- [23] Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. An empirical survey on long document summarization: Datasets, models, and metrics. *ACM computing surveys*, 55(8):1–35, 2022.
- [24] Ansh N Vora, Rinit Mayur Jain, Aastha Sanjeev Shah, and Sheetal Sonawane. Extractive summarization using extended textrank algorithm. In *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*, pages 462–471, 2024.
- [25] Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. Dependency-based discourse parser for single-document summarization. In *EMNLP*, pages 1834–1839, 2014.
- [26] Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. Learning-based single-document summarization with compression and anaphoricity constraints. In *ACL*, pages 1998–2008, 2016.
- [27] Abishek Rao, Shivani Aithal, and Sanjay Singh. Single-document abstractive text summarization: A systematic literature review. *ACM Computing Surveys*, 57(3):1–37, 2024.
- [28] Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. Boookscore: A systematic exploration of book-length summarization in the era of llms. *arXiv preprint arXiv:2310.00785*, 2023.
- [29] Bo Pang, Erik Nijkamp, Wojciech Kryściński, Silvio Savarese, Yingbo Zhou, and Caiming Xiong. Long document summarization with top-down and bottom-up inference. In *EACL*, pages 1267–1284, 2023.
- [30] Tongyan Hua, Haotian Bai, Zidong Cao, Ming Liu, Dacheng Tao, and Lin Wang. Hi-map: Hierarchical factorized radiance field for high-fidelity monocular dense mapping. *arXiv preprint arXiv:2401.03203*, 2024.
- [31] Huy Quoc To, Ming Liu, Guangyan Huang, Hung-Nghiep Tran, Andr’e Greiner-Petter, Felix Beierle, and Akiko Aizawa. Skt5scisumm-revisiting extractive-generative approach for multi-document scientific summarization. *arXiv preprint arXiv:2402.17311*, 2024.
- [32] Yun-Zhu Song, Yi-Syuan Chen, and Hong-Han Shuai. Improving multi-document summarization through referenced flexible extraction with credit-awareness. In *ACL*, pages 1667–1681, 2022.
- [33] Simão Gonçalves, Gonçalo Correia, Diogo Pernes, and Afonso Mendes. Super-vising the centroid baseline for extractive multi-document summarization. In *EMNLP*, page 87, 2023.
- [34] Zixuan Zhang, Heba Elfardy, Markus Dreyer, Kevin Small, Heng Ji, and Mohit Bansal. Enhancing multi-document summarization with cross-document graph-based information extraction. In *EACL*, pages 1696–1707, 2023.
- [35] Congbo Ma, Wei Emma Zhang, Hu Wang, Haojie Zhuang, and Mingyu Guo. Disentangling specificity for abstractive multi-document summarization. In *IJCNN*, pages 1–8. IEEE, 2024.
- [36] Kung-Hsiang Huang, Philippe Laban, Alexander Richard Fabbri, Prafulla Kumar Choubey, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. Embrace divergence for richer insights: A multi-document summarization benchmark and a case study on summarizing diverse information from news articles. In *NAACL*, pages 570–593, 2024.
- [37] Catarina G Belem, Pouya Pezeskhpour, Hayate Iso, Seiji Maekawa, Nikita Bhutani, and Estevam Hruschka. From single to multi: How llms hallucinate in multi-document summarization. *arXiv preprint arXiv:2410.13961*, 2024.
- [38] Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. Assisting in writing wikipedia-like articles from scratch with large language models. In *NAACL*, pages 6252–6278, 2024.
- [39] Yutong Li, Lu Chen, Aiwei Liu, Kai Yu, and Lijie Wen. Chatcite: Llm agent with human workflow guidance for comparative literature summary. In *COLING*, pages 3613–3630, 2025.
- [40] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [41] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.
- [42] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [43] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.
- [44] OpenAI. Gpt-4o: Advancing multimodal ai with real-time audio, vision, and text integration. <https://openai.com/index/hello-gpt-4o/>, 2024. Technical Report.
- [45] Binyuan Hui, Jian Yang, Zeyu Cui, Jiayi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, et al. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.
- [46] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.
- [47] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

- [48] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024.
- [49] Meta AI. Llama 3: Open foundation for generative ai. <https://github.com/meta-llama/llama3>, 2024. 8B parameter instruct-tuned model.
- [50] Meta AI. Codellama: Open foundation models for code. <https://ai.meta.com/research/publications/code-llama-open-foundation-models-for-code/>, 2023. Code-specific adaptation of Llama 2.
- [51] Anthropic. The claude 3 model family: Opus, sonnet, haiku. <https://www.anthropic.com/news/claude-3-family>, 2024. Technical Report.
- [52] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.