SurveyGen-I: Consistent Scientific Survey Generation with Evolving Plans and Memory-Guided Writing

Jing Chen[1,2*] , **Zhiheng Yang**[1*†] , **Yixian Shen**[1], **Jie Liu**[1],
Adam Belloum[1], **Paola Grosso**[1], **Chrysa Papagianni**[1 †]
[1]University of Amsterdam, the Netherlands
[2]Vrije Universiteit Amsterdam, the Netherlands
j.chen12@student.vu.nl
{z.yang, y.shen, j.liu5, a.s.z.belloum, p.grosso, c.papagianni}@uva.nl

Abstract

Survey papers play a critical role in scientific communication by consolidating progress across a field. Recent advances in Large Language Models (LLMs) offer a promising solution by automating key steps in the survey-generation pipeline, such as retrieval, structuring, and summarization. However, existing LLM-based approaches often struggle with maintaining coherence across long, multi-section surveys and providing comprehensive citation coverage. To address these limitations, we introduce *SurveyGen-I*, an automatic survey generation framework that combines coarse-to-fine retrieval, adaptive planning, and memory-guided generation. *SurveyGen-I* performs survey-level retrieval to construct the initial outline and writing plan, then dynamically refines both during generation through a memory mechanism that stores previously written content and terminology, ensuring coherence across subsections. When the system detects insufficient context, it triggers fine-grained subsection-level retrieval. During generation, *SurveyGen-I* leverages this memory mechanism to maintain coherence across subsections. Experiments across six scientific domains demonstrate that *SurveyGen-I* consistently outperforms previous works in content quality, consistency, and citation coverage. The code is available at https://github.com/SurveyGens/SurveyGen-I.

1 Introduction

The exponential expansion of scholarly literature, with thousands of new papers published daily, presents significant challenges for researchers to efficiently acquire and synthesize comprehensive knowledge. Consequently, writing survey papers requires substantial expertise and time commitment from researchers, as it traditionally involves an iterative and labor-intensive process of reading, note-taking, clustering, and synthesis (Carrera-Rivera et al., 2022). Recent advances in Large Language Models (LLMs) offer a promising solution to this bottleneck by enabling the automation of key steps in the survey-writing pipeline, such as retrieving, organizing, and summarizing large volumes of papers (Wang et al., 2024; Liang et al., 2025; Yan et al., 2025; Agarwal et al., 2024, 2025).

Despite recent advances, current LLM-based survey generation frameworks remain limited in several key aspects. **First, literature retrieval scope and depth remain limited.** Most systems rely on embedding-based similarity search over a fixed local paper database (Wang et al., 2024; Yan et al., 2025). While efficient, such surface-level matching often fails to identify important papers with different terminology or at a more conceptual level, resulting in incomplete or biased coverage. **Second, lack of cross-subsection consistency.** Most systems generate all subsections in parallel as isolated units without modeling dependencies across subsections (Wang et al., 2024; Liang et al., 2025; Yan et al., 2025). This often leads to redundant content, inconsistent terminology, and fragmented discourse. Moreover, they always follow a static, once-for-all outline that cannot adapt to newly generated content, making it difficult to maintain content coherence or integrate emerging insights. **Finally, indirect citations are often left unresolved.** Retrieval-augmented generation (RAG) typically extracts passages from retrieved papers to support writing. These passages often include indirect citations such as "[23]" and "Smith et al., 2022", which refer to influential prior work not present in the retrieval results. Without tracing these references, the system may miss influential papers, leading to incomplete citation coverage and broken linkage between ideas and their original sources.

To address these limitations, we introduce *SurveyGen-I*, an end-to-end framework for gen-

[*] Equal contribution.
[†] Corresponding author.

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 3687–3714
December 20-24, 2025 ©2025 Association for Computational Linguistics

erating academic surveys with consistent content and comprehensive literature coverage. **First**, *SurveyGen-I* performs coarse-to-fine literature retrieval at both the survey and subsection levels, augmented with citation expansion and LLM-based relevance scoring. This retrieval strategy substantially enhances literature coverage and topical relevance. **Second**, *SurveyGen-I* introduces **PlanEvo**, a dynamic planning mechanism powered by an evolving memory that continuously accumulates terminology and content from earlier generated subsections. This memory is used to construct the outline and a dependency-aware writing plan that captures the logical and conceptual relationships between subsections, allowing foundational topics to be generated before more advanced or derivative ones. As writing progresses, both the outline and plan are continually refined based on the updated memory, ensuring consistent terminology and coherent content flow across the survey. **Finally**, *SurveyGen-I* introduces **CaM-Writing**, which combines a citation-tracing module that detects indirect references in retrieved passages and resolves them back to their original source papers, with memory-guided generation that uses the evolving memory to maintain coherent terminology and content across the survey.

Extensive results highlight the strengths of *SurveyGen-I* across multiple dimensions of academic survey generation. Compared to the strongest baseline, *SurveyGen-I* yields an 8.5% improvement in content quality, a 27% increase in citation density, and more than twice as many distinct references, while also demonstrating significantly better citation recency. These improvements show the effectiveness of the system in enabling high-quality and consistent survey generation.

Our contributions are summarized as follows:

- We propose *SurveyGen-I*, a novel framework for high-quality, reference-rich, and consistent survey generation.
- We design a coarse-to-fine **Literature Retrieval** pipeline that combines keyword search, citation expansion, and LLM-based filtering to construct comprehensive paper sets at both survey and subsection levels.
- We introduce **PlanEvo**, a dynamic planning mechanism that constructs and continuously refines the outline and writing plan based on inter-subsection dependencies and evolving memory, ensuring coherent survey generation.
- We develop a **CaM-Writing** pipeline that combines citation tracing and memory-guided generation to improve reference coverage and ensure consistent, well-structured writing.

2 Related work

Component-Oriented and Hybrid Approaches. A longstanding approach to assisting literature surveys has been to tackle the problem in stages, where components handle retrieval, structuring, or writing, etc., independently (Susnjak et al., 2025; Lai et al., 2024; Li et al., 2025). Early systems organized citation sentences through clustering or classification (Nanba et al., 2000; Wang et al., 2018), employed rule-based content models (Hoang and Kan, 2010), or optimization-based extractive framework for related work generation (Hu and Wan, 2014). These systems often relied on static heuristics or surface-level topic associations, making them difficult to generalize across domains or maintain narrative coherence.

The rise of LLMs brought a wave of hybrid designs that integrated neural summarization with structured control (Fok et al., 2025; An et al., 2024; Zhang et al., 2024). Template-based generation (Sun and Zhuge, 2019) and extractive-abstractive hybrids (Shinde et al., 2022) introduced more fluent synthesis but retain rigid structures. Meanwhile, RAG-based methods (Lewis et al., 2020; Ali et al., 2024; Agarwal et al., 2024) enhanced retrieval fidelity (Gao et al., 2023), and agent-driven systems like the framework proposed by Brett and Myatt (2025), RAAI (Pozzobon and Pinheiro, 2024) and AutoSurveyGPT (Xiao, 2023) broke down the pipeline into stages such as retrieval, filtration, and generation. More recent works emphasize pre-writing planning, such as COI-Agent (Li et al., 2024b), which organizes references into conceptual chains to enhance topic coverage.

However, these designs remain fundamentally decomposed: content selection and writing are planned in isolation, and their execution often lacks global coordination across stages.

End-to-End Automated Literature Review/Survey Generation. With increasing demand for scalability and consistency, end-to-end frameworks have emerged to streamline the full pipeline from retrieval to synthesis. Multi-agent architectures (Sami et al., 2024; Rouzrokh and Shariatnia, 2025) have been widely used, and decompose the pipeline into specialized agent roles, mimicking human editorial workflows. Representatively, AutoSurvey (Wang et al., 2024) introduces a retrieval-outline-generation sequence that produces entire surveys via section-wise prompting. SurveyForge (Yan et al., 2025) extends this with memory modules and outline heuristics, aiming to enforce consistency across segments. SurveyX (Liang et al., 2025) scales this further by relying on larger models and a more complex pipeline, producing more robust and strict step-by-step outputs.

Despite these advances, many systems adopt a static and compartmentalized approach. Outlines are typically fixed in advance, with no capacity to revise structure based on intermediate content. Subsections are often generated in parallel, lacking shared context, which weakens narrative flow and increases repetition or terminology drift. Citation usage often lacks depth and evidential grounding, with models prone to hallucinations and missing fine-grained details (Kasanishi et al., 2023). In response, our work views survey writing as a dynamic process, one that requires adaptive planning, context-aware memory, and citation-traced retrieval. By continuously refining structural plans, maintaining cross-section consistency, and grounding generation in citation chains, we move toward producing more coherent and adaptive surveys.

3 Methodology

In this section, we propose *SurveyGen-I*, a novel framework for automatic survey generation. As shown in Figure 1, it consists of three key stages: (1) **Literature Retrieval (LR)** performs coarse-to-fine literature retrieval at both survey and subsection levels; (2) **Structure Planning with Dynamic Outline Evolution (PlanEvo)** generates a hierarchical outline and a dependency-aware writing plan, and dynamically updates both during generation to ensure cross-subsection consistency; (3) **CaM-Writing** generates each subsection with high content consistency and rich citation coverage, combining a citation-tracing mechanism to recover influential references, memory-guided skeleton planning for content consistency, and best-of-N draft selection to ensure high-quality generation.

3.1 LR: Literature Retrieval

To ensure that the generated survey is grounded with the most relevant and comprehensive research, our system adopts a coarse-to-fine literature retrieval strategy that operates at both the survey and subsection levels. As shown in Figure 1, this retrieval process provides the reference foundation for both structure planning (SDP; see Sec 3.2.1) and writing (CaM-Writing; see Section 3.3). The overall workflow is shown in Figure 2; implementation details are provided in Appendix B.1.

3.1.1 Survey-Level Retrieval for Structure Planning

For survey-level literature retrieval, an LLM is first prompted to generate a keyword set $\mathcal{K}$ based on the input topic T and its description E (see prompt in Figure 13). These keywords are used to query Semantic Scholar (Ammar et al., 2018), producing an initial set of papers $\mathcal{P}_{\text{init}}$. While keyword-based retrieval offers broad initial coverage, it may include irrelevant papers. To enhance topical precision, a semantic filtering step is applied. Specifically, both the input (T, E) and each paper abstract a_i are embedded using the `all-mpnet-base-v2` sentence transformer (Song et al., 2020). Candidate papers with high cosine similarity to the input (T, E) are retained, yielding a refined set $\mathcal{P}_{\text{sem}}$:

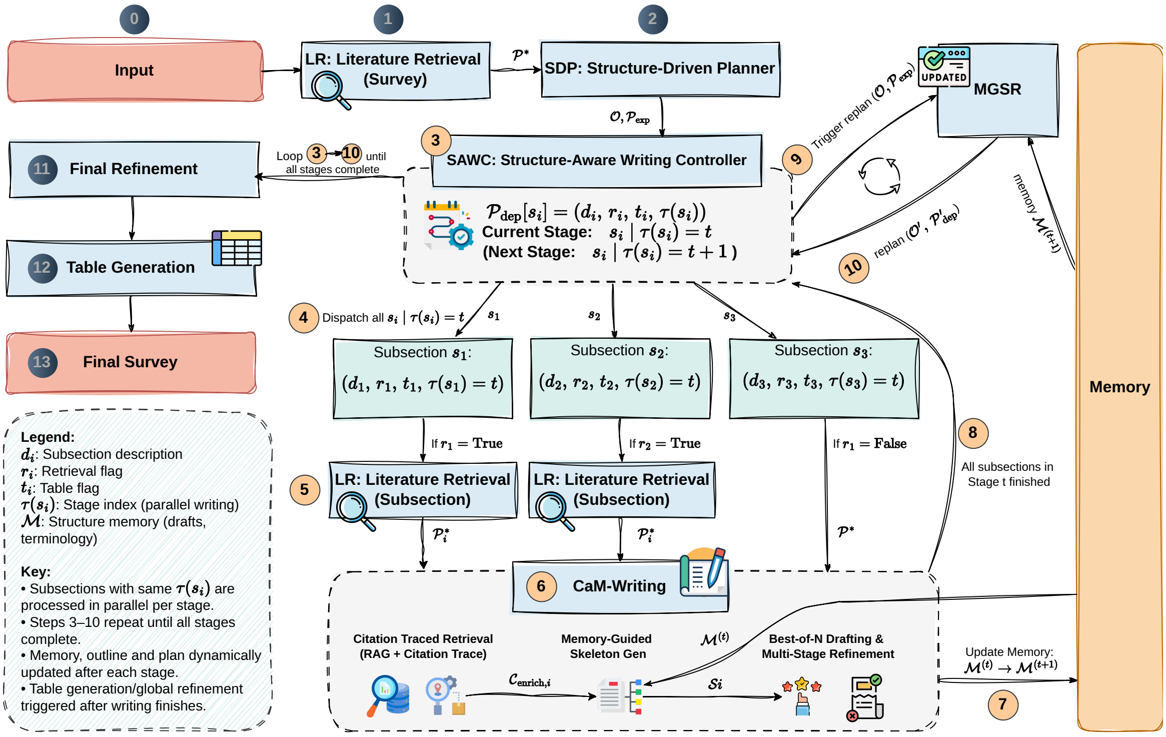
$$\mathcal{P}_{\text{sem}} = \{p_i \in \mathcal{P}_{\text{init}} \mid \cos(\mathbf{e}_{T,E}, \mathbf{e}_{a_i}) \geq \theta\}. \quad (1)$$

To improve coverage and avoid missing influential work, we perform a citation expansion step. Specifically, we select the top-10 papers from $\mathcal{P}_{\text{sem}}$ and retrieve both their references and citations using the Semantic Scholar API, forming the expanded candidate set $\mathcal{P}_{\text{exp}}$. We then remove duplicates and re-rank all papers by topical relevance, first using semantic similarity, followed by LLM-based relevance scoring with respect to (T, E). Finally, the top 30 papers are retained as the final **survey-level literature set** $\mathcal{P}^*$ to support outline generation (see prompt in Figure 16).

Full texts are retrieved based on the access information contained in the Semantic Scholar API metadata. We prioritize the `openAccessPdf` fields from the metadata or arXiv ID when available, and fall back to DOI links otherwise.

3.1.2 Subsection-Level Retrieval for Writing

In addition to survey-level retrieval for structure planning, subsection-level retrieval is optionally triggered during writing. For each subsection s_i with its description d_i, the **subsection-level paper set** $\mathcal{P}_i$ is constructed using the same retrieval pipeline as above, with (s_i, d_i) as input. Whether this step is performed is controlled by a retrieval flag r_i in the dependency-aware writing plan (see


0
Input
1
LR: Literature Retrieval (Survey)
$\mathcal{P}^*$
2
SDP: Structure-Driven Planner
$\mathcal{O}, \mathcal{P}_{\text{exp}}$
UPDATED
MGSR
Trigger replan $(\mathcal{O}, \mathcal{P}_{\text{exp}})$
9
3
SAWC: Structure-Aware Writing Controller
11 Final Refinement
Loop 3→10 until all stages complete
$\mathcal{P}_{\text{dep}}[s_i] = (d_i, r_i, t_i, \tau(s_i))$
Current Stage: $s_i \mid \tau(s_i) = t$
(Next Stage: $s_i \mid \tau(s_i) = t + 1$)
10
replan $(\mathcal{O}', \mathcal{P}'_{\text{dep}})$
memory $\mathcal{M}^{(t+1)}$
12 Table Generation
13 Final Survey
4 Dispatch all $s_i \mid \tau(s_i) = t$
s_1
s_2
s_3
Subsection s_1: $(d_1, r_1, t_1, \tau(s_1) = t)$
Subsection s_2: $(d_2, r_2, t_2, \tau(s_2) = t)$
Subsection s_3: $(d_3, r_3, t_3, \tau(s_3) = t)$
If r_1 = True
If r_2 = True
If r_1 = False
8
All subsections in Stage t finished
Memory
5
LR: Literature Retrieval (Subsection)
LR: Literature Retrieval (Subsection)
$\mathcal{P}_i^*$
$\mathcal{P}_i^*$
$\mathcal{P}^*$
6 CaM-Writing
Citation Traced Retrieval (RAG + Citation Trace)
Memory-Guided Skeleton Gen
$\mathcal{M}^{(t)}$
Best-of-N Drafting & Multi-Stage Refinement
$\mathcal{C}_{\text{enrich},i}$
s_i
Update Memory: $\mathcal{M}^{(t)} \rightarrow \mathcal{M}^{(t+1)}$
7

Legend:
d_i: Subsection description
r_i: Retrieval flag
t_i: Table flag
$\tau(s_i)$: Stage index (parallel writing)
$\mathcal{M}$: Structure memory (drafts, terminology)

Key:
- Subsections with same $\tau(s_i)$ are processed in parallel per stage.
- Steps 3–10 repeat until all stages complete.
- Memory, outline and plan dynamically updated after each stage.
- Table generation/global refinement triggered after writing finishes.

Figure 1: Overview of the *SurveyGen-I* pipeline for automatic academic survey generation. The system comprises three stages: (1) coarse-to-fine **Literature Retrieval** (LR); (2) **PlanEvo**, a structure planning module integrating SDP (planning), SAWC (scheduling), and MGSR (dynamic replanning); (3) **CaM-Writing** for citation-aware subsection generation. Final refinement and table generation are performed after writing. Memory $\mathcal{M}$ accumulates writing content and terminology across stages to guide planning and ensure consistency.

Section 3.2.1). The final **global literature set** $\mathcal{P}_i^*$ used for writing each subsection is the combination of the survey-level set $\mathcal{P}^*$ and the subsection-level set $\mathcal{P}_i$. This paper set captures both the global scope of the survey and the specific focus of each subsection.

3.2 PlanEvo: Structure Planning with Dynamic Outline Evolution

We introduce **PlanEvo**, a planning-centric framework for scalable and coherent survey outline generation and writing plan construction. PlanEvo consists of three tightly integrated components: the Structure-Driven Planner (SDP), the Structure-Aware Writing Controller (SAWC), and the Memory-Guided Structure Replanner (MGSR). Detailed designs for each component are presented in Section 3.2.1, Section 3.2.2, and Section 3.2.3.

3.2.1 SDP: Structure-Driven Planner

The SDP module serves as the entry point of PlanEvo, transforming a specific research topic (T, E) into a structured, executable plan that guides the full survey generation process. The overall workflow is shown in Figure 2.

