AutoSurvey2: Empowering Researchers with Next Level Automated Literature Surveys

Siyi Wu[*1], XinLiang Chia[*2], Ziqian Bi[*2,3], Leyi Zhao[3],
Tianyang Wang[4], Junhao Song[5], Yichao Zhang[6], Keyu Chen[6], Benji Peng[7], Xinyuan Song[†8]

[1]University of Texas at Arlington, USA
[2]AI Agent Lab, USA
[3]Indiana University, USA
[4]Ohio State University, USA
[5]Imperial College London, UK
[6]Georgia Institute of Technology, USA
[7]Appcubic, USA
[8]Emory University, USA

Abstract

The rapid growth of research literature, particularly in large language models (LLMs), has made producing comprehensive and current survey papers increasingly difficult. This paper introduces AutoSurvey2, a multi-stage pipeline that automates survey generation through retrieval-augmented synthesis and structured evaluation. The system integrates parallel section generation, iterative refinement, and real-time retrieval of recent publications to ensure both topical completeness and factual accuracy. Quality is assessed using a multi-LLM evaluation framework that measures coverage, structure, and relevance in alignment with expert review standards. Experimental results demonstrate that AutoSurvey2 consistently outperforms existing retrieval-based and automated baselines, achieving higher scores in structural coherence and topical relevance while maintaining strong citation fidelity. By combining retrieval, reasoning, and automated evaluation into a unified framework, AutoSurvey2 provides a scalable and reproducible solution for generating long-form academic surveys and contributes a solid foundation for future research on automated scholarly writing. All code and resources are available at https://github.com/annihi1ation/auto_research.

1 Introduction

Survey papers play a central role in consolidating and disseminating knowledge across rapidly evolving research domains, offering comprehensive overviews of key advances, emerging trends, and open challenges (Chang et al., 2023; Khan et al., 2022; Pouyanfar et al., 2018; Zhao et al., 2023b). In fields such as artificial intelligence and large language models (LLMs) (Achiam et al., 2023; Goodfellow et al., 2016; Kirillov et al., 2023; LeCun et al., 2015), the exponential increase in publication volume has made the preparation of high-quality surveys increasingly demanding. Although recent progress in LLMs (Achiam et al., 2023; Touvron et al., 2023) enables long-context text generation (Chen et al., 2023b,a; Wang et al., 2024a), directly employing them for survey writing remains challenging for three major reasons.

First, context window limitations (Kaddour et al., 2023; Li et al., 2023, 2024; Liu et al., 2024b; Shi et al., 2023) restrict LLMs from processing large-scale literature corpora. Comprehensive surveys typically integrate

[*]Equal contribution.
[†]Correspondence to: Xinyuan Song ,xsong30@emory.edu

hundreds of papers, requiring tens of thousands of tokens—far beyond the stable generation capacity of current models. Second, depending solely on an LLM's internal knowledge is insufficient for constructing complete and up-to-date reviews (Ji et al., 2023; Shao et al., 2024; Wang et al., 2023), particularly for newly published works that may be absent from training data. This increases the risk of outdated or fabricated citations. Third, the lack of standardized evaluation protocols for LLM-generated academic content (Wang et al., 2024d; Yu et al., 2024; Zheng et al., 2024) makes quality assurance difficult, as expert evaluation is costly, subjective, and hard to scale.

To address these challenges, we propose AutoSurvey2, an end-to-end automated framework for generating academic survey papers. The system integrates multi-stage retrieval, modular content generation, and automated evaluation into a cohesive pipeline (Section 3). It operates in four coordinated stages: (1) survey outline construction through retrieval-guided planning, (2) section-level literature retrieval and analysis using high-recall embeddings, (3) section content generation grounded in retrieved evidence, and (4) final post-processing and assembly into a publication-ready LaTeX document. The pipeline is fully modular and compatible with multiple LLM providers, enabling scalable and reproducible survey generation.

Beyond generation, AutoSurvey2 incorporates two key mechanisms that directly address prior limitations: a real-time retrieval-augmented generation strategy (Gao et al., 2023; Jiang et al., 2023; Lewis et al., 2020) that ensures integration of recent literature, and a multi-LLM evaluation framework (Wang et al., 2024d; Yu et al., 2024; Zheng et al., 2024) that automatically assesses coverage, structure, and relevance in alignment with expert review criteria. Together, these components enable the system to produce surveys that are both comprehensive and stylistically coherent.

Empirical studies demonstrate that AutoSurvey2 achieves consistent improvements in content organization, topic relevance, and factual coverage compared with existing automated baselines. The generated surveys exhibit strong structural coherence and accurate citation grounding, approaching the quality of human-written reviews while remaining reproducible and scalable.

The main contributions of this work are as follows:

- We present AutoSurvey2, a modular and fully automated pipeline for generating long-form academic surveys through retrieval, structured planning, and large-model synthesis.
- We design a graph-based execution framework that decomposes survey writing into four coordinated stages, improving scalability and consistency.
- We demonstrate through experiments that the proposed system produces coherent, well-organized, and factually grounded survey papers, providing a reproducible foundation for automated academic writing.

In summary, this work contributes a unified framework that combines retrieval, reasoning, and evaluation for scalable and reliable survey generation, bridging the gap between automated synthesis and expert-level academic authorship.

2 Related Work

Auto literature survey generation via LLMs. Recent work has begun harnessing large language models (LLMs) to automate the writing of literature surveys. AutoSurvey (Wang et al., 2024b) introduced a structured pipeline for LLM-driven survey generation to address the exploding volume of publications in rapidly evolving fields. The first version of AutoSurvey employs multiple specialised LLMs in a staged framework: relevant papers are initially retrieved via embedding-based search, and an outline is produced with an outline-generation LLM. Each section and subsection is then generated by separate, topic-specific LLM modules. The approach explicitly tackles the challenges of (1) limited context windows (through retrieval and chunked generation) and (2) incomplete parametric knowledge (via real-time Retrieval-Augmented Generation, RAG) (Khuong and Rachmat, 2023; Maiorino et al., 2023; Tan et al., 2024). AutoSurvey further integrates dedicated revision modules to enhance section coherence and introduces an evaluation mechanism in which multiple LLMs act as judges to score generated content for citation quality, coverage, and logical

structure (Dominguez-Olmedo et al., 2023). This modular pipeline enables near-comprehensive coverage and rigorous quality control, as demonstrated by extensive benchmarking against human-written surveys (Jiang et al., 2019; Zhao et al., 2023a).