Reference-Grounded Outline Generation. A literature-grounded outline is essential for generating a coherent and well-structured survey. To build such an outline, the system first identifies review articles within the survey-level literature set $\mathcal{P}^*$ by analyzing metadata such as publication type. The structural outlines of these reviews $\mathcal{R}$ are then extracted from their full texts using LLMs and used as representative structural patterns to inspire the design of new outlines. The system then collects titles and abstracts of non-review papers in $\mathcal{P}^*$ to form the abstract-level content set $\mathcal{C}_{\text{abs}}$, which is then combined with $\mathcal{R}$ into a composite context $\mathcal{C}$. Given $\mathcal{C}$ and (T, E), an LLM is prompted to generate an initial outline $\mathcal{O}_0$:

$$\mathcal{O}_0 = \{(s_i, d_i)\}_{i=1}^{N} \quad (2)$$

where each subsection heading s_i is paired with a brief description d_i to provide more detailed guidance for writing subsections. To improve coherence and reduce redundancy, the initial outline $\mathcal{O}_0$ is refined by an LLM, obtaining the final outline $\mathcal{O}$.

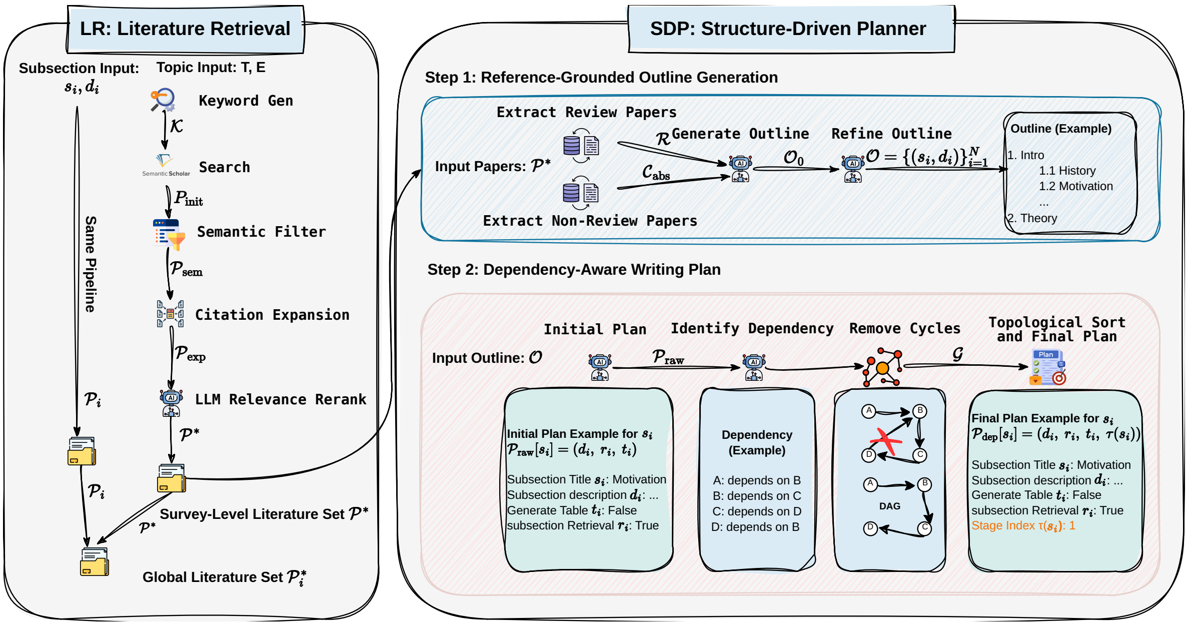

LR: Literature Retrieval
Subsection Input: s_i, d_i
Topic Input: T, E
Keyword Gen
$\mathcal{K}$
Semantic Scholar
Search
$\mathcal{P}_{\text{init}}$
Semantic Filter
$\mathcal{P}_{\text{sem}}$
Citation Expansion
$\mathcal{P}_{\text{exp}}$
LLM Relevance Rerank
$\mathcal{P}^*$
Same Pipeline
$\mathcal{P}_i$
$\mathcal{P}_i$
$\mathcal{P}^*$
Survey-Level Literature Set $\mathcal{P}^*$
Global Literature Set $\mathcal{P}_i^*$
SDP: Structure-Driven Planner
Step 1: Reference-Grounded Outline Generation
Extract Review Papers
Input Papers: $\mathcal{P}^*$
Extract Non-Review Papers
$\mathcal{R}$
$\mathcal{C}_{\text{abs}}$
Generate Outline
$\mathcal{O}_0$
Refine Outline
$\mathcal{O} = \{(s_i, d_i)\}_{i=1}^N$
Outline (Example)
1. Intro
1.1 History
1.2 Motivation
...
2. Theory
Step 2: Dependency-Aware Writing Plan
Input Outline: $\mathcal{O}$
Initial Plan
$\mathcal{P}_{\text{raw}}$
Identify Dependency
Remove Cycles
$\mathcal{G}$
Topological Sort and Final Plan
Plan
Initial Plan Example for s_i
$\mathcal{P}_{\text{raw}}[s_i] = (d_i,\ r_i,\ t_i)$
Subsection Title s_i: Motivation
Subsection description d_i: ...
Generate Table t_i: False
subsection Retrieval r_i: True
Dependency (Example)
A: depends on B
B: depends on C
C: depends on D
D: depends on B
A
B
C
D
DAG
Final Plan Example for s_i
$\mathcal{P}_{\text{dep}}[s_i] = (d_i,\ r_i,\ t_i,\ \tau(s_i))$
Subsection Title s_i: Motivation
Subsection description d_i: ...
Generate Table t_i: False
subsection Retrieval r_i: True
Stage Index $\tau(s_i)$: 1

Figure 2: Details of the **Literature Retrieval** and **Structure-Driven Planner** components in *SurveyGen-I*.

An example of the generated outline is provided in Appendix B.2.

Dependency-Aware Writing Plan. To enable logically coherent and coordinated writing across subsections, we construct a dependency-aware writing plan $\mathcal{P}_{\text{dep}}$ based on the survey outline $\mathcal{O}$.

First, an initial plan $\mathcal{P}_{\text{raw}}$ is generated by prompting an LLM with $\mathcal{O}$. For each subsection s_i with its description d_i, the plan specifies two control signals: whether additional literature retrieval is required (r_i), and whether a comparative table should be generated (t_i). These signals guide downstream tasks in subsection-level literature retrieval and table generation. Each subsection takes the form:

$$\mathcal{P}_{\text{raw}}[s_i] = (d_i,\ r_i,\ t_i), \tag{3}$$

Next, to ensure logical dependency alignment across subsections, we construct a structural dependency graph $\mathcal{G}_{\text{raw}} = (\mathcal{V}, \mathcal{E})$. Each node $s_i \in \mathcal{V}$ represents a subsection, and a directed edge $(s_i \to s_j) \in \mathcal{E}$ indicates that subsection s_i is an essential prerequisite for writing s_j.

To infer these prerequisite relations, we prompt an LLM to analyze the outline $\mathcal{O}$ and identify, for each subsection, which earlier subsections it depends on (see prompt in Figure 16). The LLM assigns a dependency score from 1 to 5, where 5 denotes that the prior subsection is absolutely essential for writing the current one. Only dependencies with a score of 5 are retained as edges in $\mathcal{E}$. If cycles are detected, we remove one edge per cycle, obtaining a Directed Acyclic Graph (DAG) $\mathcal{G}$ that captures the constraints among subsections.

Based on this dependency graph $\mathcal{G}$, we derive the writing order of all subsections with topological sorting. Each subsection s is assigned a stage index $\tau(s)$, representing the depth of its dependency chain:

$$\tau(s) = \begin{cases} 0 & \text{if } \mathrm{In}(s) = \emptyset, \\ \max\limits_{s' \in \mathrm{In}(s)} \tau(s') + 1 & \text{otherwise.} \end{cases} \tag{4}$$

Finally, the dependency-aware plan $\mathcal{P}_{\text{dep}}$ extends $\mathcal{P}_{\text{raw}}$ by attaching the stage index $\tau(s_i)$ to each subsection:

$$\mathcal{P}_{\text{dep}}[s_i] = (d_i,\ r_i,\ t_i,\ \tau(s_i)). \tag{5}$$

Subsections assigned the same stage index can be written in parallel, while respecting the dependency constraints imposed by $\mathcal{G}$. An example of the generated dependency-aware writing plan is provided in Appendix B.2.

3.2.2 SAWC: Structure-Aware Writing Controller

The SAWC module serves as the central orchestration engine across the entire writing process in the *SurveyGen-I* pipeline. Rather than being a single step, SAWC coordinates a sequence of interdependent modules, including writing stage scheduling (Step 4), subsection-level literature retrieval (Step

5), citation-aware writing (Step 6), memory updating (Step 7), dynamic structure replanning (Steps 9–10), final refinement (Step 11), and table generation (Step 12). Its control flow is illustrated throughout the center path of Figure 1.

Parallel Subsection Execution. SAWC executes the dependency-aware writing plan $\mathcal{P}$dep by activating all subsections with the same writing stage index $\tau(s_i)$ in parallel (Step 4). For each active subsection s_i, SAWC first checks the retrieval control flag r_i in $\mathcal{P}_{\text{dep}}[s_i]$. If retrieval is required, SAWC triggers subsection-level literature retrieval (Step 5; see Section 3.1.2). The resulting paper set $\mathcal{P}_i^*$ is passed to the writing module (Step 6; see Section 3.3) for citation-aware subsection generation.

Memory Mechanism for Global Consistency. To ensure structural coherence and terminological consistency across the survey, SAWC maintains a dynamic **structure memory** $\mathcal{M}$ throughout writing. After each subsection s_i is written (Step 6), the system extracts key domain-specific terminology using LLMs, and stores both the terminology and the draft content into $\mathcal{M}$ (Step 7). This accumulated memory is then used to (1) guide subsequent subsection writing by enforcing consistency (see Section 3.3.2), and (2) provide feedback for dynamic updates of the outline $\mathcal{O}$ and writing plan $\mathcal{P}_{\text{dep}}$ during structure replanning (see Section 3.2.3).

Dynamic Structure Refinement. At the end of each writing stage, which corresponds to the completion of all subsections s_i with the same stage index $\tau(s_i)$, SAWC triggers the Memory-Guided Structure Replanner (MGSR; see Section 3.2.3) to revise the outline and the writing plan based on the accumulated memory $\mathcal{M}$ (Step 9–10). This stage-wise feedback loop ensures that structural adjustments are continuously informed by prior writing outputs before the next stage begins.

Final Refinement and Table Generation. After all subsections are written, SAWC performs a final refinement step to improve global coherence (Step 11). An LLM analyzes the full draft to detect logical contradictions, redundancy, and terminological/style inconsistencies. Based on this diagnosis, the system rewrites the relevant subsections to ensure consistency. Then, for each subsection with the table flag t_i enabled in $\mathcal{P}_{\text{dep}}[s_i]$, SAWC generates a structured table based on the retrieved paper set (Step 12). See Appendix D for details.

3.2.3 MGSR: Memory-Guided Structure Replanner

After each writing stage, the MGSR module performs dynamic refinement of the outline and writing plan based on the accumulated memory $\mathcal{M}$ and the current outline $\mathcal{O}$. MGSR prompts an LLM (see prompt in Figure 15) to analyze redundancy, missing conceptual gaps, or suboptimal ordering within the unwritten subsections. It produces a set of structured revision actions (merge, delete, rename, reorder, add) applied to the remaining outline. The updated writing plan $\mathcal{P}'_{\text{dep}}$ is then derived from the revised outline $\mathcal{O}'$ through the same planning method used in the initial plan $\mathcal{P}_{\text{dep}}$ (see Section 3.2.1). This enables memory-guided structural evolution throughout writing, ensuring that later sections are adaptively optimized based on prior content while maintaining global consistency.

3.3 CaM-Writing: Citation-Aware Subsection Writing with Memory Guidance

This section introduces **CaM-Writing**, a citation-aware, memory-guided writing pipeline for generating each survey subsection. The pipeline integrates a citation-tracing mechanism to enhance literature coverage and citation diversity, skeleton-based generation guided by the memory $\mathcal{M}$ to ensure content consistency, and multi-stage refinement to improve clarity, coherence, and citation integrity.

3.3.1 Context Construction with Citation Tracing

To construct a rich and contextually relevant evidence set for writing each subsection s_i with description d_i, a RAG step is first applied over the global literature set $\mathcal{P}_i^*$, which includes both survey-level and subsection-specific literature. Top-ranked passages are selected to form the initial writing context $\mathcal{C}_{\text{rag},i}$. However, the retrieved passages from academic papers often contain indirect citations such as "[23]" and "Ge et al., 2023". These citations typically refer to influential prior work that is not directly included in the retrieved literature. If the system relies solely on these secondary mentions without further resolution, it may overlook foundational or highly relevant papers.

To address this issue, we introduce a **citation-tracing mechanism** that automatically identifies and resolves indirect citations within each retrieved context set $\mathcal{C}rag, i$. Specifically, the mechanism detects citation markers such as “[3]” or “(Smith et al., 2020)” in $\mathcal{C}rag, i$ and employs an LLM to determine whether each refers to an original source of a key concept or result. The tracing process is confined to the source papers of the top-ranked passages in $\mathcal{C}rag, i$. For each of these papers, we retrieve structured reference metadata, including stable identifiers, author lists, titles, publication years, and abstracts, using the Semantic Scholar API. Markers identified as relevant are labeled as *primary-source markers* and aligned with the corresponding bibliography entries of their source papers by matching author names, titles, and publication years. Successfully matched references are resolved through the API, and their abstracts are appended to $\mathcal{C}rag, i$ to construct the citation-enriched context $\mathcal{C}_{\text{enrich},i}$. Unmatched markers are safely ignored to avoid error propagation.

To maintain explicit traceability between indirect citations and their original source, each enriched paper's abstract is linked back to the original passage where its citation marker appears, allowing the writing model to reason about the origin and relevance. For example, consider the passage:

> "... recent work has introduced reward-balanced fine-tuning for alignment (Ge et al., 2023), showing improvements over DPO and RLHF ..."

The LLM flags "(Ge et al., 2023)" as *primary-source marker*, identifying it as introducing a core method. The system then resolves it to:

> **Title:** Preserve Your Own Correlation: A New Reward-Balanced Fine-Tuning Method
> **Abstract:** 'We introduce a reward-balanced fine-tuning (RBF) framework for language model alignment..."

This abstract is appended to the context, enabling the system to cite this traced paper directly in the generation. Further details and a citation-enriched context example are provided in Appendix E.

3.3.2 Memory-Aligned Skeleton-Guided Generation

Given the enriched context $\mathcal{C}_{\text{enrich},i}$, subsection title and description (t_i, d_i), and the accumulated structure memory $\mathcal{M}$, the system first uses an LLM (see prompt in Figure 17) to generate a writing skeleton $\mathcal{S}_i$ outlining the key conceptual points. The memory $\mathcal{M}$, which includes prior subsections and extracted terminology, ensures content coherence and terminology consistency across the survey.

Best-of-N Selection. N candidate drafts are first generated based on the subsection title s_i, description d_i, writing skeleton $\mathcal{S}_i$, and enriched context $\mathcal{C}_{\text{enrich},i}$ (see prompt in Figure 18). An LLM then evaluates the candidates and selects the best version based on alignment with the skeleton, contextual relevance, and overall writing quality.

Subsection-Level Refinement. To further improve the selected draft, a three-stage refinement is applied. First, the structure is adjusted to better reflect the conceptual flow defined by $\mathcal{S}i$. Second, the draft undergoes citation refinement, where the LLM rewrites the text based on $\mathcal{C}_{\text{enrich},i}$. Finally, the draft is polished to enhance fluency and clarity.

4 Experiments and Results

4.1 Evaluation Setup

SurveyGen-I is evaluated across six major scientific domains, which jointly cover both theoretical and applied areas of AI. Detailed topic distribution is provided in Appendix H.

We compare *SurveyGen-I* with three representative baselines: AutoSurvey (Wang et al., 2024), SurveyForge (Yan et al., 2025), and SurveyX (Liang et al., 2025). Demo reports from SurveyForge[1] and SurveyX[2] are collected from their official project pages. For SurveyForge, we select reports on exactly matched topics, while for SurveyX, domain-level matching is applied due to differences in topic coverage. Reports for AutoSurvey are generated on matched topics using the same model as *SurveyGen-I* (GPT4o-mini (OpenAI, 2024)) to ensure fair comparison.

4.2 Evaluation Metrics

We comprehensively evaluate *SurveyGen-I* against three competitive baselines across two core dimensions, content quality and reference quality.

Content Quality Evaluation. Measures the structural and semantic strength of the generated survey. This includes five sub-dimensions: *coverage, relevance, structure, synthesis, and consistency*. Each aspect is scored by LLM-as-Judge (Li et al., 2024a) (specifically, rated by GPT4o-mini), with explanation-based prompts to reduce variance. This directly reflects the impact of our MGSR and

[1] SurveyForge: https://github.com/Alpha-Innovator/SurveyForge

[2] SurveyX: https://github.com/IAAR-Shanghai/SurveyX

CaM-Writing, which aim to improve global coherence, abstraction, and flow. Evaluation criteria can be found in Appendix N. We compute the final content quality score (CQS) as the average of five evaluation dimensions.

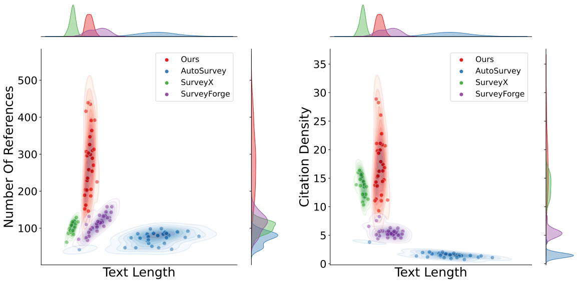
Reference Quality Evaluation. To assess the effectiveness and recency of reference usage in the generated survey, we adopt three reference-level metrics that reflect citation coverage, intensity, and timeliness. The Number of References (NR) counts the distinct cited works, measuring the breadth of literature coverage. The Citation Density (CD) computes the number of unique citation markers per character of text (excluding the reference section), reflecting how frequently references are integrated into the main narrative. For reporting clarity, we scale CD by a factor of 10^4. The Recency Ratio (RR@k) measures the proportion of all cited references that were published within a recent time window (e.g., within the past k=3 years). A higher RR indicates better engagement with the latest developments in the field, and reflects the model's ability to retrieve and integrate timely literature.

4.3 Main Results

We report evaluation results across content quality and reference behavior, along with an ablation-based component analysis. *SurveyGen-I* is compared against three state-of-the-art baselines: AutoSurvey (Wang et al., 2024), SurveyX (Liang et al., 2025), and SurveyForge (Yan et al., 2025). Our results demonstrate that *SurveyGen-I* achieves significant improvements across all dimensions, showing its effectiveness for automated survey generation.