In parallel, the Auto-survey Challenge established a shared task to systematically evaluate LLMs' capabilities to both compose and peer-review survey articles (Wan et al., 2023). In this challenge, LLM-based systems are required to autonomously write survey drafts and participate in an iterative, LLM-mediated peer-review process. Systems are evaluated on writing quality, factual accuracy, citation relevance, and review insight, with human organisers providing ground-truth assessments. These works collectively position LLMs as collaborative tools for survey generation, able to accelerate drafting while still requiring human validation to ensure scientific rigour.

LLMs for data annotation and scientific workflows. Beyond survey writing, LLMs have increasingly been integrated into broader scientific research workflows, including data annotation, survey design, and automated evaluation. Recent surveys (Joos et al., 2024; Ke and Ng, 2025; Kitadai et al., 2024) outline how LLMs are applied across the research lifecycle, which from hypothesis generation and experimental planning to manuscript drafting and peer review. For data annotation, LLMs such as GPT-4.1 and Llama 4 are leveraged to generate labels or synthetic data at scale, reducing manual effort but raising challenges in annotation consistency and quality assurance (Kang and Xiong, 2024; Rewina et al., 2025; Ziming et al., 2025). Quality control typically requires hybrid human-in-the-loop approaches, where human experts validate or curate LLM-generated outputs. In survey design, LLMs can automatically draft questionnaire items and suggest answer choices (Wuttke et al., 2024). However, expert intervention remains essential for methodological soundness and contextual adaptation.

A further line of research investigates LLMs as evaluators or judges in scientific workflows. Also some research assess the use of LLMs for summarising and evaluating open-ended survey responses (Gao et al., 2024), showing that LLM-based ratings align moderately with human judgments, but may miss subtle content or context-specific nuances. These studies demonstrate both the scalability of LLMs for automating tedious research tasks and their current limitations in factual verification, nuanced reasoning, and domain-specific reliability. The prevailing consensus is that LLMs are most effective as assistive tools in scientific research when complemented by human expertise for final validation and interpretation.

3 Methodology

3.1 Overview

Figure 1 illustrates the architecture of AutoSurvey2, an automated framework designed to generate comprehensive academic survey papers. Building upon recent advances in automated literature review generation (Bosselut et al., 2018; Cho et al., 2018; Wang et al., 2019), the system integrates large language models (LLMs), semantic vector retrieval, and a modular graph-based execution mechanism to emulate the scholarly writing process in a structured and reproducible manner. The workflow proceeds through four interconnected stages that together form a complete pipeline for survey generation. It begins with the construction of a semantically indexed database, which serves as the foundation for all subsequent operations. In this stage, metadata from ArXiv is collected and processed into dense embeddings using transformer-based encoders (Nussbaum et al., 2024). The resulting representations are stored in a PostgreSQL database equipped with `pgvector` support, enabling efficient similarity searches that underpin the retrieval capabilities of the entire system.

Once the database is established, the process transitions to the research planning stage, where a user-defined topic $\mathcal{T}$ serves as input. The system retrieves a relevant subset of papers and employs an LLM to analyze them, producing a structured outline $\mathcal{O} = \{s_1, s_2, \ldots, s_m\}$ that captures the main themes, methods, and research directions associated with $\mathcal{T}$ (Shao et al., 2024). Each element s_i in the outline is further explored during the research phase, which retrieves section-specific papers, identifies their methodological contributions, and synthesizes this information into structured prompts. These prompts guide the subsequent generation phase, where the system produces coherent academic text with inline IEEE-style citations (Balepur et al.,

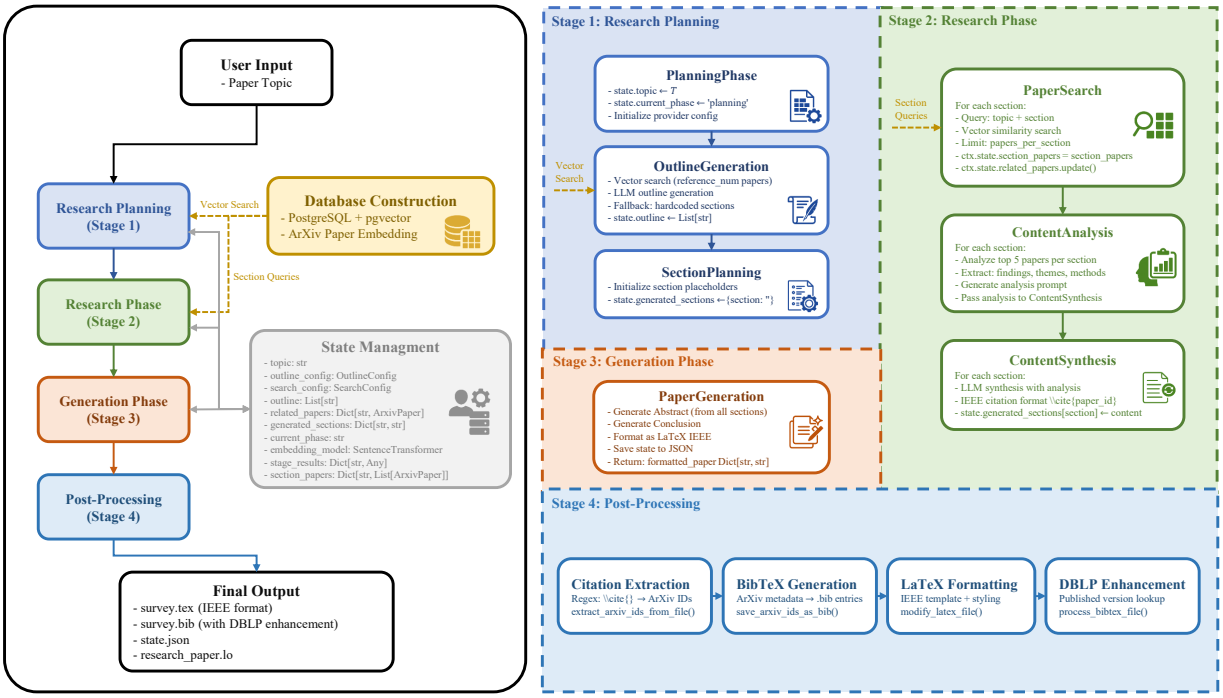

Figure 1: Overall workflow of AutoSurvey2. The system operates through four sequential stages organized as a graph-based state machine: *Database Construction* (prerequisite), *Research Planning* (Stage 1), *Research Phase* (Stage 2), *Generation Phase* (Stage 3), and *Post-processing* (Stage 4). Each stage comprises specialized functional nodes that communicate through a shared global state Σ.

2023). The final post-processing stage refines the manuscript by extracting citation data, generating BibTeX entries, and applying IEEE formatting. During this process, DBLP is queried to replace preprint references with their formally published counterparts, thereby enhancing citation accuracy and overall document quality.

All stages are orchestrated through a graph-based state machine that ensures systematic and reproducible execution. The system employs a directed acyclic graph (DAG) in which each node corresponds to a specialized operation—such as `OutlineGeneration`, `PaperSearch`, or `ContentSynthesis`—and edges encode task dependencies. These nodes share a global state Σ that stores intermediate data including the topic, outline, retrieved papers, generated sections, and configuration parameters. Maintaining this shared state guarantees consistency and traceability throughout the workflow while allowing modular development, testing, and debugging of individual components. The DAG-based design thus provides both flexibility and stability, enabling independent evolution of system modules without compromising the integrity of the end-to-end process. The following subsections detail the technical implementation of each stage, outlining the algorithms, models, and design principles that enable high-quality automated survey generation.

3.2 Database Construction

Before survey generation, a semantically indexed repository of academic papers is constructed to serve as the retrieval foundation for all downstream stages. Similar large-scale retrieval infrastructures have been used in prior works on semantic vector search and text embedding retrieval (Fang et al., 2020; Monir et al., 2024; Shan et al., 2018). This repository transforms raw ArXiv metadata into a vector-searchable database optimized for large-scale semantic similarity queries. Metadata is retrieved via the Kaggle API using the official `Cornell-University/arxiv` dataset, which contains comprehensive bibliographic records including paper identifiers, titles, abstracts, author lists, categories, and version histories. The system authenticates

using JSON-based credentials and downloads the complete dataset snapshot, which is then filtered to retain only papers in AI- and ML-related categories:

$$\mathcal{D}_{\text{filtered}} = \{p \in \mathcal{D}_{\text{raw}} \mid \texttt{category}(p) \cap \{\text{cs.AI, cs.CL, cs.CV, cs.LG, stat.ML}\} \neq \emptyset\}. \tag{1}$$

This filtering ensures that the resulting collection remains domain-specific while preserving comprehensive coverage across artificial intelligence and machine learning research areas, following common practices in embedding-based scientific indexing (Liu et al., 2024a).

For each paper $p \in \mathcal{D}_{\text{filtered}}$, a dense semantic embedding is generated by encoding its abstract using a transformer-based sentence encoder. Specifically, we employ the `nomic-embed-text-v2-moe` model (Nussbaum and Duderstadt, 2025), a mixture-of-experts architecture that produces 768-dimensional representations:

$$\mathbf{e}_p = f_\theta(\texttt{abstract}_p), \quad \mathbf{e}_p \in \mathbb{R}^{768}, \tag{2}$$

where f_θ denotes the pretrained encoder. This design follows the growing trend of mixture-of-expert embedding models for high-dimensional semantic representation (Nussbaum et al., 2025; Wu et al., 2025b). The model runs on GPU (CUDA or MPS) when available and gracefully falls back to CPU otherwise. Embeddings are computed in batches of 10,000 papers to balance memory efficiency and throughput, a strategy widely adopted in scalable embedding pipelines (Shan et al., 2018).

These representations are stored in a PostgreSQL database extended with the `pgvector` module, which enables native indexing and querying of high-dimensional vectors. Each record includes standard metadata fields such as identifiers, titles, authors, abstracts, categories, and structured version histories, along with an `embedding` field that serializes 768-dimensional vectors into PostgreSQL array format. This setup supports efficient semantic search through the inner-product operator `<->`, providing the retrieval backbone for subsequent stages of the system (Monir et al., 2024).

To facilitate incremental updates while minimizing redundant computation, the ingestion pipeline first checks for existing paper identifiers before insertion:

$$\mathcal{D}_{\text{new}} = \mathcal{D}_{\text{filtered}} \setminus \{p \mid \texttt{id}_p \in \mathcal{D}_{\text{existing}}\}, \tag{3}$$

where $\mathcal{D}_{\text{existing}}$ represents the set of papers already present in the database. Only new entries $\mathcal{D}_{\text{new}}$ undergo embedding generation and insertion, which substantially reduces processing time during incremental updates. The ingestion process is continuously monitored via detailed logging and progress tracking, ensuring transparency and fault tolerance. The final repository $\mathcal{D}_{\text{arxiv}}$ comprises hundreds of thousands of papers with precomputed embeddings, enabling sub-second retrieval of semantically related documents and forming the essential retrieval layer that powers automated survey generation (Fang et al., 2020; Nussbaum and Duderstadt, 2025).

3.3 Stage 1: Research Planning

The research planning stage transforms a user-specified topic into a structured survey blueprint through three sequential nodes—`PlanningPhase`, `OutlineGeneration`, and `SectionPlanning`—executed within the graph-based state machine. Each node reads from and writes to the shared global state Σ, maintaining consistency and traceability throughout the process. The workflow begins at the `PlanningPhase` node, which initializes the system state with the user-defined topic:

$$\Sigma.\texttt{topic} \leftarrow \mathcal{T}, \quad \Sigma.\texttt{current\_phase} \leftarrow \text{“planning”}. \tag{4}$$

During this initialization, the system also loads configuration parameters that define the operating environment, including the LLM provider (e.g., Ollama, OpenAI, or OpenRouter), the model identifier, and retrieval hyper-parameters. These settings are stored in $\Sigma.\texttt{stage\_results}$ to ensure consistent downstream behavior and reproducibility across all stages of the pipeline.