To further validate the findings and evaluation protocol, we have conducted an additional human evaluation, setups and results are in Appendix K.

Content Quality. *SurveyGen-I* achieves consistent improvements across all five content quality dimensions compared to prior systems (Table 1). The content quality score (CQS) reaches 4.59, outperforming the best baseline (SurveyForge: 4.23). Largest gains are observed in structural flow (STRUC: +0.21) and synthesis (SYN: +0.41), indicating that the model maintains a coherent narrative while integrating information from diverse sources. Coverage (4.72) and relevance (4.76) also lead all baselines, suggesting high topical breadth and alignment. Consistency (4.59) improves notably over SurveyX (4.29), reflecting stability in terminology and phrasing across sections. Notably, SurveyX uses GPT4o (Hurst et al., 2024), whereas our system relies on a smaller and more cost-efficient model, making the performance gap especially significant. Observed quality gains suggest that systems combining structural adaptivity, iterative refinement, and citation-tracing can more reliably generate high-quality surveys.



(a) Ref count vs. length

(b) Citation density

Figure 3: Citation quality comparisons across models using KDE-enhanced scatter plots. **(a)** Number of references vs. text length. *SurveyGen-I* demonstrates a steeper citation scaling curve, suggesting deeper integration of references even in longer texts. **(b)** Citation density vs. text length. *SurveyGen-I* maintains denser citation patterns across all lengths.

Model	CQS ↑	COV ↑	REL ↑	STRUC ↑	SYN ↑	CONSIS ↑
AutoSurvey	4.08	4.10	4.17	4.03	4.10	4.00
SurveyX	4.13	4.10	4.33	4.00	3.95	4.29
SurveyForge	4.23	4.31	4.41	4.07	4.21	4.17
Ours	**4.59**	**4.72**	**4.76**	**4.28**	**4.62**	**4.59**

Table 1: LLM-based evaluation scores across multiple survey quality dimensions. Higher is better for coverage (COV), relevance (REL), structural flow (STRUC), synthesis (SYN), and consistency (CONSIS).

Reference Quality. In terms of citation quality and scientific grounding, *SurveyGen-I* exhibits both broader and denser reference usage. It cites 281 unique works per survey on average (Table 2), representing a sharp increase over SurveyX (102), and AutoSurvey (73). Citation density also rises substantially (17.28), exceeding SurveyForge (5.52) by around 3 times, indicating tighter integration of references into the body text. Importantly, 89.1% of all citations are published within the past 5 years (RR@5), compared to 66.7% in SurveyX and SurveyForge, demonstrating significantly improved recency alignment.

The steep scaling trend between reference count and text length in Figure 3a shows that text length remains relatively stable in *SurveyGen-I*, reflecting the fixed-length constraint imposed during generation. It also demonstrates that *SurveyGen-I* includes the most references overall. In contrast, AutoSurvey consistently generates fewer references, and its citation count remains relatively flat, even as its text length slightly increases, which is unexpected given that the length parameter was controlled across all generations. This suggests weaker responsiveness to contextual expansion and underutilization of available content space. Figure 3b further shows that *SurveyGen-I* consistently maintains a high citation density across varying text lengths, indicating robust integration of information-dense content.

Model	RR@1	RR@3	RR@5	RR@7	RR@10	CD	NR
AutoSurvey	0.174	0.639	0.837	0.940	**0.992**	1.54	73
SurveyX	0.239	0.484	0.667	0.792	0.916	13.57	102
SurveyForge	0.137	0.437	0.667	0.824	0.907	5.52	113
Ours	**0.478**	**0.759**	**0.891**	**0.955**	0.985	**17.28**	**281**

Table 2: Recency-focused and structural citation metrics. RR@k measures the proportion of cited references published within the past k years. CD (scaled by $\times 10^4$) measures citation density; NR is the total number of cited references.

A fine-grained evaluation for citation recency across six domains is provided in Appendix I.

4.4 Ablation and Further Analysis

Ablation Studies. We conduct ablation studies on the Language Models domain to evaluate the contribution of three key mechanisms: (1) **w/o Citation Trace** disables the citation-tracing mechanism (see Section 3.3.1); (2) **w/o Plan Update** disables the MGSR (Memory-Guided Structure Replanner) module (see Section 3.2.3), fixing the outline and writing plan without dynamic adjustment as new subsections are generated; and (3) **w/o Final Refinement** removes the final-stage multi-pass refinement phase (see Section 3.2.2).

Model	CQS ↑	COV ↑	REL ↑	STRUC ↑	SYN ↑	CONSIS ↑	NR ↑
Ours (w/o Citation Trace)	4.60	4.57	4.57	4.43	4.71	**4.71**	225
Ours (w/o Plan Update)	4.49	4.57	4.71	4.29	4.43	4.43	212
Ours (w/o Refine)	4.34	4.43	4.43	4.14	4.43	4.29	**286**
Ours (Full)	**4.77**	**4.71**	**4.86**	**4.71**	**4.86**	**4.71**	**286**

Table 3: Evaluation results of ablation variants. Each component (Trace, Plan Update, Refine) contributes to overall quality.

Table 3 reports the impact of removing specific behaviors from *SurveyGen-I*. The full model yields the highest CQS (4.77), with synthesis and structure both at 4.86 and 4.71, respectively. Disabling final refinement results in the steepest quality drop (CQS: -0.43), particularly in synthesis (-0.43) and structure (-0.57), indicating that single-pass generation without revision is insufficient for maintaining narrative integration. Fixed planning further reduces structural flow (STRUC: -0.42) and consistency (CONSIS: -0.28), suggesting that static outlines limit the model's ability to adjust to unfolding content. Removing citation resolution reduces the number of distinct references by 61, despite stable consistency.

Effect of Citation Tracing Mechanism. To further analyze the role of the citation tracing mechanism, we quantify the number of indirect references traced during context construction. Across 121 subsections from four surveys generated by *SurveyGen-I*, an average of 8.43 indirect references per subsection were successfully traced. The proportion of new references introduced by tracing is summarized in Table 4.

Proportion Range (%)	# Subsections	Percentage of Total (%)
0	27	22.3
0–20	16	13.2
20–40	27	22.3
40–60	32	26.4
60–80	16	13.2
80–100	3	2.5
Total	**121**	**100.0**

Table 4: Proportion of newly introduced references per subsection after citation tracing (first column).

Effect of Memory Length and Context Length. We further examine the token length of enriched contexts and memory used during subsection generation. Across 121 subsections, the citation-enriched contexts $\mathcal{C}_{enrich,i}$ contain an average of 4.6k tokens, with a maximum length of approximately 21k tokens, while the memory $\mathcal{M}$ used for skeleton generation averages 13.4k tokens, reaching up to 30k tokens. Both remain well within the 128k-token input limit of GPT4o-mini.

5 Conclusion

We present *SurveyGen-I*, a fully automated framework for generating academic surveys with high consistency, citation coverage, and structural coherence. By integrating coarse-to-fine retrieval, adaptive planning, and memory-guided writing, *SurveyGen-I* effectively captures complex literature landscapes and produces high-quality surveys without manual intervention. Extensive evaluations demonstrate its effectiveness over existing methods, marking a step forward in reliable and scalable scientific synthesis.

Limitations

While *SurveyGen-I* shows consistently strong performance across benchmarks, our framework adopts an online retrieval strategy to ensure access to up-to-date literature. However, this design introduces network sensitivity, variable latency, and reliance on third-party APIs, which may restrict full-text access due to licensing constraints. Compared to offline-indexed corpora used in prior work, our approach trades retrieval speed and infrastructure control for broader coverage and freshness.

Additionally, for niche or emerging topics with limited source material, the achievable survey length and depth are naturally constrained. This shows a general challenge in automatic survey generation: content quality is ultimately bounded by the availability and granularity of the source literature. Moreover, some evaluation signals may reflect subjective preferences rather than universal writing standards. Our current evaluation primarily focuses on AI-related domains, which provides a strong but domain-specific validation. Future work will expand evaluation to broader scientific areas to test generalizability across disciplines.

Future work may focus on minimizing external dependencies and enabling more local processing by developing modular, domain-specialized variants of the framework, where different agents are supported by parameter-efficient, locally fine-tuned models (Shen et al., 2025; Huang et al., 2025). Complementary reinforcement-learning-based methods (Sun et al., 2024; Jiang et al., 2025a) could further enhance adaptive control and synthesis stability.

Ethics Statement

This research focuses on developing a transparent and responsible framework for automated survey generation. *SurveyGen-I* is intended solely as a research and writing assistant to support scholars in organizing, retrieving, and synthesizing literature. It does not aim to produce publishable manuscripts without human validation. Human users must carefully review and verify all generated content before any academic use.

Our experiments use only open-access scientific literature and publicly available benchmarks. No personal, sensitive, or proprietary data were used. We acknowledge that automated generation may still produce incomplete or biased representations of prior work, and therefore encourage human verification in any downstream use.

Acknowledgement

This work was supported by the Dutch National Growth Fund through the 6G flagship project "Future Network Services" (Grant Number: NGF6G2301).

References

Shubham Agarwal, Gaurav Sahu, Abhay Puri, Issam H Laradji, Krishnamurthy DJ Dvijotham, Jason Stanley, Laurent Charlin, and Christopher Pal. 2024. Litllm: A toolkit for scientific literature review. *arXiv preprint arXiv:2402.01788*.

Shubham Agarwal, Gaurav Sahu, Abhay Puri, Issam H. Laradji, Krishnamurthy Dj Dvijotham, Jason Stanley, Laurent Charlin, and Christopher Pal. 2025. LitLLMs, LLMs for literature review: Are we there yet? *Transactions on Machine Learning Research*.

Nurshat Fateh Ali, Md. Mahdi Mohtasim, Shakil Mosharrof, and T. Gopi Krishna. 2024. Automated literature review using nlp techniques and llm-based retrieval-augmented generation. In *2024 International Conference on Innovations in Science, Engineering and Technology*, pages 1–6.

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, and 1 others. 2018. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 84–91.

Hongye An, Arpit Narechania, Emily Wall, and Kai Xu. 2024. Vitality 2: Reviewing academic literature using large language models. *arXiv preprint arXiv:2408.13450*.

Artifex. 2025. PyMuPDF 1.25.5 documentation. Accessed: 2025-05-20.

David Brett and Anniek Myatt. 2025. Patience is all you need! an agentic system for performing scientific literature review. *arXiv preprint arXiv:2504.08752*.

Angela Carrera-Rivera, William Ochoa, Felix Larrinaga, and Ganix Lasa. 2022. How-to conduct a systematic literature review: A quick guide for computer science research. *MethodsX*, 9:101895.

Ruediger Ehlers. 2017. Formal verification of piece-wise linear feed-forward neural networks. In *International symposium on automated technology for verification and analysis*, pages 269–286.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Raymond Fok, Joseph Chee Chang, Marissa Radensky, Pao Siangliulue, Jonathan Bragg, Amy X Zhang, and Daniel S Weld. 2025. Facets, taxonomies, and syntheses: Navigating structured representations in llm-assisted literature review. *arXiv preprint arXiv:2504.18496*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *Coling 2010: Posters*, pages 427–435.

Yue Hu and Xiaojun Wan. 2014. Automatic generation of related work sections in scientific papers: an optimization approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1624–1633.

Jia-Hong Huang, Yixian Shen, Hongyi Zhu, Stevan Rudinac, and Evangelos Kanoulas. 2025. Gradient weight-normalized low-rank projection for efficient llm training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24123–24131.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Xia Jiang, Yaoxin Wu, Minshuo Li, Zhiguang Cao, and Yingqian Zhang. 2025a. Large language models as end-to-end combinatorial optimization solvers. *arXiv preprint arXiv:2509.16865*.

Xia Jiang, Yaoxin Wu, Chenhao Zhang, and Yingqian Zhang. 2025b. DRoc: Elevating large language models for complex vehicle routing via decomposed retrieval of constraints. In *The 13th International Conference on Learning Representations*.

Tetsu Kasanishi, Masaru Isonuma, Junichiro Mori, and Ichiro Sakata. 2023. SciReviewGen: A large-scale dataset for automatic literature review generation. In *Findings of the Association for Computational Linguistics*, pages 6695–6715.

Guy Katz, Derek A Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, and 1 others. 2019. The marabou framework for verification and analysis of deep neural networks. In *International Conference on Computer Aided Verification*, pages 443–452.

Yuxuan Lai, Yupeng Wu, Yidan Wang, Wenpeng Hu, and Chen Zheng. 2024. Instruct large language models to generate scientific literature survey step by step. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 484–496.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024a. LLMs-as-judges: a comprehensive survey on LLM-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, and 1 others. 2024b. Chain of ideas: Revolutionizing research via novel idea development with llm agents. *arXiv preprint arXiv:2410.13185*.

Yutong Li, Lu Chen, Aiwei Liu, Kai Yu, and Lijie Wen. 2025. Chatcite: Llm agent with human workflow guidance for comparative literature summary. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3613–3630.

Xun Liang, Jiawei Yang, Yezhaohui Wang, Chen Tang, Zifan Zheng, Shichao Song, Zehao Lin, Yebin Yang, Simin Niu, Hanyu Wang, and 1 others. 2025. Surveyx: Academic survey automation via large language models. *arXiv preprint arXiv:2502.14776*.

Jiaxiang Liu, Yunhan Xing, Xiaomu Shi, Fu Song, Zhiwu Xu, and Zhong Ming. 2024. Abstraction and refinement: towards scalable and exact verification of neural networks. *ACM Transactions on Software Engineering and Methodology*, 33(5):1–35.

Hidetsugu Nanba, Noriko Kando, and Manabu Okumura. 2000. Classification of research papers using citation links and citation types: Towards automatic review article generation. *Advances in Classification Research Online*, pages 117–134.

Benjamin Newman, Yoonjoo Lee, Aakanksha Naik, Pao Siangliulue, Raymond Fok, Juho Kim, Daniel S Weld, Joseph Chee Chang, and Kyle Lo. 2024. ArxivDIGESTables: Synthesizing scientific literature into tables using language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9612–9631.

Zach Nussbaum, John Xavier Morris, Andriy Mulyar, and Brandon Duderstadt. 2025. Nomic embed: Training a reproducible long context text embedder.

Transactions on Machine Learning Research. Reproducibility Certification.

OpenAI. 2024. Gpt-4o mini: Advancing cost-efficient intelligence. Accessed: 2025-05-20.

Eugênio Piveta Pozzobon and Humberto Pinheiro. 2024. Implementation of a research assistant for literature review with large language models. In *16th Seminar on Power Electronics and Control*, pages 1–5. IEEE.

Pouria Rouzrokh and Moein Shariatnia. 2025. Lattereview: A multi-agent framework for systematic review automation using large language models. *arXiv preprint arXiv:2501.05468*.

Abdul Malik Sami, Zeeshan Rasheed, Kai-Kristian Kemell, Muhammad Waseem, Terhi Kilamo, Mika Saari, Anh Nguyen Duc, Kari Systä, and Pekka Abrahamsson. 2024. System for systematic literature review using multiple ai agents: Concept and an empirical evaluation. *arXiv preprint arXiv:2403.08399*.

Yixian Shen, Qi Bi, Jia-hong Huang, Hongyi Zhu, Andy D. Pimentel, and Anuj Pathania. 2025. MaCP: Minimal yet mighty adaptation via hierarchical cosine projection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20602–20618, Vienna, Austria. Association for Computational Linguistics.

Kartik Shinde, Trinita Roy, and Tirthankar Ghosal. 2022. An extractive-abstractive approach for multi-document summarization of scientific articles for literature review. In *Proceedings of the 3rd Workshop on Scholarly Document Processing*, pages 204–209.

Amanpreet Singh, Joseph Chee Chang, Chloe Anastasiades, Dany Haddad, Aakanksha Naik, Amber Tanaka, Angele Zamarron, Cecile Nguyen, Jena D Hwang, Jason Dunkleberger, and 1 others. 2025. Ai2 scholar qa: Organized literature synthesis with attribution. *arXiv preprint arXiv:2504.10861*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867.

Chuanneng Sun, Songjun Huang, and Dario Pompili. 2024. Llm-based multi-agent reinforcement learning: Current and future directions. *arXiv preprint arXiv:2405.11106*.

Xiaoping Sun and Hai Zhuge. 2019. Automatic generation of survey paper based on template tree. In *15th International Conference on Semantics, Knowledge and Grids*, pages 89–96. IEEE.

Teo Susnjak, Peter Hwang, Napoleon Reyes, Andre LC Barczak, Timothy McIntosh, and Surangika Ranathunga. 2025. Automating research synthesis with domain-specific large language model fine-tuning. *ACM Transactions on Knowledge Discovery from Data*, 19(3):1–39.

Jie Wang, Chengzhi Zhang, Mengying Zhang, and Sanhong Deng. 2018. CitationAS: A tool of automatic survey generation based on citation content. *J. Data Inf. Sci.*, 3(2):20–37.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Qingsong Wen, Wei Ye, and 1 others. 2024. Autosurvey: Large language models can automatically write surveys. *Advances in Neural Information Processing Systems*, 37:115119–115145.

Chang Xiao. 2023. Autosurveygpt: Gpt-enhanced automated literature discovery. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–3.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

Xiangchao Yan, Shiyang Feng, Jiakang Yuan, Renqiu Xia, Bin Wang, Lei Bai, and Bo Zhang. 2025. SURVEYFORGE : On the outline heuristics, memory-driven generation, and multi-dimensional evaluation for automated survey writing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12444–12465.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Arik. 2024. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237.

Appendix Contents

A SurveyGen-I Pipeline

B SurveyGen-I Implementation Details

B.1 Implementation of Literature Retrieval

Our literature retrieval system differentiates between survey-level and subsection-level retrieval.

At the survey level, retrieval is guided by the user-specified topic and its explanatory text, which are refined through an LLM-based intent clarification step to produce a focused query representing the overall survey scope. At the subsection level, retrieval uses the section and subsection titles and descriptions as input, enabling the LLM to disambiguate fine-grained intent within the broader survey context.

For citation expansion, citation-based expansion is performed only for the top 10 papers with higher similarity score, where their references and citations are retrieved to enrich topic coverage.

For filtering, we adopt a two-stage relevance selection strategy after citation expansion. First, all retrieved papers are pre-filtered by cosine similarity (threshold = 0.3) using `all-mpnet-base-v2` embeddings (Song et al., 2020) to remove semantically irrelevant results. Second, the remaining papers are evaluated by an LLM that assigns a relevance score in the range [0, 100]; only papers with scores $\geq$ 70 are retained. These thresholds were chosen empirically to balance precision and recall across diverse topics. If no papers meet the relevance threshold, a fallback selects the top 5 most relevant candidates to ensure writing continuity. To control downstream computational cost, the number of retained papers per query is capped at 30.