Once initialization is complete, the `OutlineGeneration` node constructs a hierarchical outline $\mathcal{O} = \{s_1, s_2, \ldots, s_m\}$ that determines the overall organizational structure of the survey. This step combines

large-scale semantic retrieval with LLM-based theme identification, following recent methods in automated survey generation (Lai et al., 2024a; Wang et al., 2024c). A reference corpus $\mathcal{D}_{\text{ref}}$ of K_{ref} papers (default $K_{\text{ref}} = 1500$) is retrieved by computing the cosine similarity between the topic embedding and the embeddings of all papers in the database:

$$\mathcal{D}_{\text{ref}} = \underset{D' \subset \mathcal{D}_{\text{arxiv}}, |D'| = K_{\text{ref}}}{\arg\max} \sum_{p \in D'} \frac{\mathbf{e}_{\mathcal{T}}^{\top} \mathbf{e}_p}{\|\mathbf{e}_{\mathcal{T}}\| \|\mathbf{e}_p\|}. \tag{5}$$

The titles and abstracts of $\mathcal{D}_{\text{ref}}$ are then provided as context to an LLM, which is prompted to identify major research themes, methodological paradigms, and open challenges related to the target topic. The prompt constrains the output to generate exactly m sections (default $m = 8$), omits subsections, and enforces a numbered list format (Lai et al., 2024a). The resulting outline is parsed using regular expressions to extract section titles and stored as $\Sigma.\texttt{outline}$, which serves as the structural foundation for subsequent stages.

Finally, the `SectionPlanning` node prepares the internal representation for each section defined in $\mathcal{O}$ by initializing empty content placeholders:

$$\Sigma.\texttt{generated\_sections} \leftarrow \{s_i \mapsto \text{“”} \mid s_i \in \mathcal{O}\}. \tag{6}$$

This ensures that every section has a corresponding entry in the global state even before generation begins, preserving structural completeness. Each section is also annotated with metadata describing its order and hierarchical relationships, facilitating coordinated retrieval and generation in later stages. Upon completion, the global state Σ contains the finalized outline $\mathcal{O}$, initialized section placeholders, and all configuration parameters required for content synthesis. This structured planning process establishes a coherent organizational backbone grounded in relevant literature, ensuring that the generated survey adheres to a logically consistent and thematically comprehensive design.

3.4 Stage 2: Research Phase

The research phase retrieves, analyzes, and synthesizes relevant literature for each section defined in the outline $\mathcal{O}$. It consists of three sequential nodes—`PaperSearch`, `ContentAnalysis`, and `ContentSynthesis`—that progressively transform raw paper collections into structured content suitable for LLM-based generation. For each section $s_i \in \mathcal{O}$, the `PaperSearch` node performs targeted retrieval to identify papers most relevant to that section's theme. The search query is formed by concatenating the main topic with the section title:

$$q_i = \mathcal{T} \oplus s_i, \tag{7}$$

where $\oplus$ denotes string concatenation. The query is embedded using the same encoder f_θ employed during database construction, yielding a vector representation $\mathbf{e}_{q_i} \in \mathbb{R}^{768}$. The system then retrieves the top-K_{sec} most similar papers (default $K_{\text{sec}} = 20$) from the database:

$$\mathcal{P}_i = \underset{P' \subset \mathcal{D}_{\text{arxiv}}, |P'| = K_{\text{sec}}}{\arg\max} \sum_{p \in P'} \text{sim}(\mathbf{e}_{q_i}, \mathbf{e}_p), \tag{8}$$

where $\text{sim}(\cdot, \cdot)$ represents cosine similarity computed via PostgreSQL's `<->` operator. The retrieved papers $\mathcal{P}_i$ are stored in $\Sigma.\texttt{section\_papers}[s_i]$, and their identifiers are aggregated in $\Sigma.\texttt{related\_papers}$ to maintain a global record of all referenced sources across sections.

After retrieval, the `ContentAnalysis` node examines the top-k papers (default $k = 5$) from each set $\mathcal{P}_i$ to extract structured insights. To balance computational efficiency with informational richness, only the most relevant subset of papers undergo detailed analysis. The node constructs a context string by concatenating the titles and abstracts of selected papers:

$$C_i = \bigoplus_{p \in \mathcal{P}_i[:k]} (\texttt{title}_p \oplus \texttt{abstract}_p). \tag{9}$$

This context is passed to an LLM along with a structured prompt that requests identification of key contributions, common methodologies, contrasting perspectives, technical nuances, and existing limitations. The model's response, denoted as A_i, provides a concise synthesis of these dimensions and is stored in Σ.`section_analysis`$[s_i]$, forming a distilled knowledge base for the subsequent synthesis stage.

Building on these analyses, the `ContentSynthesis` node generates draft text for each section by combining the analytical summary A_i, the retrieved paper set $\mathcal{P}_i$, and explicit stylistic and structural constraints. The LLM is instructed to write in a formal academic tone, ensure logical flow, use IEEE-style inline citations via `\cite{paper_id}` commands, and maintain cohesive coverage without introducing subsections. The generated section content is formalized as:

$$c_i = \text{LLM}(\mathcal{T}, s_i, \mathcal{P}_i, A_i \mid \theta, \rho_{\text{synthesis}}), \tag{10}$$

where θ represents the model parameters and $\rho_{\text{synthesis}}$ denotes the synthesis prompt template. The resulting drafts are stored in Σ.`generated_sections`$[s_i]$, while sections such as "Abstract" or "Conclusion" are deferred for later generation. This staged process ensures that each section is grounded in relevant literature, enriched with contextual understanding, and stylistically consistent across the entire survey (Wei et al., 2025).