Algorithm 1 SurveyGen-I Pipeline

```
Require: Research topic $T$, description $E$
    // LR: Coarse-to-Fine Literature Retrieval
    $\mathcal{P}_{\text{init}} \leftarrow \textsc{KeywordSearch}(T, E)$
    $\mathcal{P}_{\text{sem}} \leftarrow \textsc{SemanticFilter}(\mathcal{P}_{\text{init}}, T, E)$
    $\mathcal{P}^* \leftarrow \textsc{ExpandAndRank}(\mathcal{P}_{\text{sem}}, T, E)$
    // PlanEvo: Structure Planning and Scheduling
    $\mathcal{O} \leftarrow \textsc{GenerateOutline}(\mathcal{P}^*, T, E)$
    $\mathcal{P}_{\text{dep}} \leftarrow \textsc{BuildSchedule}(\mathcal{O})$
    $\mathcal{M} \leftarrow \emptyset$
    // CaM-Writing
    for $t = 0$ to $\max_{s_i \in \mathcal{O}} \tau(s_i)$ do
        $\mathcal{S}_t \leftarrow \{s_i \in \mathcal{O} \mid \tau(s_i) = t\}$
        for all $s_i \in \mathcal{S}_t$ do
            $(d_i, r_i, t_i) \leftarrow \mathcal{P}_{\text{dep}}[s_i]$
            if $r_i = \texttt{True}$ then
                $\mathcal{P}_i \leftarrow \textsc{SubsectionRetrieval}(s_i, d_i)$
            else
                $\mathcal{P}_i \leftarrow \emptyset$
            end if
            $\mathcal{P}_i^* \leftarrow \mathcal{P}^* \cup \mathcal{P}_i$
            $\mathcal{C}_{\text{rag},i} \leftarrow \textsc{RAGRetrieve}(s_i, \mathcal{P}_i^*)$
            $\mathcal{C}_{\text{enrich},i} \leftarrow \textsc{CitationTrace}(\mathcal{C}_{\text{rag},i})$
            $\mathcal{S}_i \leftarrow \textsc{MakeSkeleton}(s_i, d_i, \mathcal{M})$
            $\hat{s}_i^{(1:N)} \leftarrow \textsc{Write}(s_i, d_i, \mathcal{S}_i, \mathcal{C}_{\text{enrich},i})$
            $\hat{s}_i^* \leftarrow \textsc{SelectBest}(\hat{s}_i^{(1:N)})$
            $\hat{s}_i \leftarrow \textsc{Refine}(\hat{s}_i^*, \mathcal{S}_i, \mathcal{C}_{\text{enrich},i})$
            $\mathcal{M} \leftarrow \mathcal{M} \cup \textsc{ExtractMemory}(\hat{s}_i)$
        end for
        $(\mathcal{O}', \mathcal{P}'_{\text{dep}}) \leftarrow \textbf{MGSR}.\textsc{UpdatePlan}(\mathcal{O}, \mathcal{P}_{\text{dep}}, \mathcal{M})$
        $\mathcal{O} \leftarrow \mathcal{O}'$;   $\mathcal{P}_{\text{dep}} \leftarrow \mathcal{P}'_{\text{dep}}$
    end for
    $\textsc{GlobalRefine}(\mathcal{O}, \mathcal{M})$
    return Final survey draft
```

All filtered papers are associated with structured metadata, including bibkey, title, abstract, paper ID, and download URL. We download each paper's PDF asynchronously using available metadata from Semantic Scholar. The downloaded PDFs are parsed with PyMuPDF (Artifex, 2025) to extract clean textual content, which is stored alongside the metadata for downstream use in vectorstore construction, RAG retrieval, and citation tracing. All paper metadata and extracted content are persisted in CSV format, and BibTeX entries for retained papers are compiled into a unified reference file for LaTeX formatting.

B.2 Outline and Writing Plan Generation

B.2.1 Survey Outline Example

The following example illustrates the JSON-based outline used to guide planning and writing. It defines high-level sections and their conceptual subsections:

```
{
  "section_title": "Fine-Tuning
      Methodologies for Enhanced
      Translation",
  "section_description": "This section
      categorizes and analyzes various
      fine-tuning methodologies employed
       in multilingual models,
      emphasizing their effectiveness
      for Chinese to Malay translation."
      ,
  "subsections": [
    {
      "subsection_title": "Adaptive Fine
          -Tuning Techniques",
      "subsection_description": "
          Discusses adaptive fine-tuning
           methods, including Layer-
          Freezing and Low-Rank
          Adaptation, and their roles in
           optimizing model performance.
          "
    },
    {
      "subsection_title": "Utilization
          of Adapter Modules",
      "subsection_description": "
          Explores the implementation of
          adapter modules for fine-
          tuning, highlighting their
          efficiency and effectiveness
          in multilingual speech
          translation tasks."
    },
    {
      "subsection_title": "Data
          Augmentation Strategies",
      "subsection_description": "
          Investigates the role of data
          augmentation techniques,
          including code-switching, in
          enhancing model robustness and
           translation quality."
    }
  ]
}
```

B.2.2 Writing Plan Example

The following snippet is a real system-generated writing plan segment for the same section. It includes execution flags, dependency controls, and writing index:

```
[
  {
    "section_title": "Fine-Tuning
        Methodologies for Enhanced
        Translation",
    "subsection_title": "Adaptive Fine-
        Tuning Techniques",
    "subsection_description": "Discusses
         adaptive fine-tuning methods,
        including Layer-Freezing and Low
        -Rank Adaptation, and their
        roles in optimizing model
        performance.",
    "index": 2,
    "trigger_additional_search": true,
    "generate_table": true,
    "depends_on": [
      "Challenges in Data Availability
          and Quality"
    ]
  },
  {
    "section_title": "Fine-Tuning
        Methodologies for Enhanced
        Translation",
    "subsection_title": "Utilization of
        Adapter Modules",
    "subsection_description": "Explores
        the implementation of adapter
        modules for fine-tuning,
        highlighting their efficiency
        and effectiveness in
        multilingual speech translation
        tasks.",
    "index": 3,
    "trigger_additional_search": true,
    "generate_table": true,
    "depends_on": [
      "Adaptive Fine-Tuning Techniques"
    ]
  },
  {
    "section_title": "Fine-Tuning
        Methodologies for Enhanced
        Translation",
    "subsection_title": "Data
        Augmentation Strategies",
    "subsection_description": "
        Investigates the role of data
        augmentation techniques,
        including code-switching, in
        enhancing model robustness and
        translation quality.",
    "index": 3,
    "trigger_additional_search": true,
    "generate_table": true,
    "depends_on": [
      "Challenges in Data Availability
          and Quality",
      "Adaptive Fine-Tuning Techniques"
    ]
  }
]
```

B.2.3 Implementation Notes

The index field determines batch-level parallel writing: subsections with the same index value can be written concurrently. The depends_on field specifies logical dependencies between subsections, indicating which earlier subsections must be completed before the current subsection.

Initial AutoML Survey Outline (Before MGSR Update)

- **Section 1: Introduction** – Historical Evolution of AutoML; Motivations for Automation in Machine Learning.
- **Section 2: Conceptual Foundations of AutoML** – Bi-level Optimization Framework; Search Space Design and Strategy Dynamics; Evaluation Metrics for AutoML Systems.
- **Section 3: Methodological Innovations in AutoML** – Neural Architecture Search (NAS); Hyperparameter Optimization Strategies; Meta-Learning in AutoML.
- **Section 4: Applications of AutoML Across Domains** – Healthcare Applications; Financial Services Applications; Marketing and Customer Analytics.
- **Section 5: Challenges and Limitations of AutoML** – Transparency and Interpretability Challenges; Data Quality and Bias Issues; Scalability and Resource Limitations.
- **Section 6: Emerging Trends in AutoML** – Integration with Cloud Computing; Advancements for Unstructured Data; Ethical Considerations and Governance Frameworks.
- **Section 7: Comparative Analysis of AutoML Frameworks** – Benchmarking Methodologies for AutoML Tools; Usability and Accessibility of Frameworks; Performance Evaluation Across Diverse Datasets.
- **Section 8: Future Directions in AutoML Research** – Advancements in Explainable AI; Cross-Domain Generalization Strategies; Integration with Other AI Paradigms.
- **Section 9: Conclusion** – Recap of Key Insights; Future Research Directions.

Figure 4: Initial AutoML outline before MGSR-based refinement.

The `trigger_additional_search` flag controls whether extra paper retrieval will be performed per subsection. Finally, `generate_table` signals whether this subsection requires automatic generation of a literature table summarizing relevant methods, datasets, or evaluation metrics.

C Case Study: Dynamic Outline Updating

To illustrate the impact of our dynamic outline updating mechanism (MGSR, see Section 3.2.3), we compare the survey outline before and after iterative updates for the generated survey titled "Foundations and Future Directions of Automated Machine Learning: Advancements and Applications." The initial outline (Figure 4) represents the first automatically generated structure, while the final version (Figure 5) shows the result after seven rounds of MGSR-based refinement. The comparison highlights how MGSR improves topic coverage and conceptual organization.

D Table Generation Implementation

D.1 Overview

For each subsection in the writing plan, the system determines whether to generate a table based on the number and diversity of relevant papers. If the number of filtered, non-review papers exceeds a certain threshold (typically 10), a *method aggregation table* is created to group papers into conceptual categories. Otherwise, the system generates an *aspect-based comparison table* that compares papers across a small set of important dimensions.

D.2 Method Aggregation Table

When a subsection contains at least ten papers, the system generates a table that organizes the literature into high-level categories under a common comparative theme. The core aspect, for example, training paradigm, objective, or architecture type is first inferred using an LLM based on the subsection description and paper abstracts. A set of 4–6 concise method categories is then proposed by analyzing the previously written subsection content.

Each paper is assigned to one or more categories using RAG. The system constructs a query from the paper's metadata, retrieves relevant content from the paper's vectorized representation, and uses an LLM to classify the paper accordingly. Categories with fewer than two papers or labeled as "Others" are excluded from the final table.

D.3 Aspect-Based Comparison Table

If a subsection contains fewer than ten papers, the system generates a table comparing these papers across several dimensions, typically three to five, following the method proposed in Newman et al.

Refined AutoML Survey Outline (After 7 Rounds of MGSR Updates)

- **Section 1: Introduction** – Historical Evolution of AutoML; Motivations for Automation in Machine Learning.
- **Section 2: Conceptual Foundations of AutoML** – Bi-level Optimization Framework; Search Space Design and Strategy Dynamics; Evaluation Metrics for AutoML Systems; Robustness and Generalization in AutoML; Theoretical Underpinnings of AutoML.
- **Section 3: Methodological Innovations in AutoML** – Neural Architecture Search (NAS); Hyperparameter Optimization Strategies; Meta-Learning in AutoML; Transfer Learning in AutoML; Automated Feature Engineering Techniques.
- **Section 4: Applications of AutoML Across Domains** – Healthcare Applications; Financial Services Applications; Marketing and Customer Analytics; Emerging Applications in Industry.
- **Section 5: Challenges and Limitations of AutoML** – Transparency and Interpretability Challenges; Data Quality and Bias Issues; Ethical and Social Implications; Scalability and Resource Limitations; Deployment Bottlenecks.
- **Section 6: Emerging Trends in AutoML** – Ethical Considerations and Governance Frameworks; Integration with Cloud Computing; Advancements in Explainable AI; Advancements for Unstructured Data.
- **Section 7: Comparative Analysis of AutoML Frameworks** – Benchmarking Methodologies for AutoML Tools; Usability and Accessibility of Frameworks; Performance Evaluation Across Diverse Datasets; Evaluation of Frameworks for Unstructured Data.
- **Section 8: Future Directions in AutoML Research** – Advancements in Explainable AI; Exploration of Domain-Specific AutoML Frameworks; Research into Bias Detection Mechanisms; Exploration of Ethical AI Practices.
- **Section 9: Conclusion** – Recap of Key Insights; Future Research Directions.

Figure 5: Final AutoML outline after seven MGSR iterations, showing enhanced conceptual coverage and structure depth.

(2024). These aspects are automatically selected by analyzing the subsection's topic, abstract, and the written text of this subsection. For each (paper, aspect) pair, a dedicated query is constructed, and top-ranked passages are retrieved from the paper's content using FAISS-based similarity search and reranking. These snippets are summarized using an LLM into short, informative values. The resulting table presents papers as rows and aspects as columns.

E Details of Citation Tracing for Context Grounding

Subsection-Specific RAG Retrieval and Reranking. Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) combines dense retrieval with generation, enabling models to ground outputs on external knowledge (Jiang et al., 2025b; Fan et al., 2024). Given a subsection title s_i and description d_i, we form a semantic query q_i and retrieve at the passage level. The retrieval corpus is the union of two sources: the survey-level collection $\mathcal{P}^*$ (full-scope) and the subsection-level collection $\mathcal{P}_i$ (specific to s_i). All papers are segmented into textual passages and embedded with `BAAI/bge-small-en-v1.5`; each passage carries metadata (`bibkey`, `title`, `abstract`, `pdf_path`). For each source *separately*, we retrieve the top-K passages by cosine similarity to q_i (top-K from $\mathcal{P}^*$ and top-K from $\mathcal{P}_i$). We then deduplicate across sources to obtain a merged candidate set. To refine semantic precision, we apply a cross-encoder reranker `BAAI/bge-reranker-base` (Xiao et al., 2024) on $(q_i,$ passage$)$ pairs from the merged set and select the top-10 passages by the reranker score. These passages constitute the initial RAG context $\mathcal{C}_{\text{rag},i}$ for subsection s_i.

Citation Marker Detection and Primary-source Markers Decision. Each passage in $\mathcal{C}_{\text{rag},i}$ is scanned for citation markers using regular expressions. The system detects both numeric patterns such as "[17]" and author-year formats such as "(Ge et al., 2023)". These markers are then evaluated for whether they refer to an original contribution worth tracing.

To perform this evaluation, we prompt an LLM with the subsection title, description, and raw passage text. The prompt asks the LLM to identify all citation markers and return a structured assessment for each, including a boolean flag and a natural lan-

Enriched Context Example

Original Document:
Title: Abstraction and Refinement: Towards Scalable and Exact Verification of Neural Networks
Bibkey: Liu2022AbstractionAR

RAG Snippet:

> ... restoring the same amount of accuracy. Our approach is orthogonal to and can be integrated with many existing approaches. For evaluation, we implement our approach as a tool *NARv* using two promising and exact tools *Marabou* [Katz et al., 2019] and *Planet* [Ehlers, 2017] as back-end verification engines.

Reasoning Trace: **(Katz et al., 2019)** in RAG Snippet → **Traced BIBKEY:** Katz2019TheMF — **Title:** The Marabou Framework for Verification and Analysis of Deep Neural Networks. **Abstract:** Deep neural networks are revolutionizing the way complex systems are designed. (summary omitted for brevity).

(Ehlers et al., 2017) in RAG Snippet → **Traced BIBKEY:** Ehlers2017FormalVO — **Title:** Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks. **Abstract:** We present an approach for the verification of feed-forward neural networks in which all nodes have a piece-wise linear activation function. (summary omitted).

Figure 6: Illustration of citation tracing enriching retrieved RAG context.

guage explanation. The output is parsed into a list of dictionaries; only citations flagged as primary source markers are considered for tracing.

Example. Figure 6 illustrates how citation tracing expands the retrieved context by resolving referenced works back to their original sources. The example passage is from Liu et al. (2024), whose inline markers trace to Katz et al. (2019) and Ehlers (2017).

F AutoSurvey Implementation

We follow an open-source AutoSurvey implementation with the generation model set to `gpt-4o-mini-2024-07-18`, the embedding model set to `nomic-ai/nomic-embed-text-v1` (Nussbaum et al., 2025), and retrieval configured with 2500 papers for outline construction. To ensure a fair comparison, the list of topics used for generation is aligned exactly with that of our system.

G Supplementary Experiments: Ai2 Scholar QA

Our task (end-to-end survey generation) optimizes document-scale qualities, including global structure, cross-section consistency, and rich, traceable citations. This differs from long-form scientific QA, which targets query-focused aggregation of evidence. We therefore position a deployed QA system, Ai2 Scholar QA (Singh et al., 2025), as a cross-task reference to contextualize *SurveyGen-I* among user-facing literature tools and to contrast behaviors across settings. We run Ai2 Scholar QA on the Language Models topic in default settings to generate the report, and evaluate it with the identical rubrics used in *SurveyGen-I*.

Content Quality. As shown in Table 5, Ai2 Scholar QA produces generally coherent but less structured reports, achieving an overall content quality score (CQS) of 3.94. While coverage and relevance remain reasonable, synthesis (3.43) and structural flow (4.00) lag behind survey-oriented systems.

Model	CQS ↑	COV ↑	REL ↑	STRUC ↑	SYN ↑	CONSIS ↑
Ai2 Scholar QA	3.94	4.00	4.29	4.00	3.43	4.00
Ours	**4.59**	**4.72**	**4.76**	**4.28**	**4.62**	**4.59**

Table 5: Supplementary content-quality evaluation results for Ai2 Scholar QA.

Reference Quality. Table 6 summarizes reference-based metrics. Although Ai2 Scholar QA exhibits relatively high recency (RR@1–RR@5) and citation density (15.19), the total number of references per survey (35) is significantly smaller.

Model	**RR@1**	**RR@3**	**RR@5**	**RR@7**	**RR@10**	**CD**	**NR**
Ai2 Scholar QA	0.468	0.676	0.773	0.833	0.857	15.19	35
Ours	**0.478**	**0.759**	**0.891**	**0.955**	**0.985**	**17.28**	**281**

Table 6: Supplementary reference-quality metrics for Ai2 Scholar QA.

Overall, Ai2 Scholar QA shows good citation density and recency, but the total number of cited papers is notably limited compared with survey-generation systems.

H Topic Coverage

Table 7 summarizes the benchmark composition used in our experiments. The benchmark spans six major scientific domains, covering a diverse range of research areas. Each domain includes representative survey papers automatically generated by *SurveyGen-I*.

I Subtopic-Level Analysis of Recency Behavior

The impact of different subtopics on survey metrics is mainly reflected in timeliness. Table 8 presents updated recency ratio comparisons across six major subtopics. *SurveyGen-I* maintains consistently strong performance, ranking first in RR@1 across all subtopics, and achieving top-2 results in nearly every other metric. This indicates superior prioritization of the most recent literature compared to all baselines. In particular, our model leads by a large margin on early recency in complex domains such as *AI Applications*, *Vision*, and *Networks*. While AutoSurvey occasionally matches or slightly outperforms on higher-k metrics (e.g., RR@10 in *Databases*), its RR@1 remains substantially lower overall. However, this is partly due to the smaller number of available source papers in that subtopic, which reduces retrieval diversity and makes the evaluation more sensitive to a few recent citations. SurveyForge and SurveyX trail significantly, especially in foundational domains like *Learning Algorithms* and *Big Data*, reflecting weaker temporal grounding.

J Discussion on Metrics and Usage

Note on Metric Usage and Purpose. Our evaluation incorporates several citation-related quantitative metrics, such as recency ratio, and citation density, to benchmark the structural and referential behavior of generated surveys. However, these indicators offer only partial signals of quality. Academic survey writing is inherently diverse and context-dependent: field maturity, topic breadth, and venue-specific formatting constraints (e.g., strict page limits) all shape citation practices and content scope. Moreover, certain high-quality surveys may intentionally limit citations to emphasize synthesis or conceptual framing.

Thus, these tools collectively provide an empirical lens into model behavior, but should not be interpreted as exhaustive or definitive measures of writing quality.

Disclaimer on Intended Use. This tool is developed to assist researchers in efficiently exploring relevant literature and identifying topic structures. The generated content is not intended for direct use in scholarly publication. We cannot guarantee the factual correctness or citation fidelity of all outputs. This limitation is a known challenge in current LLM-based generation systems, especially for tasks involving factual grounding or citation synthesis. Users remain responsible for verifying accuracy, attribution, and appropriateness. We explicitly discourage the direct submission of model-generated text for academic writing or peer-reviewed publication. All outputs require human review, editing, and contextual judgment.

K Human Evaluation Study

To complement the automatic evaluation and examine whether the improvements observed in automatic scores are also reflected in expert judgment along key qualitative dimensions, we conducted a human evaluation.

K.1 Setup and Procedure

A group of graduate-level participants with prior experience in reading and writing survey papers were recruited as evaluators. Each participant independently assessed four anonymized survey papers generated by different systems (AutoSurvey, SurveyForge, SurveyX, and Ours) covering identical research topics and titles.

Participants were asked to rate each survey along five quality dimensions, using a 1–5 Likert scale (1 = very poor, 5 = excellent). The evaluation protocol was delivered via a structured questionnaire, which included both rating and optional comment fields. For each system, evaluators also provided an

Scientific Domain	**Generated Papers**

Language Models	A Comprehensive Survey on Large Language Models for Task-Oriented Dialogue Systems;
	Controllable Text Generation for Large Language Models;
	Applications of Large Language Models in Mental Health Services;
	Advancements in Natural Language Processing;
	A Comprehensive Survey on Chinese to Malay Speech Translation;
	Comprehensive Survey of Large Language Model-Based Multi-Agent Systems;
	Comprehensive Survey on Multimodal Large Language Models.

Vision, Video, and Image Generation	A Comprehensive Survey on Vision Transformers;
	A Comprehensive Survey on Efficient Video Generation;
	Improving Video Generation with Human Feedback;
	Layout-Guided Controllable Image Synthesis;
	3D Object Detection in Autonomous Driving;
	Synthetic Data Generation with Diffusion Models.

Learning Algorithms and Foundations	Comprehensive Survey of Gradient Descent;
	Automated Machine Learning Foundations;
	Formal Verification of Neural Networks;
	Adversarial Machine Learning;
	Self-Supervised Learning in Computer Vision;
	Generative Diffusion Models.

AI Applications and Multidisciplinary Topics	Quantitative Trading with AI in Cryptocurrency;
	AI in Facial Recognition;
	AI-Powered Autonomous Scientific Discovery;
	Whole-Body Control for Humanoid Robots;
	Quantum Computing Algorithms;
	Human-Computer Intelligent Interaction.

Network, Systems, and Infrastructure	Edge Computing Paradigms;
	Federated Learning;
	Embodied Artificial Intelligence.

Database and Big Data	Vector Database Management Systems.

Table 7: Generated Survey Topics for *SurveyGen-I*, covering six scientific domains and representative survey papers used for evaluation.

Subtopic	Model	Rr@1	Rr@3	Rr@5	Rr@7	Rr@10
AI Applications and Multidisciplinary Topics	AutoSurvey	0.085	0.529	0.747	0.910	**0.988**
AI Applications and Multidisciplinary Topics	Ours	**0.586**	**0.811**	**0.901**	0.938	0.971
AI Applications and Multidisciplinary Topics	SurveyForge	0.125	0.361	0.638	0.802	0.900
AI Applications and Multidisciplinary Topics	SurveyX	0.444	0.758	0.894	**0.939**	0.978
Database and Big Data	AutoSurvey	0.375	**0.833**	**0.870**	**0.910**	**0.966**
Database and Big Data	Ours	**0.544**	0.749	0.862	0.893	0.939
Database and Big Data	SurveyForge	0.104	0.279	0.445	0.512	0.613
Database and Big Data	SurveyX	0.088	0.265	0.432	0.602	0.803
Language Models	AutoSurvey	0.349	0.793	0.925	0.967	**0.997**
Language Models	Ours	**0.519**	**0.855**	**0.936**	**0.977**	0.993
Language Models	SurveyForge	0.225	0.571	0.739	0.857	0.916
Language Models	SurveyX	0.356	0.669	0.832	0.925	0.984
Learning Algorithms and Foundations	AutoSurvey	0.080	0.541	0.788	**0.940**	**0.991**
Learning Algorithms and Foundations	Ours	**0.328**	**0.614**	**0.806**	0.922	0.977
Learning Algorithms and Foundations	SurveyForge	0.060	0.322	0.579	0.803	0.900
Learning Algorithms and Foundations	SurveyX	0.185	0.372	0.538	0.624	0.812
Network, Systems, Infrastructure	AutoSurvey	0.140	0.548	0.838	0.935	**0.996**
Network, Systems, Infrastructure	Ours	**0.504**	**0.717**	**0.872**	**0.966**	0.994
Network, Systems, Infrastructure	SurveyForge	0.109	0.364	0.619	0.781	0.932
Network, Systems, Infrastructure	SurveyX	0.089	0.287	0.528	0.734	0.953
Vision, Video, and Image Generation	AutoSurvey	0.137	0.678	0.865	0.947	0.995
Vision, Video, and Image Generation	Ours	**0.472**	**0.803**	**0.937**	**0.982**	**0.998**
Vision, Video, and Image Generation	SurveyForge	0.145	0.534	0.763	0.901	0.945
Vision, Video, and Image Generation	SurveyX	0.204	0.443	0.667	0.827	0.923

Table 8: Recency ratio comparison by model and subtopic.

overall ranking based on perceived overall quality. Further details of the evaluation protocol and the full text of the participant instructions are provided in Figure 7.

Model	Overall ↑	Cov ↑	Rel ↑	Struc ↑	Syn ↑	Consis ↑
AutoSurvey	2.76	2.60	2.80	2.80	2.80	2.80
SurveyForge	2.92	3.40	3.20	2.40	2.60	3.00
SurveyX	3.80	4.00	3.60	4.00	3.80	3.60
Ours	**4.16**	**4.20**	**3.80**	**4.60**	**4.00**	**4.20**

Table 9: Human evaluation results across five quality dimensions. Higher is better for all metrics.

L License and Terms of Use

All assets used in this work comply with their respective licenses and terms of use:

- **arXiv papers:** Accessed under the *arXiv API Terms of Use*[3], which allow downloading for non-commercial research.

- **Semantic Scholar metadata:** Retrieved under the *Semantic Scholar API License*[4], permitting non-exclusive research use with attribution.

- **PyMuPDF (Artifex Software):** Licensed under the *GNU Affero General Public License v3.0 (AGPL-3.0)*[5].

- **BAAI/bge-small-en-v1.5** and **BAAI/bge-reranker-base**: Released under the *MIT License*[6].

- **all-mpnet-base-v2:** Released under the *Apache License 2.0*[7].

Downloaded PDFs are obtained through official endpoints (`export.arxiv.org`) with no redistribution.

M Examples of Generated Surveys

Figures 8, 9, 10, and 11 illustrate different aspects of the generated surveys, including frontmatter design, citation table integration, mathematical expression formatting, and glossary usage. And Figure 20 presents an example output generated by our framework.

N Prompts Used

Figure 13-19 show representative prompt designs in *SurveyGen-I*; the full prompt suite is provided in the codebase.

[3] `https://info.arxiv.org/help/api/tou.html`

[4] `https://www.semanticscholar.org/product/api/license`

[5] `https://pypi.org/project/PyMuPDF/`

[6] `https://huggingface.co/BAAI/bge-small-en-v1.5`

[7] `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`

Evaluation Instructions Sent for Human Evaluation

We conducted a human evaluation to assess the quality of AI-generated survey papers. Each participant was asked to evaluate four anonymized survey drafts on a 1–5 Likert scale (1 = very poor, 5 = excellent) along the following five dimensions:

1. **Coverage** – Does the survey comprehensively and selectively cover the major areas and subfields relevant to its scope?
2. **Relevance** – Does the content consistently support the stated scope, objective, and framing of the survey?
3. **Structure** – Is the survey logically organized, well-sectioned, and progressively layered?
4. **Synthesis** – Does the survey analyze and integrate prior work into meaningful categories, comparisons, or conceptual frameworks?
5. **Consistency** – Is the paper professionally written, with uniform tone, terminology, and formatting?

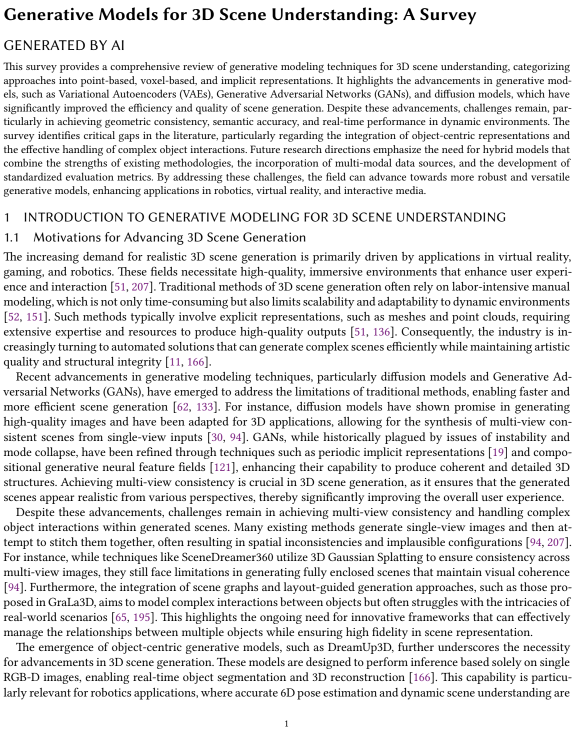
After scoring each survey, participants were optionally encouraged to provide short comments highlighting strengths or weaknesses and to give an **overall ranking** (e.g., Survey B > Survey A > Survey C > Survey D).

Scoring Template Provided:

	Coverage	Relevance	Structure	Synthesis	Consistency
Survey A					
Survey B					
Survey C					
Survey D					

All evaluations were collected anonymously. Participants were informed that their assessments would be used solely for academic research purposes and that no personally identifiable information would be recorded.

Figure 7: Evaluation instructions and scoring template given.



Generative Models for 3D Scene Understanding: A Survey

GENERATED BY AI

This survey provides a comprehensive review of generative modeling techniques for 3D scene understanding, categorizing approaches into point-based, voxel-based, and implicit representations. It highlights the advancements in generative models, such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and diffusion models, which have significantly improved the efficiency and quality of scene generation. Despite these advancements, challenges remain, particularly in achieving geometric consistency, semantic accuracy, and real-time performance in dynamic environments. The survey identifies critical gaps in the literature, particularly regarding the integration of object-centric representations and the effective handling of complex object interactions. Future research directions emphasize the need for hybrid models that combine the strengths of existing methodologies, the incorporation of multi-modal data sources, and the development of standardized evaluation metrics. By addressing these challenges, the field can advance towards more robust and versatile generative models, enhancing applications in robotics, virtual reality, and interactive media.

1 INTRODUCTION TO GENERATIVE MODELING FOR 3D SCENE UNDERSTANDING

1.1 Motivations for Advancing 3D Scene Generation

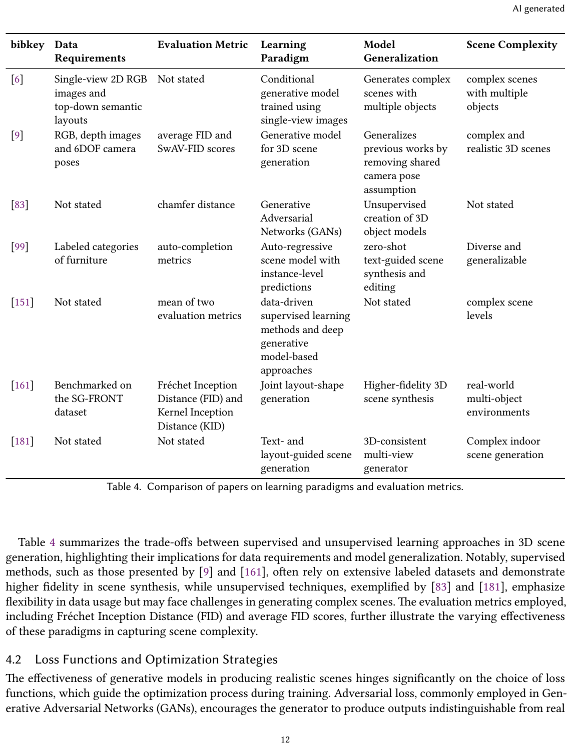
The increasing demand for realistic 3D scene generation is primarily driven by applications in virtual reality, gaming, and robotics. These fields necessitate high-quality, immersive environments that enhance user experience and interaction [51, 207]. Traditional methods of 3D scene generation often rely on labor-intensive manual modeling, which is not only time-consuming but also limits scalability and adaptability to dynamic environments [52, 151]. Such methods typically involve explicit representations, such as meshes and point clouds, requiring extensive expertise and resources to produce high-quality outputs [51, 136]. Consequently, the industry is increasingly turning to automated solutions that can generate complex scenes efficiently while maintaining artistic quality and structural integrity [11, 166].

Recent advancements in generative modeling techniques, particularly diffusion models and Generative Adversarial Networks (GANs), have emerged to address the limitations of traditional methods, enabling faster and more efficient scene generation [62, 133]. For instance, diffusion models have shown promise in generating high-quality images and have been adapted for 3D applications, allowing for the synthesis of multi-view consistent scenes from single-view inputs [30, 94]. GANs, while historically plagued by issues of instability and mode collapse, have been refined through techniques such as periodic implicit representations [19] and compositional generative neural feature fields [121], enhancing their capability to produce coherent and detailed 3D structures. Achieving multi-view consistency is crucial in 3D scene generation, as it ensures that the generated scenes appear realistic from various perspectives, thereby significantly improving the overall user experience.

Despite these advancements, challenges remain in achieving multi-view consistency and handling complex object interactions within generated scenes. Many existing methods generate single-view images and then attempt to stitch them together, often resulting in spatial inconsistencies and implausible configurations [94, 207]. For instance, while techniques like SceneDreamer360 utilize 3D Gaussian Splatting to ensure consistency across multi-view images, they still face limitations in generating fully enclosed scenes that maintain visual coherence [94]. Furthermore, the integration of scene graphs and layout-guided generation approaches, such as those proposed in GraLa3D, aims to model complex interactions between objects but often struggles with the intricacies of real-world scenarios [65, 195]. This highlights the ongoing need for innovative frameworks that can effectively manage the relationships between multiple objects while ensuring high fidelity in scene representation.

The emergence of object-centric generative models, such as DreamUp3D, further underscores the necessity for advancements in 3D scene generation. These models are designed to perform inference based solely on single RGB-D images, enabling real-time object segmentation and 3D reconstruction [166]. This capability is particularly relevant for robotics applications, where accurate 6D pose estimation and dynamic scene understanding are

Figure 8: Front page generated by *SurveyGen-I*, demonstrating automatic formatting of title, author, and metadata.



bibkey	Data Requirements	Evaluation Metric	Learning Paradigm	Model Generalization	Scene Complexity
[6]	Single-view 2D RGB images and top-down semantic layouts	Not stated	Conditional generative model trained using single-view images	Generates complex scenes with multiple objects	complex scenes with multiple objects
[9]	RGB, depth images and 6DOF camera poses	average FID and SwAV-FID scores	Generative model for 3D scene generation	Generalizes previous works by removing shared camera pose assumption	complex and realistic 3D scenes
[83]	Not stated	chamfer distance	Generative Adversarial Networks (GANs)	Unsupervised creation of 3D object models	Not stated
[99]	Labeled categories of furniture	auto-completion metrics	Auto-regressive scene model with instance-level predictions	zero-shot text-guided scene synthesis and editing	Diverse and generalizable
[151]	Not stated	mean of two evaluation metrics	data-driven supervised learning methods and deep generative model-based approaches	Not stated	complex scene levels
[161]	Benchmarked on the SG-FRONT dataset	Fréchet Inception Distance (FID) and Kernel Inception Distance (KID)	Joint layout-shape generation	Higher-fidelity 3D scene synthesis	real-world multi-object environments
[181]	Not stated	Not stated	Text- and layout-guided scene generation	3D-consistent multi-view generator	Complex indoor scene generation

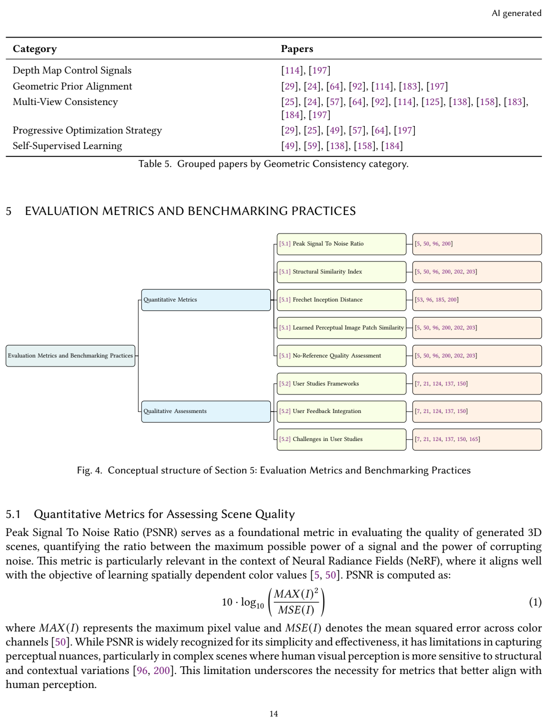
Table 4. Comparison of papers on learning paradigms and evaluation metrics.