3.5 Stage 3: Generation Phase

The generation phase, implemented by the `PaperGeneration` node, produces the final abstract and conclusion sections, formats all content into IEEE-compliant LaTeX, and saves the complete system state for reproducibility. After all main sections $\{c_i\}_{i=1}^{m}$ have been generated, the system synthesizes a concise abstract $\mathcal{A}$ that summarizes the survey's motivation, scope, and principal contributions. To capture the overall essence of the work while adhering to token constraints, a condensed summary is constructed by concatenating section titles with truncated content segments:

$$\mathcal{S}_{\text{summary}} = \bigoplus_{i=1}^{m} (s_i \oplus c_i[:500]), \tag{11}$$

where $c_i[:500]$ denotes the first 500 characters of section c_i. This summary is provided as context for an LLM prompt that generates a 250–300 word abstract following the conventional academic structure of motivation, methods, findings, and contributions (Lai et al., 2024b). The resulting abstract is defined as

$$\mathcal{A} = \text{LLM}(\mathcal{T}, \mathcal{S}_{\text{summary}} \mid \rho_{\text{abstract}}), \tag{12}$$

where ρ_{abstract} specifies the abstract generation prompt, and the output is stored in Σ.`generated_sections`["Abstract"]. Following the same approach, the system generates a conclusion $\mathcal{C}$ by prompting the LLM to compose closing remarks that summarise key insights, highlight the current research landscape, discuss future directions, and provide a reflective synthesis of the field. The process is formalised as

$$\mathcal{C} = \text{LLM}(\mathcal{T}, \mathcal{S}_{\text{summary}} \mid \rho_{\text{conclusion}}), \tag{13}$$

with the output constrained to 400–500 words and stored in Σ.`generated_sections`["Conclusion"], completing the full set of survey sections.

Once textual generation is finalised, all sections are transformed into IEEE-compliant LaTeX through the `_format_latex` method. This procedure constructs the full paper structure, including the preamble, title block, abstract environment, section organisation, and citation normalisation. The preamble defines the document class and required packages (e.g., `cite`, `amsmath`, `graphicx`), while the title block includes placeholders for author affiliations and funding acknowledgments. The abstract is wrapped in the standard `\begin{abstract}...\end{abstract}` environment with appropriate keywords, and each section s_i and its content c_i are rendered as `\section{$s_i$}` blocks with special characters escaped for compatibility. Inline citations are standardised to the `\cite{...}` format, replacing any temporary placeholders. The formatted output is represented as a dictionary mapping section names to their corresponding LaTeX strings:

$$\mathcal{F}_{\text{paper}} = \{\text{"Preamble"} \mapsto \ldots, \text{"Abstract"} \mapsto \ldots, s_i \mapsto \ldots\}. \tag{14}$$

To ensure reproducibility and facilitate downstream analysis, the complete system state is serialized to JSON and saved to the `result/` directory. The saved state includes metadata, topic definition, outline, related papers, retrieved paper sets, generated sections, and the formatted paper:

$$\mathcal{J}_{\text{state}} = \{\texttt{metadata}, \texttt{topic}, \texttt{outline}, \texttt{related\_papers}, \texttt{section\_papers}, \texttt{generated\_sections}, \texttt{final\_paper}\}. \tag{15}$$

Within this snapshot, `metadata` records timestamps, model identifiers, and configuration parameters, enabling precise reconstruction of experimental conditions. This serialized state allows individual sections to be modified or regenerated independently without re-running the full pipeline. The `PaperGeneration` node ultimately outputs $\mathcal{F}_{\text{paper}}$, which is passed to the post-processing stage for citation refinement and final compilation (Wu et al., 2025a).

3.6 Stage 4: Post-processing

The post-processing stage refines the generated survey into a publication-ready document through a series of automated transformations, including citation extraction, BibTeX generation, LaTeX formatting enhancement, and citation quality improvement via DBLP integration. Implemented by the `post_process_paper` function, this stage operates directly on the generated `.tex` file to ensure that all citations, formatting elements, and metadata conform to publication standards. The process begins with citation extraction, where the system parses the LaTeX source to identify all citation keys referenced in the text. Using regular expression matching, every occurrence of the `\cite{}` command is extracted:

$$\mathcal{C}_{\text{keys}} = \{c \mid c \in \text{match}(\texttt{\textbackslash cite\{\}}), c \neq \varnothing\}. \tag{16}$$

yielding a complete set of unique citation identifiers (typically ArXiv IDs). Multiple citations within a single command are split on commas and whitespace to ensure full coverage. For each key $c \in \mathcal{C}_{\text{keys}}$, the system retrieves the corresponding metadata from the ArXiv snapshot and converts it into an IEEE-style BibTeX entry following the `@misc` schema:

$$\texttt{BibEntry}(c) = g_\phi(\texttt{metadata}_c), \tag{17}$$

where g_ϕ maps metadata fields such as authors, title, year, and categories into structured BibTeX format. Author names are formatted in "FirstName LastName" order using the parsed author field, and publication years are inferred from ArXiv ID prefixes. The resulting entries are written to a `.bib` file co-located with the LaTeX source, ensuring full bibliographic consistency.

Once the citation data have been assembled, the system refines the generated LaTeX source to ensure IEEE-compliant formatting. The `modify_latex_file` function performs a series of structural enhancements: it replaces the default `cite` package with `natbib` to enable advanced citation control, and integrates additional packages such as `tikz`, `forest`, `booktabs`, and `hyperref` to support high-quality typesetting. Hyperlink colours are configured in dark blue to improve readability and maintain a professional appearance. The title block is automatically derived from the filename, converted to title case while preserving acronyms (e.g., "MLLM"), and suffixed with "A Comprehensive Survey." The abstract is enclosed within the standard `\begin{abstract}...\end{abstract}` environment, and each section s_i is formatted as a `\section{$s_i$}` block with escaped special characters for syntactic safety. The bibliography is appended through `\bibliographystyle{apalike}` and `\bibliography{filename}` commands, while consistent vertical spacing is inserted before section headers to enhance visual clarity. Collectively, these adjustments ensure that the final LaTeX document compiles seamlessly and conforms to professional academic presentation standards (Lee et al., 2025).

To further enhance citation quality, the system integrates with the DBLP database to identify formally published versions of ArXiv pre-prints. For each BibTeX entry, the `process_bibtex_file` function constructs a DBLP query from the title and first author, retrieves the search results, and parses available BibTeX entries. If a formally published version is found, the system replaces the ArXiv entry while retaining the original

cite key for consistency; otherwise, the original entry is preserved with a comment indicating that only a pre-print is available. This conditional replacement can be formalised as

$$\texttt{BibEntry}_{\text{final}}(c) = \begin{cases} \texttt{BibEntry}_{\text{DBLP}}(c) & \text{if published version exists,} \\ \texttt{BibEntry}_{\text{ArXiv}}(c) & \text{otherwise.} \end{cases} \tag{18}$$

Duplicate keys are subsequently removed to maintain a clean bibliography. Finally, the post-processing pipeline assembles all outputs into a finalised directory:

$$\mathcal{D}_{\text{output}} = \{\texttt{survey.tex}, \texttt{survey.bib}, \texttt{figs/}\}, \tag{19}$$

ready for compilation using `pdflatex` and `bibtex`. The resulting PDF constitutes a fully formatted academic survey with verified references and publication-ready IEEE styling.