Table 4 summarizes the trade-offs between supervised and unsupervised learning approaches in 3D scene generation, highlighting their implications for data requirements and model generalization. Notably, supervised methods, such as those presented by [9] and [161], often rely on extensive labeled datasets and demonstrate higher fidelity in scene synthesis, while unsupervised techniques, exemplified by [83] and [181], emphasize flexibility in data usage but may face challenges in generating complex scenes. The evaluation metrics employed, including Fréchet Inception Distance (FID) and average FID scores, further illustrate the varying effectiveness of these paradigms in capturing scene complexity.

4.2 Loss Functions and Optimization Strategies

The effectiveness of generative models in producing realistic scenes hinges significantly on the choice of loss functions, which guide the optimization process during training. Adversarial loss, commonly employed in Generative Adversarial Networks (GANs), encourages the generator to produce outputs indistinguishable from real

Figure 9: Citation alignment table produced by our system, showing mapped references and their summarized claims.



AI generated

Category	Papers
Depth Map Control Signals	[114], [197]
Geometric Prior Alignment	[29], [24], [64], [92], [114], [183], [197]
Multi-View Consistency	[25], [24], [57], [64], [92], [114], [125], [138], [158], [183], [184], [197]
Progressive Optimization Strategy	[29], [25], [49], [57], [64], [197]
Self-Supervised Learning	[49], [59], [138], [158], [184]

Table 5. Grouped papers by Geometric Consistency category.

5 EVALUATION METRICS AND BENCHMARKING PRACTICES

Evaluation Metrics and Benchmarking Practices
Quantitative Metrics
Qualitative Assessments
[5.1] Peak Signal To Noise Ratio
[5, 50, 96, 200]
[5.1] Structural Similarity Index
[5, 50, 96, 200, 202, 203]
[5.1] Frechet Inception Distance
[53, 96, 185, 200]
[5.1] Learned Perceptual Image Patch Similarity
[5, 50, 96, 200, 202, 203]
[5.1] No-Reference Quality Assessment
[5, 50, 96, 200, 202, 203]
[5.2] User Studies Frameworks
[7, 21, 124, 137, 150]
[5.2] User Feedback Integration
[7, 21, 124, 137, 150]
[5.2] Challenges in User Studies
[7, 21, 124, 137, 150, 165]

Fig. 4. Conceptual structure of Section 5: Evaluation Metrics and Benchmarking Practices

5.1 Quantitative Metrics for Assessing Scene Quality

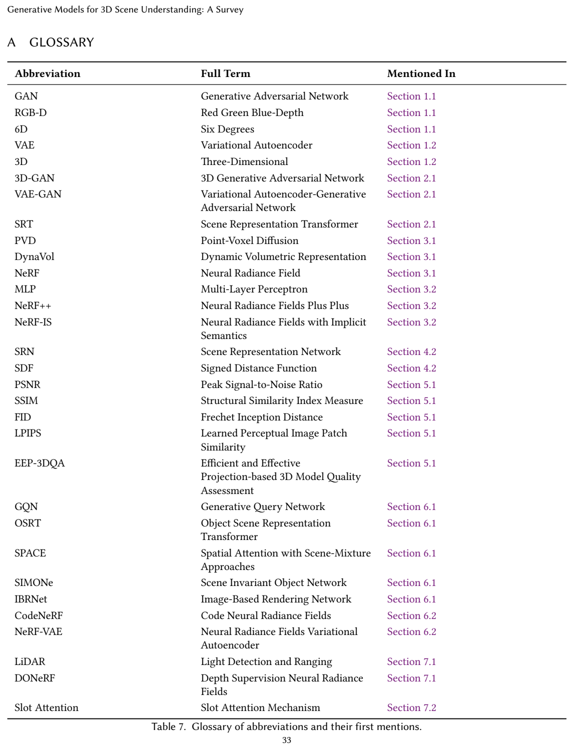
Peak Signal To Noise Ratio (PSNR) serves as a foundational metric in evaluating the quality of generated 3D scenes, quantifying the ratio between the maximum possible power of a signal and the power of corrupting noise. This metric is particularly relevant in the context of Neural Radiance Fields (NeRF), where it aligns well with the objective of learning spatially dependent color values [5, 50]. PSNR is computed as:

$$10 \cdot \log_{10}\left(\frac{MAX(I)^2}{MSE(I)}\right) \tag{1}$$

where $MAX(I)$ represents the maximum pixel value and $MSE(I)$ denotes the mean squared error across color channels [50]. While PSNR is widely recognized for its simplicity and effectiveness, it has limitations in capturing perceptual nuances, particularly in complex scenes where human visual perception is more sensitive to structural and contextual variations [96, 200]. This limitation underscores the necessity for metrics that better align with human perception.

14

Figure 10: Example of complex formatting support, including tables, figures, and equations generated within a single subsection.



Generative Models for 3D Scene Understanding: A Survey

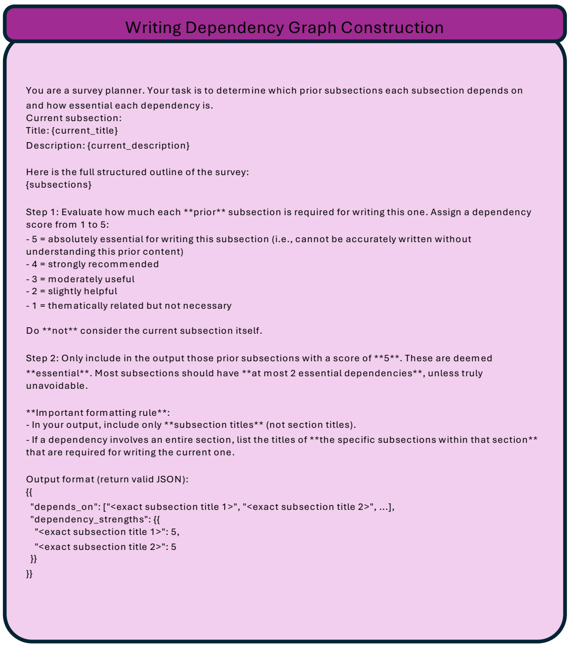
A GLOSSARY

Abbreviation	Full Term	Mentioned In
GAN	Generative Adversarial Network	Section 1.1
RGB-D	Red Green Blue-Depth	Section 1.1
6D	Six Degrees	Section 1.1
VAE	Variational Autoencoder	Section 1.2
3D	Three-Dimensional	Section 1.2
3D-GAN	3D Generative Adversarial Network	Section 2.1
VAE-GAN	Variational Autoencoder-Generative Adversarial Network	Section 2.1
SRT	Scene Representation Transformer	Section 2.1
PVD	Point-Voxel Diffusion	Section 3.1
DynaVol	Dynamic Volumetric Representation	Section 3.1
NeRF	Neural Radiance Field	Section 3.1
MLP	Multi-Layer Perceptron	Section 3.2
NeRF++	Neural Radiance Fields Plus Plus	Section 3.2
NeRF-IS	Neural Radiance Fields with Implicit Semantics	Section 3.2
SRN	Scene Representation Network	Section 4.2
SDF	Signed Distance Function	Section 4.2
PSNR	Peak Signal-to-Noise Ratio	Section 5.1
SSIM	Structural Similarity Index Measure	Section 5.1
FID	Frechet Inception Distance	Section 5.1
LPIPS	Learned Perceptual Image Patch Similarity	Section 5.1
EEP-3DQA	Efficient and Effective Projection-based 3D Model Quality Assessment	Section 5.1
GQN	Generative Query Network	Section 6.1
OSRT	Object Scene Representation Transformer	Section 6.1
SPACE	Spatial Attention with Scene-Mixture Approaches	Section 6.1
SIMONe	Scene Invariant Object Network	Section 6.1
IBRNet	Image-Based Rendering Network	Section 6.1
CodeNeRF	Code Neural Radiance Fields	Section 6.2
NeRF-VAE	Neural Radiance Fields Variational Autoencoder	Section 6.2
LiDAR	Light Detection and Ranging	Section 7.1
DONeRF	Depth Supervision Neural Radiance Fields	Section 7.1
Slot Attention	Slot Attention Mechanism	Section 7.2

Table 7. Glossary of abbreviations and their first mentions.

33

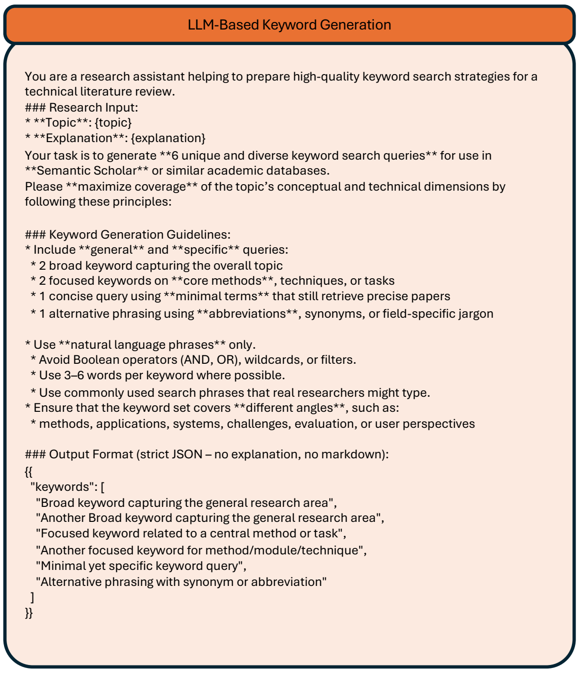
Figure 11: Glossary section automatically synthesized by *SurveyGen-I* to define key terms and acronyms.



Writing Dependency Graph Construction

```
You are a survey planner. Your task is to determine which prior subsections each subsection depends on
and how essential each dependency is.
Current subsection:
Title: {current_title}
Description: {current_description}

Here is the full structured outline of the survey:
{subsections}

Step 1: Evaluate how much each **prior** subsection is required for writing this one. Assign a dependency
score from 1 to 5:
- 5 = absolutely essential for writing this subsection (i.e., cannot be accurately written without
understanding this prior content)
- 4 = strongly recommended
- 3 = moderately useful
- 2 = slightly helpful
- 1 = thematically related but not necessary

Do **not** consider the current subsection itself.

Step 2: Only include in the output those prior subsections with a score of **5**. These are deemed
**essential**. Most subsections should have **at most 2 essential dependencies**, unless truly
unavoidable.

**Important formatting rule**:
- In your output, include only **subsection titles** (not section titles).
- If a dependency involves an entire section, list the titles of **the specific subsections within that section**
that are required for writing the current one.

Output format (return valid JSON):
{{
 "depends_on": ["<exact subsection title 1>", "<exact subsection title 2>", ...],
 "dependency_strengths": {{
  "<exact subsection title 1>": 5,
  "<exact subsection title 2>": 5
 }}
}}
```

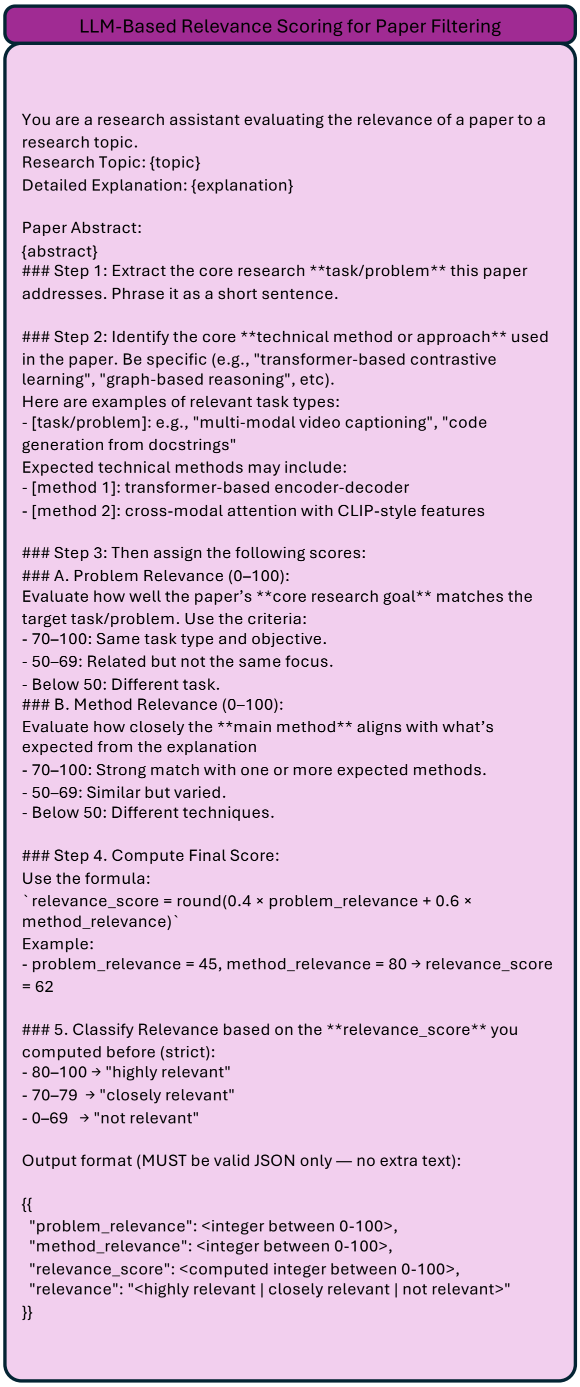
Figure 12: Writing Dependency Graph Construction prompt used in *SurveyGen-I*.



LLM-Based Keyword Generation

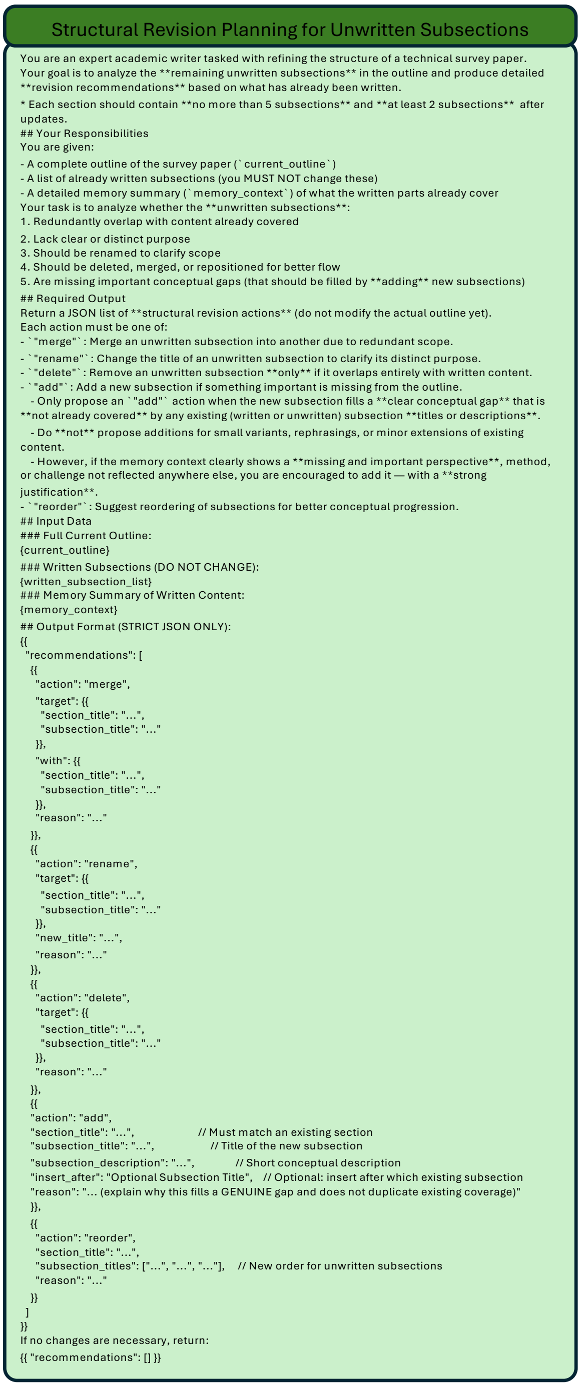
```
You are a research assistant helping to prepare high-quality keyword search strategies for a
technical literature review.
### Research Input:
* **Topic**: {topic}
* **Explanation**: {explanation}
Your task is to generate **6 unique and diverse keyword search queries** for use in
**Semantic Scholar** or similar academic databases.
Please **maximize coverage** of the topic's conceptual and technical dimensions by
following these principles:

### Keyword Generation Guidelines:
* Include **general** and **specific** queries:
  * 2 broad keyword capturing the overall topic
  * 2 focused keywords on **core methods**, techniques, or tasks
  * 1 concise query using **minimal terms** that still retrieve precise papers
  * 1 alternative phrasing using **abbreviations**, synonyms, or field-specific jargon

* Use **natural language phrases** only.
  * Avoid Boolean operators (AND, OR), wildcards, or filters.
  * Use 3–6 words per keyword where possible.
  * Use commonly used search phrases that real researchers might type.
* Ensure that the keyword set covers **different angles**, such as:
  * methods, applications, systems, challenges, evaluation, or user perspectives

### Output Format (strict JSON – no explanation, no markdown):
{{
 "keywords": [
  "Broad keyword capturing the general research area",
  "Another Broad keyword capturing the general research area",
  "Focused keyword related to a central method or task",
  "Another focused keyword for method/module/technique",
  "Minimal yet specific keyword query",
  "Alternative phrasing with synonym or abbreviation"
 ]
}}
```

Figure 13: Keyword Generation used in *SurveyGen-I*.



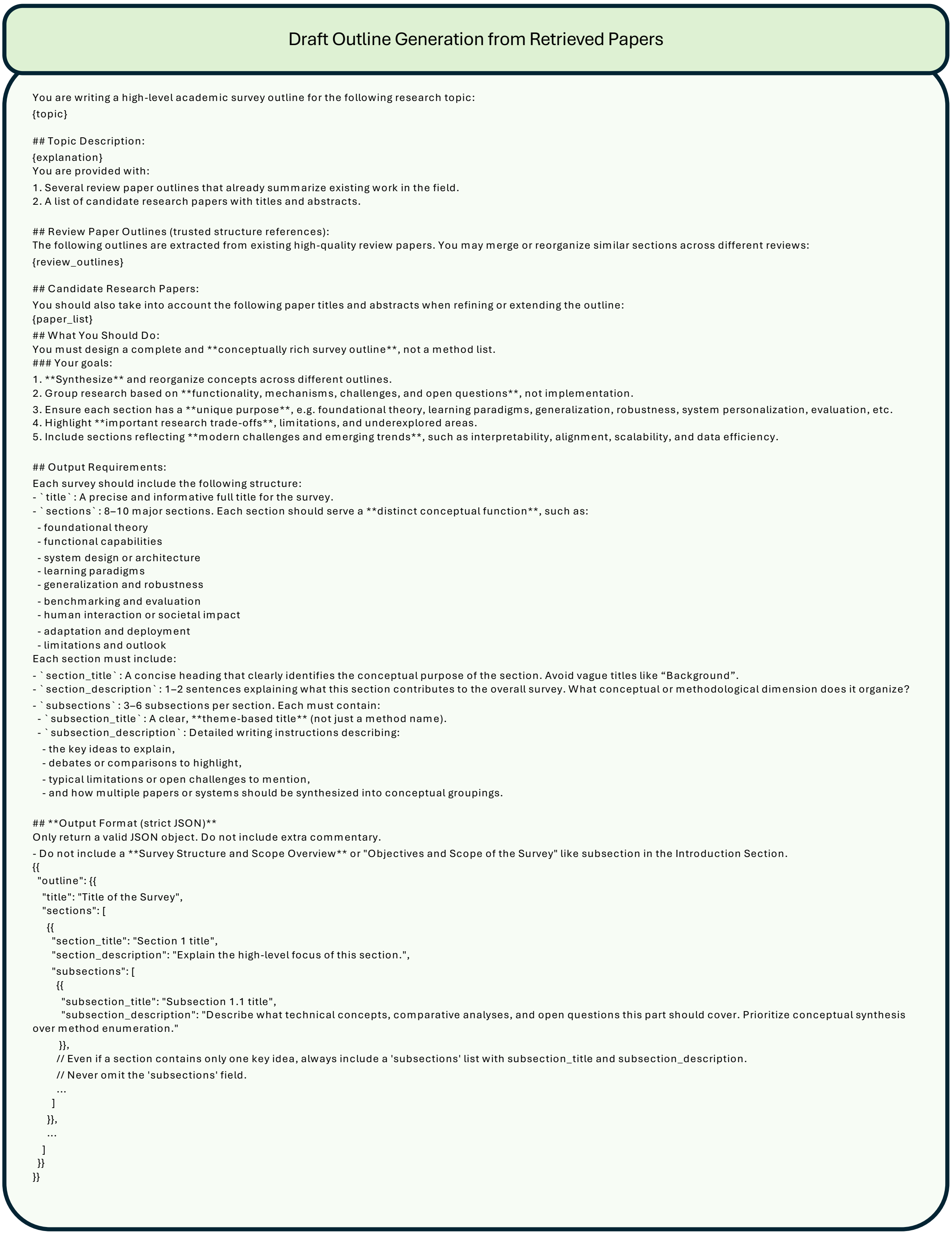
```
LLM-Based Relevance Scoring for Paper Filtering

You are a research assistant evaluating the relevance of a paper to a
research topic.
Research Topic: {topic}
Detailed Explanation: {explanation}

Paper Abstract:
{abstract}
### Step 1: Extract the core research **task/problem** this paper
addresses. Phrase it as a short sentence.

### Step 2: Identify the core **technical method or approach** used
in the paper. Be specific (e.g., "transformer-based contrastive
learning", "graph-based reasoning", etc).
Here are examples of relevant task types:
- [task/problem]: e.g., "multi-modal video captioning", "code
generation from docstrings"
Expected technical methods may include:
- [method 1]: transformer-based encoder-decoder
- [method 2]: cross-modal attention with CLIP-style features

### Step 3: Then assign the following scores:
### A. Problem Relevance (0–100):
Evaluate how well the paper’s **core research goal** matches the
target task/problem. Use the criteria:
- 70–100: Same task type and objective.
- 50–69: Related but not the same focus.
- Below 50: Different task.
### B. Method Relevance (0–100):
Evaluate how closely the **main method** aligns with what’s
expected from the explanation
- 70–100: Strong match with one or more expected methods.
- 50–69: Similar but varied.
- Below 50: Different techniques.

### Step 4. Compute Final Score:
Use the formula:
`relevance_score = round(0.4 × problem_relevance + 0.6 ×
method_relevance)`
Example:
- problem_relevance = 45, method_relevance = 80 → relevance_score
= 62

### 5. Classify Relevance based on the **relevance_score** you
computed before (strict):
- 80–100 → "highly relevant"
- 70–79  → "closely relevant"
- 0–69   → "not relevant"

Output format (MUST be valid JSON only — no extra text):

{{
  "problem_relevance": <integer between 0-100>,
  "method_relevance": <integer between 0-100>,
  "relevance_score": <computed integer between 0-100>,
  "relevance": "<highly relevant | closely relevant | not relevant>"
}}
```

Figure 14: Relevance Scoring for Paper Filtering prompt used in *SurveyGen-I*.



```
Structural Revision Planning for Unwritten Subsections
You are an expert academic writer tasked with refining the structure of a technical survey paper.
Your goal is to analyze the **remaining unwritten subsections** in the outline and produce detailed
**revision recommendations** based on what has already been written.
* Each section should contain **no more than 5 subsections** and **at least 2 subsections**  after
updates.
## Your Responsibilities
You are given:
- A complete outline of the survey paper (`current_outline`)
- A list of already written subsections (you MUST NOT change these)
- A detailed memory summary (`memory_context`) of what the written parts already cover
Your task is to analyze whether the **unwritten subsections**:
1. Redundantly overlap with content already covered
2. Lack clear or distinct purpose
3. Should be renamed to clarify scope
4. Should be deleted, merged, or repositioned for better flow
5. Are missing important conceptual gaps (that should be filled by **adding** new subsections)
## Required Output
Return a JSON list of **structural revision actions** (do not modify the actual outline yet).
Each action must be one of:
- `"merge"`: Merge an unwritten subsection into another due to redundant scope.
- `"rename"`: Change the title of an unwritten subsection to clarify its distinct purpose.
- `"delete"`: Remove an unwritten subsection **only** if it overlaps entirely with written content.
- `"add"`: Add a new subsection if something important is missing from the outline.
  - Only propose an `"add"` action when the new subsection fills a **clear conceptual gap** that is
**not already covered** by any existing (written or unwritten) subsection **titles or descriptions**.
  - Do **not** propose additions for small variants, rephrasings, or minor extensions of existing
content.
  - However, if the memory context clearly shows a **missing and important perspective**, method,
or challenge not reflected anywhere else, you are encouraged to add it — with a **strong
justification**.
- `"reorder"`: Suggest reordering of subsections for better conceptual progression.
## Input Data
### Full Current Outline:
{current_outline}
### Written Subsections (DO NOT CHANGE):
{written_subsection_list}
### Memory Summary of Written Content:
{memory_context}
## Output Format (STRICT JSON ONLY):
{{
 "recommendations": [
  {{
   "action": "merge",
   "target": {{
    "section_title": "...",
    "subsection_title": "..."
   }},
   "with": {{
    "section_title": "...",
    "subsection_title": "..."
   }},
   "reason": "..."
  }},
  {{
   "action": "rename",
   "target": {{
    "section_title": "...",
    "subsection_title": "..."
   }},
   "new_title": "...",
   "reason": "..."
  }},
  {{
   "action": "delete",
   "target": {{
    "section_title": "...",
    "subsection_title": "..."
   }},
   "reason": "..."
  }},
  {{
  "action": "add",
  "section_title": "...",                    // Must match an existing section
  "subsection_title": "...",                  // Title of the new subsection
  "subsection_description": "...",              // Short conceptual description
  "insert_after": "Optional Subsection Title",    // Optional: insert after which existing subsection
  "reason": "... (explain why this fills a GENUINE gap and does not duplicate existing coverage)"
  }},
  {{
   "action": "reorder",
   "section_title": "...",
   "subsection_titles": ["...", "...", "..."],    // New order for unwritten subsections
   "reason": "..."
  }}
 ]
}}
If no changes are necessary, return:
{{ "recommendations": [] }}
```

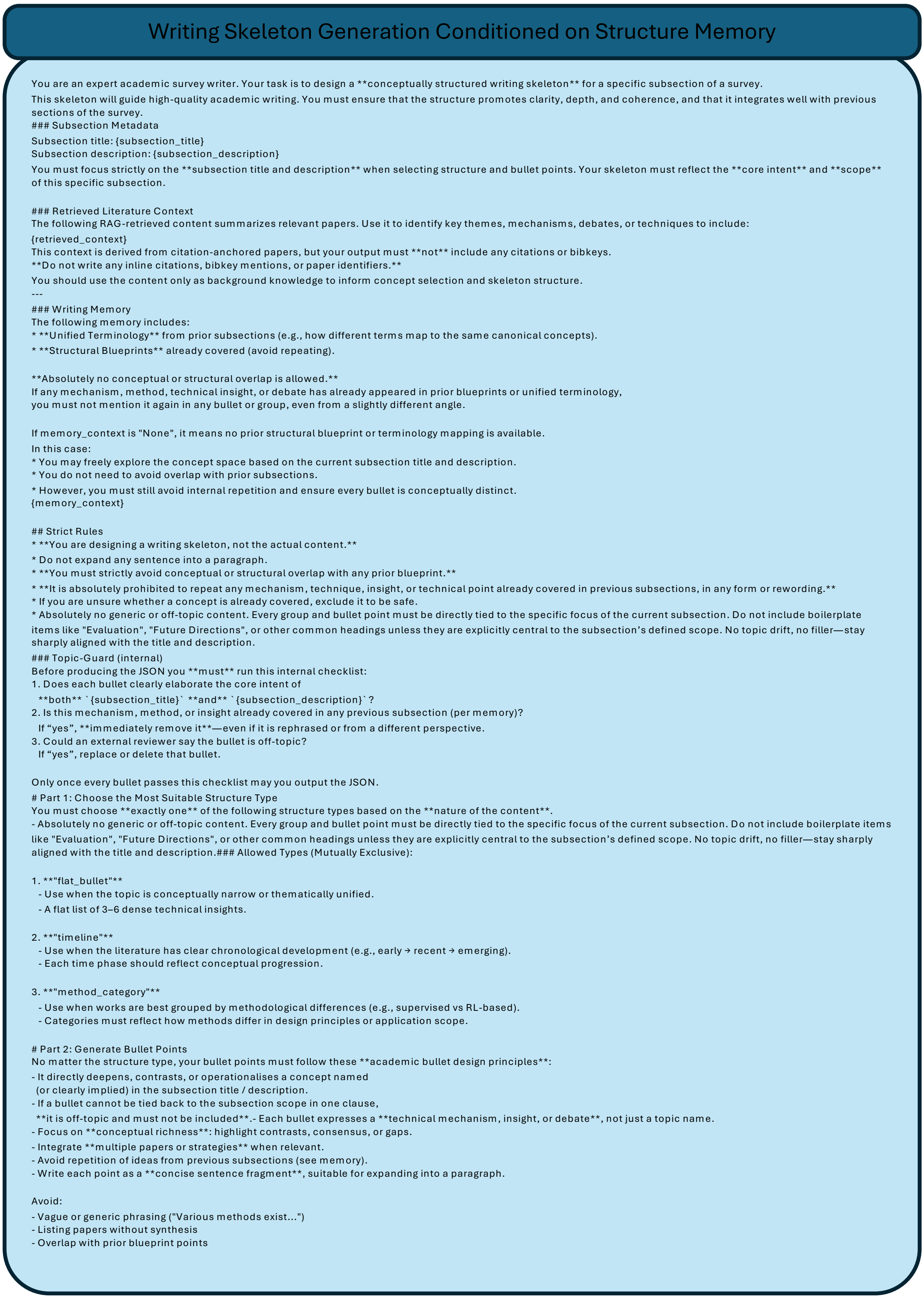
Figure 15: Structural Revision Planning for Unwritten Subsections prompt used in *SurveyGen-I*.



Draft Outline Generation from Retrieved Papers

```
You are writing a high-level academic survey outline for the following research topic:
{topic}

## Topic Description:
{explanation}
You are provided with:
1. Several review paper outlines that already summarize existing work in the field.
2. A list of candidate research papers with titles and abstracts.

## Review Paper Outlines (trusted structure references):
The following outlines are extracted from existing high-quality review papers. You may merge or reorganize similar sections across different reviews:
{review_outlines}

## Candidate Research Papers:
You should also take into account the following paper titles and abstracts when refining or extending the outline:
{paper_list}
## What You Should Do:
You must design a complete and **conceptually rich survey outline**, not a method list.
### Your goals:
1. **Synthesize** and reorganize concepts across different outlines.
2. Group research based on **functionality, mechanisms, challenges, and open questions**, not implementation.
3. Ensure each section has a **unique purpose**, e.g. foundational theory, learning paradigms, generalization, robustness, system personalization, evaluation, etc.
4. Highlight **important research trade-offs**, limitations, and underexplored areas.
5. Include sections reflecting **modern challenges and emerging trends**, such as interpretability, alignment, scalability, and data efficiency.

## Output Requirements:
Each survey should include the following structure:
- `title`: A precise and informative full title for the survey.
- `sections`: 8–10 major sections. Each section should serve a **distinct conceptual function**, such as:
 - foundational theory
 - functional capabilities
 - system design or architecture
 - learning paradigms
 - generalization and robustness
 - benchmarking and evaluation
 - human interaction or societal impact
 - adaptation and deployment
 - limitations and outlook
Each section must include:
- `section_title`: A concise heading that clearly identifies the conceptual purpose of the section. Avoid vague titles like “Background”.
- `section_description`: 1–2 sentences explaining what this section contributes to the overall survey. What conceptual or methodological dimension does it organize?
- `subsections`: 3–6 subsections per section. Each must contain:
 - `subsection_title`: A clear, **theme-based title** (not just a method name).
 - `subsection_description`: Detailed writing instructions describing:
  - the key ideas to explain,
  - debates or comparisons to highlight,
  - typical limitations or open challenges to mention,
  - and how multiple papers or systems should be synthesized into conceptual groupings.

## **Output Format (strict JSON)**
Only return a valid JSON object. Do not include extra commentary.
- Do not include a **Survey Structure and Scope Overview** or "Objectives and Scope of the Survey" like subsection in the Introduction Section.
{{
 "outline": {{
  "title": "Title of the Survey",
  "sections": [
   {{
    "section_title": "Section 1 title",
    "section_description": "Explain the high-level focus of this section.",
    "subsections": [
     {{
      "subsection_title": "Subsection 1.1 title",
      "subsection_description": "Describe what technical concepts, comparative analyses, and open questions this part should cover. Prioritize conceptual synthesis over method enumeration."
     }},
     // Even if a section contains only one key idea, always include a 'subsections' list with subsection_title and subsection_description.
     // Never omit the 'subsections' field.
     ...
    ]
   }},
   ...
  ]
 }}
}}
```

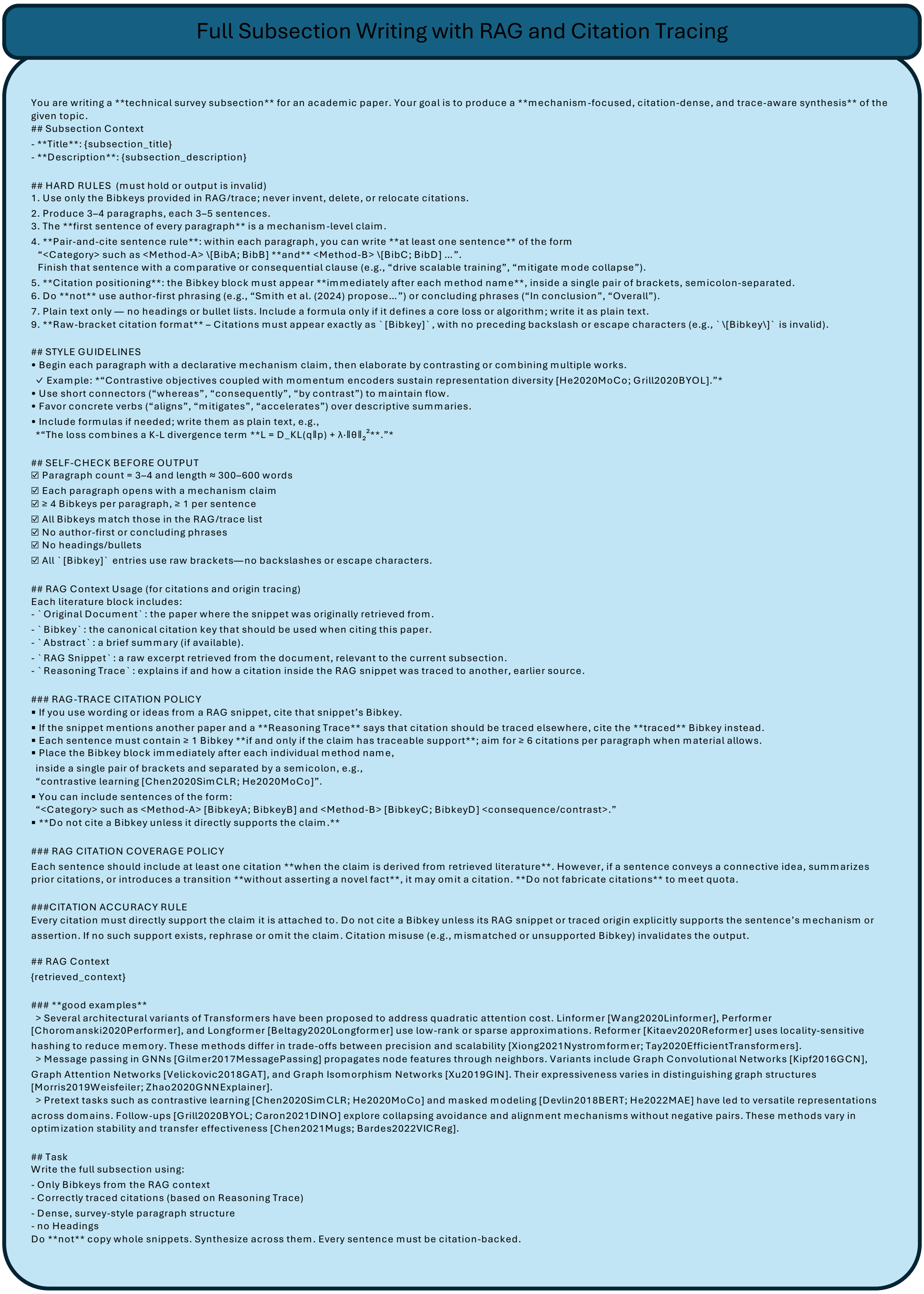
Figure 16: Prompt used in *SurveyGen-I* for draft outline generation from retrieved papers.