4 Prompt Design

Effective prompt engineering is essential for ensuring that large language models (LLMs) generate coherent, factual, and academically rigorous content throughout the AutoSurvey2 pipeline. Each stage of the workflow is guided by a compact and structured prompt that defines the task, provides contextual grounding, and enforces explicit stylistic and structural constraints. During outline generation, the model acts as a survey-outline constructor that receives the topic and a reference corpus containing paper titles and abstracts. It is instructed to produce a numbered list of section titles that follows standard academic structure while maintaining logical progression, balanced coverage, and stylistic consistency. For each section, a subsequent analysis prompt examines the most relevant retrieved papers, summarizing key contributions, shared patterns, technical approaches, and challenges into a concise, structured synthesis that forms the basis for generation. The following synthesis prompt transforms these insights into coherent academic prose, ensuring formal tone, logical organization, and accurate IEEE-style citations. Through these progressive stages, the system maintains factual grounding while producing text that integrates seamlessly within a LaTeX-based academic workflow (Amatriain, 2024; Sahoo et al., 2024).

After all main sections are generated, meta-section prompts are used to create the abstract and conclusion. The abstract prompt produces a 250–300 word summary highlighting the survey's motivation, scope, and contributions, while the conclusion prompt generates a 400–500 word synthesis discussing main findings, research trends, and open challenges. Across all stages, the prompt design adheres to several shared principles: clear role specification to align the model's behaviour with academic conventions; structured requirements that reduce ambiguity and ensure systematic coverage; contextual grounding in retrieved literature to prevent hallucinations; explicit formatting constraints that support deterministic parsing; and reasoning encouragement through step-by-step instructions to improve coherence. Collectively, these design strategies ensure that the generated survey maintains scholarly rigor, stylistic consistency, and factual reliability across all components of the automated writing pipeline (Chen et al., 2023c; Li, 2023).

5 Experiments

The experiments were conducted to assess both the effectiveness and completeness of the proposed pipeline in generating long-form academic survey papers. We designed a reproducible and transparent setup to ensure that all evaluations could be reliably replicated and interpreted.

The system infrastructure employed PostgreSQL as the primary database (`localhost:5451`, `research_db`) to manage paper metadata and retrieval results, with all vector operations executed using the `nomic-ai/nomic-embed-text-v1.5` embedding model (Nussbaum and Duderstadt, 2025). This model generates 768-dimensional representations that provide a strong balance between semantic accuracy and computational efficiency. GPT-4.1 served as the backbone language model for all generation tasks, including outline construction, section-level writing, and post-generation refinement. For retrieval-augmented

generation, vector search was configured with a top-k of 100 and a similarity threshold of 0.7 to filter semantically relevant documents. Each survey generation process considered up to 1,500 reference papers, producing eight sections per survey, with 20 candidate papers retrieved per section to ensure sufficient topical coverage while maintaining computational tractability.

To evaluate the quality of generated surveys, we adopted an automated assessment framework based on a Judge Agent that formalizes expert review criteria into quantifiable metrics. The framework evaluates three fundamental dimensions: *Coverage*, which measures the comprehensiveness of topic inclusion; *Structure*, which assesses logical organization and narrative coherence; and *Relevance*, which evaluates topical alignment and citation accuracy. Each dimension follows a 5-point Likert scale ranging from 1 (poor) to 5 (excellent), with detailed quality descriptors for intermediate levels. Evaluation prompts combine the full survey text, explicit scoring rubrics, and detailed descriptions of each criterion. The Judge Agent performs deterministic inference at zero temperature to ensure reproducibility, extracting structured numerical scores from the model output through pattern-based parsing. This automated evaluation strategy aligns with recent frameworks for LLM-based academic assessment and review automation (Bao et al., 2025; Wu et al., 2025a).

Each experiment used ten representative topics across domains such as computer science, mathematics, and physics. All generated surveys were evaluated both individually and in comparison to baseline systems to assess content coverage, structural quality, and relevance of citations, consistent with evaluation methodologies proposed in large-scale survey generation benchmarks (Lai et al., 2024a; Yan et al., 2025). The resulting scores were averaged across assessors and topics, providing a balanced view of overall system performance and enabling direct comparison across experimental configurations.

6 Results

Table 1 presents a comparative evaluation between AutoSurvey2, AutoSurvey, and a Naive RAG baseline across three dimensions: *Coverage*, *Structure*, and *Relevance*. Overall, AutoSurvey2 achieves the highest average score of 4.76, outperforming all baselines in every category. Compared with AutoSurvey, the improvements in both *Structure* and *Coverage* indicate that the integration of graph-based planning and semantic retrieval enhances the coherence and comprehensiveness of the generated surveys. The Naive RAG approach, while capable of producing generally relevant content, exhibits weak structural consistency due to its lack of hierarchical topic decomposition and state tracking. These findings confirm that structured orchestration of LLM modules plays a decisive role in generating logically organized and contextually grounded survey papers.

Table 1: Comparison between AutoSurvey2, AutoSurvey, and Naive RAG.

Methods	Coverage	Structure	Relevance	Avg.
Naive RAG	4.46	3.66	4.73	4.23
AutoSurvey	4.66	4.43	4.86	4.60
AutoSurvey2 (Ours)	4.72	4.68	4.88	4.76

To further examine the contribution of key modules, we conducted an ablation study summarized in Table 2. When the planning component was removed, the system experienced the most significant performance drop, particularly in *Structure* (from 4.68 to 3.78), reflecting the importance of global topic decomposition for maintaining logical continuity across sections. Similarly, removing the refactoring stage led to decreased *Structure* and *Relevance*, as the absence of iterative refinement reduced inter-section alignment and stylistic uniformity. Despite these degradations, both ablated variants still outperform earlier baselines, demonstrating that the underlying retrieval and synthesis mechanisms remain robust. Together, these results highlight that the modular design of AutoSurvey2 —especially its planning and refactoring components—is essential for achieving high-quality, academically coherent survey generation.

Table 2: Ablation study results for AutoSurvey2 with different components removed.

Methods	Coverage	Structure	Relevance	Avg.
AutoSurvey2	4.72	4.68	4.88	4.76
AutoSurvey2 w/o planner	4.51	3.78	4.77	4.35
AutoSurvey2 w/o refactor	4.69	4.22	4.79	4.57

7 Conclusion

This work presented AutoSurvey2, a multi-stage framework for generating long-form academic survey papers through structured planning, retrieval augmentation, and iterative refinement. By decomposing the writing process into sequential yet interdependent phases, the system effectively mitigates the limitations of large language models related to context length and incomplete domain knowledge. The integration of a retrieval-augmented database ensures timely inclusion of newly published research, thereby reducing citation obsolescence. Experimental results demonstrate that AutoSurvey2 consistently outperforms prior baselines in terms of coverage, structural coherence, and relevance, achieving survey quality that approaches human-authored standards. These findings validate the effectiveness of combining semantic retrieval, modular planning, and LLM-driven synthesis for scalable academic writing.

8 Limitations

Despite its strong performance, AutoSurvey2 still faces several constraints. The system's output quality is inherently dependent on the completeness and accuracy of the underlying literature database—missing or misclassified entries may cause important works to be overlooked. Moreover, although parallelized section generation accelerates the process, it can occasionally introduce stylistic inconsistencies that require additional refinement during post-processing. Finally, since the framework relies on large language models for both generation and evaluation, it inherits their known limitations, including potential factual inaccuracies, citation errors, or subtle biases derived from pretraining data. Addressing these issues will require tighter integration of verification modules and human-in-the-loop review mechanisms in future iterations.

9 Ethical Statement

The automation of literature synthesis through large language models introduces both opportunities and ethical challenges. On one hand, such systems can democratize access to scholarly knowledge by enabling efficient exploration of vast research corpora. On the other hand, they may inadvertently propagate factual errors, biased interpretations, or incorrect citations if not carefully verified. To mitigate these risks, AutoSurvey2 incorporates citation validation routines and encourages human review of all generated outputs prior to dissemination. Furthermore, we emphasize respect for intellectual property and data provenance when using public datasets. Ultimately, AutoSurvey2 is designed not as a replacement for human scholarship, but as a tool to augment academic productivity and support critical, expert-driven research synthesis.

References

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

Xavier Amatriain. 2024. Prompt Design and Engineering: Introduction and Advanced Methods. arXiv:2401.14423 [cs.CL] https://arxiv.org/abs/2401.14423

Nishant Balepur, Jie Huang, and Kevin Chen-Chuan Chang. 2023. Expository text generation: Imitate, retrieve, paraphrase.

Tong Bao, Mir Tafseer Nayeem, Davood Rafiei, and Chengzhi Zhang. 2025. SurveyGen: Quality-Aware Scientific Survey Generation with Large Language Models. arXiv:2508.17647 [cs.CL] https://arxiv.org/abs/2508.17647

Antoine Bosselut, Asli Celikyilmaz, Xiaodong He, Jianfeng Gao, Po-Sen Huang, and Yejin Choi. 2018. Discourse-aware neural rewards for coherent text generation.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2023c. Unleashing the Potential of Prompt Engineering in Large Language Models: A Comprehensive Review. arXiv:2310.14735 [cs.CL] https://arxiv.org/abs/2310.14735

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023b. Extending context window of large language models via positional interpolation.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023a. LongLoRA: Efficient Fine-tuning of Long-Context Large Language Models.

Woon Sang Cho, Pengchuan Zhang, Yizhe Zhang, Xiujun Li, Michel Galley, Chris Brockett, Mengdi Wang, and Jianfeng Gao. 2018. Towards coherent and cohesive long-form text generation.

Ricardo Dominguez-Olmedo, Moritz Hardt, and Celestine Mendler-Dünner. 2023. Questioning the Survey Responses of Large Language Models. doi:10.48550/ARXIV.2306.07951

Kuan Fang, Long Zhao, Zhan Shen, RuiXing Wang, RiKang Zhour, and LiWen Fan. 2020. Beyond Lexical: A Semantic Retrieval Framework for Textual Search Engine. https://arxiv.org/abs/2008.03917

Fan Gao, Hang Jiang, Rui Yang, Qingcheng Zeng, Jinghui Lu, Moritz Blum, Tianwei She, Yuang Jiang, and Irene Li. 2024. Evaluating Large Language Models on Wikipedia-Style Survey Generation. 5405–5418 pages. doi:10.18653/v1/2024.findings-acl.321

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep learning.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. 38 pages.

Xiao-Jian Jiang, Xian-Ling Mao, Bo-Si Feng, Xiaochi Wei, Bin-Bin Bian, and Heyan Huang. 2019. HSDS: An Abstractive Model for Automatic Survey Generation. 70–86 pages. doi:10.1007/978-3-030-18576-3_5

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active Retrieval Augmented Generation. 7969–7992 pages.

Lucas Joos, Daniel A. Keim, and Maximilian T. Fischer. 2024. Cutting Through the Clutter: The Potential of LLMs for Efficient Filtration in Systematic Literature Reviews. doi:10.48550/ARXIV.2407.10652

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models.

Hao Kang and Chenyan Xiong. 2024. ResearchArena: Benchmarking Large Language Models' Ability to Collect and Organize Information as Research Agents. doi:10.48550/ARXIV.2406.10291

Ping Fan Ke and Ka Chung Ng. 2025. Human-AI Synergy in Survey Development: Implications from Large Language Models in Business and Research. 39 pages. doi:10.1145/3700597

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in vision: A survey. 41 pages.

Thanh Gia Hieu Khuong and Benedictus Kent Rachmat. 2023. Auto-survey Challenge. doi:10.48550/ARXIV.2310.04480

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. 4015–4026 pages.

Ayato Kitadai, Kazuhito Ogawa, and Nariaki Nishino. 2024. Examining the Feasibility of Large Language Models as Survey Respondents. 3858–3864 pages. doi:10.1109/bigdata62323.2024.10825497

Ting-Han Lai, Chia-Wei Liu, and Yun-Nung Chen. 2024a. Step-by-Step Survey Generation: Benchmarking LLMs on Structured Scientific Writing Tasks. https://arxiv.org/abs/2407.15906

Yuxuan Lai, Yupeng Wu, Yidan Wang, Wenpeng Hu, and Chen Zheng. 2024b. Instruct Large Language Models to Generate Scientific Literature Survey Step by Step. arXiv:2408.07884 [cs.CL] https://arxiv.org/abs/2408.07884

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. 436–444 pages.

Yukyung Lee, Soonwon Ka, Bokyung Son, Pilsung Kang, and Jaewook Kang. 2025. Navigating the Path of Writing: Outline-guided Text Generation with Large Language Models. arXiv:2404.13919 [cs.CL] https://arxiv.org/abs/2404.13919

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. 9459–9474 pages.

Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. 2023. How Long Can Context Length of Open-Source LLMs truly Promise?

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context LLMs Struggle with Long In-context Learning.

Yinheng Li. 2023. A Practical Survey on Zero-shot Prompt Design for In-context Learning. arXiv:2309.13205 [cs.CL] https://arxiv.org/abs/2309.13205

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. 157–173 pages.

Y. Liu et al. 2024a. Relevance Filtering for Embedding-based Retrieval. https://arxiv.org/abs/2408.04887

Antonio Maiorino, Zoe Padgett, Chun Wang, Misha Yakubovskiy, and Peng Jiang. 2023. Application and Evaluation of Large Language Models for the Generation of Survey Questions. 5244–5245 pages. doi:10.1145/3583780.3615506

Solmaz Seyed Monir, Irene Lau, Shubing Yang, and Dongfang Zhao. 2024. VectorSearch: Enhancing Document Retrieval with Semantic Embeddings and Optimized Search. https://arxiv.org/abs/2409.17383

Zach Nussbaum et al. 2025. On the Theoretical Limitations of Embedding-Based Retrieval. https://arxiv.org/abs/2508.21038

Zach Nussbaum and Brandon Duderstadt. 2025. Training Sparse Mixture of Experts Text Embedding Models. arXiv:2502.07972 [cs.CL] https://arxiv.org/abs/2502.07972

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic Embed: Training a Reproducible Long Context Text Embedder. arXiv:2402.01613 [cs.CL]

Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. 2018. A survey on deep learning: Algorithms, techniques, and applications. 36 pages.

Bedemariam Rewina, Perez Natalie, Bhaduri S., Kapoor Satya, Gil Alex, Conjar Elizabeth, Itoku Ikkei, Theil David, Chadha Aman, and Nayyar Naumaan. 2025. Potential and Perils of Large Language Models as Judges of Unstructured Textual Data.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. arXiv:2402.07927 [cs.CL] https://arxiv.org/abs/2402.07927

Ying Shan, Jian Jiao, Jie Zhu, and J. C. Mao. 2018. Recurrent Binary Embedding for GPU-Enabled Exhaustive Retrieval from Billion-Scale Semantic Vectors. https://arxiv.org/abs/1802.06466

Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. 2024. Assisting in Writing Wikipedia-like miscs From Scratch with Large Language Models.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. 31210–31227 pages.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large Language Models for Data Annotation and Synthesis: A Survey. doi:10.48550/ARXIV.2402.13446

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. 2023. Efficient Large Language Models: A Survey. doi:10.48550/ARXIV.2312.03863

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity.

Qingyun Wang, Lifu Huang, Zhiying Jiang, Kevin Knight, Heng Ji, Mohit Bansal, and Yi Luan. 2019. PaperRobot: Incremental draft generation of scientific ideas.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2024a. Augmenting language models with long-term memory.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min Zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. 2024b. AutoSurvey: Large Language Models Can Automatically Write Surveys. doi:10.48550/ARXIV.2406.10252

Yue Wang, Yuhao Wang, Xinyu Hu, Jianbo Yuan, Xipeng Qiu, and Xuanjing Huang. 2024c. AutoSurvey: Large Language Model-Driven Automated Survey Generation. https://arxiv.org/abs/2403.06278

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. 2024d. PandaLM: An Automatic Evaluation Benchmark for LLM Instruction Tuning Optimization.

Zexuan Wei, Xinyuan Xu, Zihao Fu, Yichong Xu, Zhiruo Wang, Ruiqi Zhong, and Chenguang Zhu. 2025. PlanGen: Structuring Long-Form Generation via Hierarchical Planning with Large Language Models. https://arxiv.org/abs/2501.04512

Junjie Wu, Jiangnan Li, Yuqing Li, Lemao Liu, Liyan Xu, Jiwei Li, Dit-Yan Yeung, Jie Zhou, and Mo Yu. 2025b. SitEmb-v1.5: Improved Context-Aware Dense Retrieval for Semantic Association and Long Story Comprehension. https://arxiv.org/abs/2508.01959

S. Wu et al. 2025a. Automated Review Generation Method Powered by Large Language Models. doi:10.1093/nsr/nwaf169

Alexander Wuttke, Matthias Aßenmacher, Christopher Klamm, Max M. Lang, Quirin Würschinger, and Frauke Kreuter. 2024. AI Conversational Interviewing: Transforming Surveys with LLMs as Adaptive Interviewers. doi:10.48550/ARXIV.2410.01824

X. Yan et al. 2025. SURVEYFORGE: On the Outline Heuristics, Memory-Guided Retrieval and Generation of Survey Papers. https://aclanthology.org/2025.acl-long.609.pdf

Zhuohao Yu, Chang Gao, Wenjin Yao, Yidong Wang, Wei Ye, Jindong Wang, Xing Xie, Yue Zhang, and Shikun Zhang. 2024. KIEval: A Knowledge-grounded Interactive Evaluation Framework for Large Language Models.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023b. A survey of large language models.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023a. A Survey of Large Language Models. doi:10.48550/ARXIV.2303.18223

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena.

Luo Ziming, Yang Zonglin, Xu Zexin, Yang Wei, and Du Xinya. 2025. LLM4SR: A Survey on Large Language Models for Scientific Research.