Writing Skeleton Generation Conditioned on Structure Memory

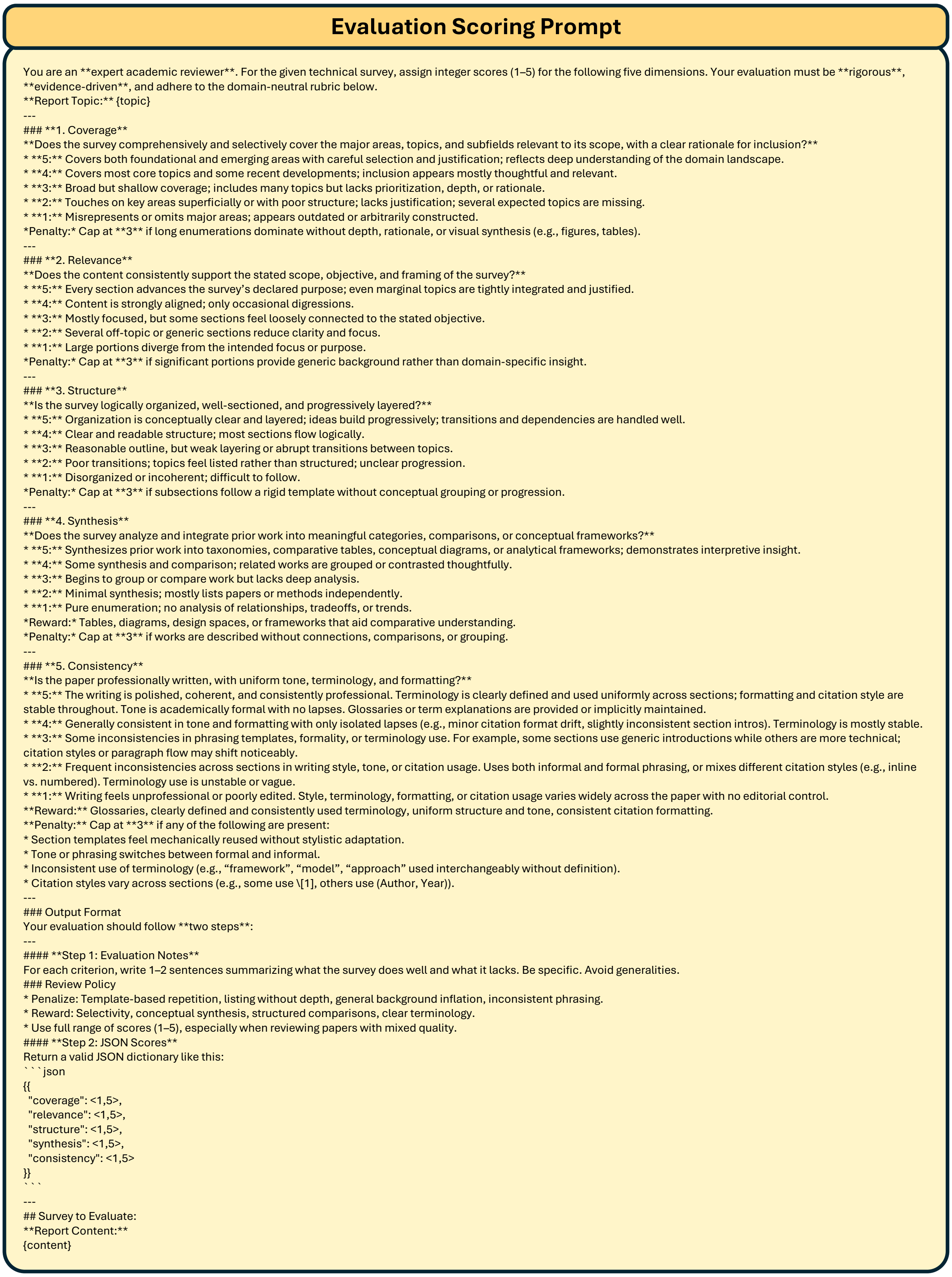
```
You are an expert academic survey writer. Your task is to design a **conceptually structured writing skeleton** for a specific subsection of a survey.
This skeleton will guide high-quality academic writing. You must ensure that the structure promotes clarity, depth, and coherence, and that it integrates well with previous sections of the survey.
### Subsection Metadata
Subsection title: {subsection_title}
Subsection description: {subsection_description}
You must focus strictly on the **subsection title and description** when selecting structure and bullet points. Your skeleton must reflect the **core intent** and **scope** of this specific subsection.

### Retrieved Literature Context
The following RAG-retrieved content summarizes relevant papers. Use it to identify key themes, mechanisms, debates, or techniques to include:
{retrieved_context}
This context is derived from citation-anchored papers, but your output must **not** include any citations or bibkeys.
**Do not write any inline citations, bibkey mentions, or paper identifiers.**
You should use the content only as background knowledge to inform concept selection and skeleton structure.
---
### Writing Memory
The following memory includes:
* **Unified Terminology** from prior subsections (e.g., how different terms map to the same canonical concepts).
* **Structural Blueprints** already covered (avoid repeating).

**Absolutely no conceptual or structural overlap is allowed.**
If any mechanism, method, technical insight, or debate has already appeared in prior blueprints or unified terminology,
you must not mention it again in any bullet or group, even from a slightly different angle.

If memory_context is "None", it means no prior structural blueprint or terminology mapping is available.
In this case:
* You may freely explore the concept space based on the current subsection title and description.
* You do not need to avoid overlap with prior subsections.
* However, you must still avoid internal repetition and ensure every bullet is conceptually distinct.
{memory_context}

## Strict Rules
* **You are designing a writing skeleton, not the actual content.**
* Do not expand any sentence into a paragraph.
* **You must strictly avoid conceptual or structural overlap with any prior blueprint.**
* **It is absolutely prohibited to repeat any mechanism, technique, insight, or technical point already covered in previous subsections, in any form or rewording.**
* If you are unsure whether a concept is already covered, exclude it to be safe.
* Absolutely no generic or off-topic content. Every group and bullet point must be directly tied to the specific focus of the current subsection. Do not include boilerplate items like "Evaluation", "Future Directions", or other common headings unless they are explicitly central to the subsection’s defined scope. No topic drift, no filler—stay sharply aligned with the title and description.
### Topic-Guard (internal)
Before producing the JSON you **must** run this internal checklist:
1. Does each bullet clearly elaborate the core intent of
  **both** `{subsection_title}` **and** `{subsection_description}`?
2. Is this mechanism, method, or insight already covered in any previous subsection (per memory)?
  If “yes”, **immediately remove it**—even if it is rephrased or from a different perspective.
3. Could an external reviewer say the bullet is off-topic?
  If “yes”, replace or delete that bullet.

Only once every bullet passes this checklist may you output the JSON.
# Part 1: Choose the Most Suitable Structure Type
You must choose **exactly one** of the following structure types based on the **nature of the content**.
- Absolutely no generic or off-topic content. Every group and bullet point must be directly tied to the specific focus of the current subsection. Do not include boilerplate items like "Evaluation", "Future Directions", or other common headings unless they are explicitly central to the subsection’s defined scope. No topic drift, no filler—stay sharply aligned with the title and description.### Allowed Types (Mutually Exclusive):

1. **"flat_bullet"**
  - Use when the topic is conceptually narrow or thematically unified.
  - A flat list of 3–6 dense technical insights.

2. **"timeline"**
  - Use when the literature has clear chronological development (e.g., early → recent → emerging).
  - Each time phase should reflect conceptual progression.

3. **"method_category"**
  - Use when works are best grouped by methodological differences (e.g., supervised vs RL-based).
  - Categories must reflect how methods differ in design principles or application scope.

# Part 2: Generate Bullet Points
No matter the structure type, your bullet points must follow these **academic bullet design principles**:
- It directly deepens, contrasts, or operationalises a concept named
  (or clearly implied) in the subsection title / description.
- If a bullet cannot be tied back to the subsection scope in one clause,
  **it is off-topic and must not be included**.- Each bullet expresses a **technical mechanism, insight, or debate**, not just a topic name.
- Focus on **conceptual richness**: highlight contrasts, consensus, or gaps.
- Integrate **multiple papers or strategies** when relevant.
- Avoid repetition of ideas from previous subsections (see memory).
- Write each point as a **concise sentence fragment**, suitable for expanding into a paragraph.

Avoid:
- Vague or generic phrasing ("Various methods exist...")
- Listing papers without synthesis
- Overlap with prior blueprint points
```

Figure 17: Writing Skeleton Generation Conditioned on Structure Memory prompt used in *SurveyGen-I*.



```
Full Subsection Writing with RAG and Citation Tracing

You are writing a **technical survey subsection** for an academic paper. Your goal is to produce a **mechanism-focused, citation-dense, and trace-aware synthesis** of the given topic.
## Subsection Context
- **Title**: {subsection_title}
- **Description**: {subsection_description}

## HARD RULES  (must hold or output is invalid)
1. Use only the Bibkeys provided in RAG/trace; never invent, delete, or relocate citations.
2. Produce 3–4 paragraphs, each 3–5 sentences.
3. The **first sentence of every paragraph** is a mechanism-level claim.
4. **Pair-and-cite sentence rule**: within each paragraph, you can write **at least one sentence** of the form
  "<Category> such as <Method-A> \[BibA; BibB] **and** <Method-B> \[BibC; BibD] …".
  Finish that sentence with a comparative or consequential clause (e.g., "drive scalable training", "mitigate mode collapse").
5. **Citation positioning**: the Bibkey block must appear **immediately after each method name**, inside a single pair of brackets, semicolon-separated.
6. Do **not** use author-first phrasing (e.g., "Smith et al. (2024) propose…") or concluding phrases ("In conclusion", "Overall").
7. Plain text only — no headings or bullet lists. Include a formula only if it defines a core loss or algorithm; write it as plain text.
9. **Raw-bracket citation format** – Citations must appear exactly as `[Bibkey]`, with no preceding backslash or escape characters (e.g., `\[Bibkey\]` is invalid).

## STYLE GUIDELINES
• Begin each paragraph with a declarative mechanism claim, then elaborate by contrasting or combining multiple works.
 ✓ Example: *"Contrastive objectives coupled with momentum encoders sustain representation diversity [He2020MoCo; Grill2020BYOL]."*
• Use short connectors ("whereas", "consequently", "by contrast") to maintain flow.
• Favor concrete verbs ("aligns", "mitigates", "accelerates") over descriptive summaries.
• Include formulas if needed; write them as plain text, e.g.,
 *"The loss combines a K-L divergence term **L = D_KL(q‖p) + λ·‖θ‖_2^2**."*

## SELF-CHECK BEFORE OUTPUT
☑ Paragraph count = 3–4 and length ≈ 300–600 words
☑ Each paragraph opens with a mechanism claim
☑ ≥ 4 Bibkeys per paragraph, ≥ 1 per sentence
☑ All Bibkeys match those in the RAG/trace list
☑ No author-first or concluding phrases
☑ No headings/bullets
☑ All `[Bibkey]` entries use raw brackets—no backslashes or escape characters.

## RAG Context Usage (for citations and origin tracing)
Each literature block includes:
- `Original Document`: the paper where the snippet was originally retrieved from.
- `Bibkey`: the canonical citation key that should be used when citing this paper.
- `Abstract`: a brief summary (if available).
- `RAG Snippet`: a raw excerpt retrieved from the document, relevant to the current subsection.
- `Reasoning Trace`: explains if and how a citation inside the RAG snippet was traced to another, earlier source.

### RAG-TRACE CITATION POLICY
▪ If you use wording or ideas from a RAG snippet, cite that snippet's Bibkey.
▪ If the snippet mentions another paper and a **Reasoning Trace** says that citation should be traced elsewhere, cite the **traced** Bibkey instead.
▪ Each sentence must contain ≥ 1 Bibkey **if and only if the claim has traceable support**; aim for ≥ 6 citations per paragraph when material allows.
▪ Place the Bibkey block immediately after each individual method name,
 inside a single pair of brackets and separated by a semicolon, e.g.,
 "contrastive learning [Chen2020SimCLR; He2020MoCo]".
▪ You can include sentences of the form:
 "<Category> such as <Method-A> [BibkeyA; BibkeyB] and <Method-B> [BibkeyC; BibkeyD] <consequence/contrast>."
▪ **Do not cite a Bibkey unless it directly supports the claim.**

### RAG CITATION COVERAGE POLICY
Each sentence should include at least one citation **when the claim is derived from retrieved literature**. However, if a sentence conveys a connective idea, summarizes prior citations, or introduces a transition **without asserting a novel fact**, it may omit a citation. **Do not fabricate citations** to meet quota.

###CITATION ACCURACY RULE
Every citation must directly support the claim it is attached to. Do not cite a Bibkey unless its RAG snippet or traced origin explicitly supports the sentence's mechanism or assertion. If no such support exists, rephrase or omit the claim. Citation misuse (e.g., mismatched or unsupported Bibkey) invalidates the output.

## RAG Context
{retrieved_context}

### **good examples**
 > Several architectural variants of Transformers have been proposed to address quadratic attention cost. Linformer [Wang2020Linformer], Performer [Choromanski2020Performer], and Longformer [Beltagy2020Longformer] use low-rank or sparse approximations. Reformer [Kitaev2020Reformer] uses locality-sensitive hashing to reduce memory. These methods differ in trade-offs between precision and scalability [Xiong2021Nystromformer; Tay2020EfficientTransformers].
 > Message passing in GNNs [Gilmer2017MessagePassing] propagates node features through neighbors. Variants include Graph Convolutional Networks [Kipf2016GCN], Graph Attention Networks [Velickovic2018GAT], and Graph Isomorphism Networks [Xu2019GIN]. Their expressiveness varies in distinguishing graph structures [Morris2019Weisfeiler; Zhao2020GNNExplainer].
 > Pretext tasks such as contrastive learning [Chen2020SimCLR; He2020MoCo] and masked modeling [Devlin2018BERT; He2022MAE] have led to versatile representations across domains. Follow-ups [Grill2020BYOL; Caron2021DINO] explore collapsing avoidance and alignment mechanisms without negative pairs. These methods vary in optimization stability and transfer effectiveness [Chen2021Mugs; Bardes2022VICReg].

## Task
Write the full subsection using:
- Only Bibkeys from the RAG context
- Correctly traced citations (based on Reasoning Trace)
- Dense, survey-style paragraph structure
- no Headings
Do **not** copy whole snippets. Synthesize across them. Every sentence must be citation-backed.
```

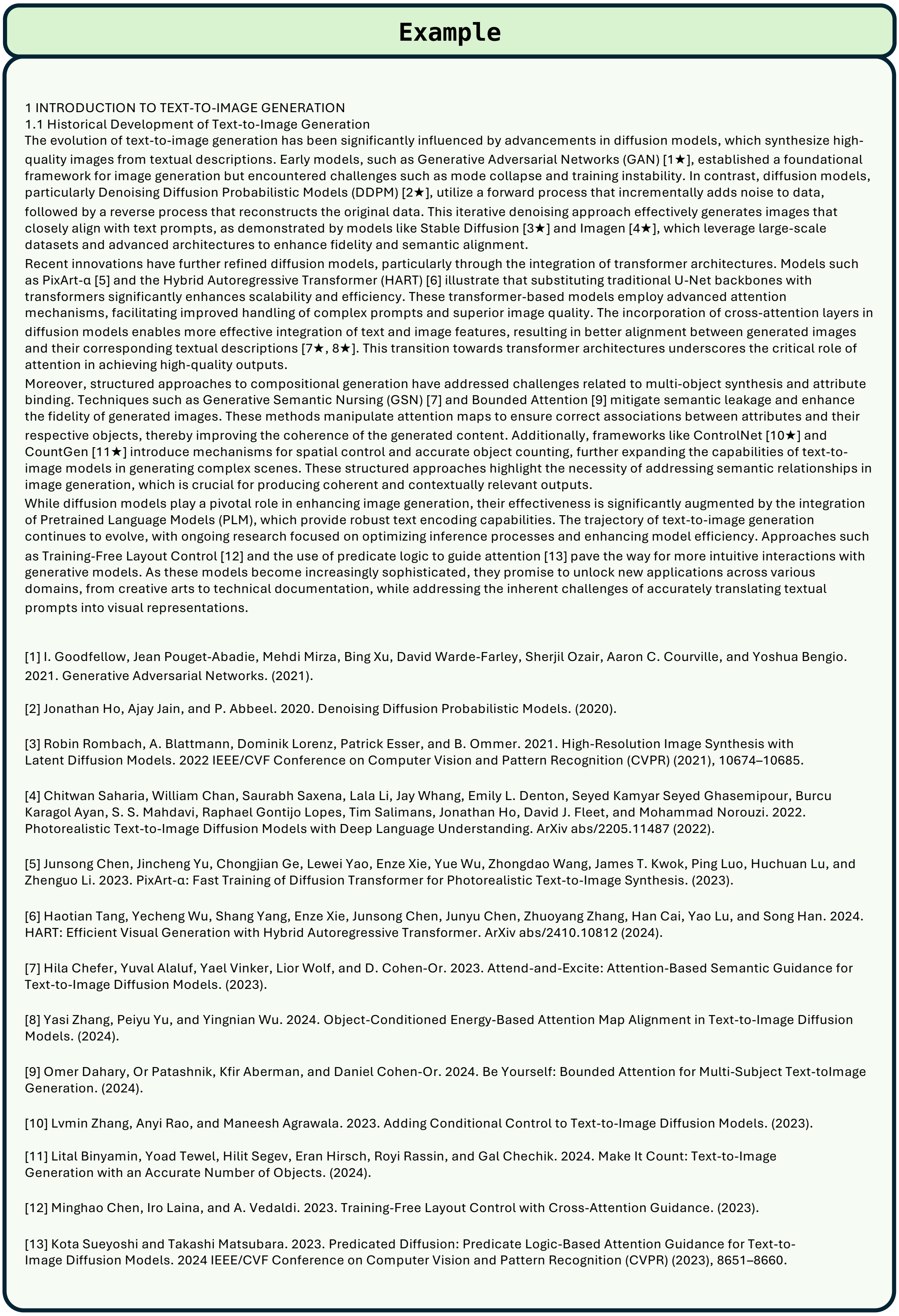
Figure 18: Full Subsection Writing with RAG and Citation Tracing prompt used in *SurveyGen-I*.



Evaluation Scoring Prompt

~~~
You are an **expert academic reviewer**. For the given technical survey, assign integer scores (1–5) for the following five dimensions. Your evaluation must be **rigorous**, **evidence-driven**, and adhere to the domain-neutral rubric below.
**Report Topic:** {topic}
---
### **1. Coverage**
**Does the survey comprehensively and selectively cover the major areas, topics, and subfields relevant to its scope, with a clear rationale for inclusion?**
* **5:** Covers both foundational and emerging areas with careful selection and justification; reflects deep understanding of the domain landscape.
* **4:** Covers most core topics and some recent developments; inclusion appears mostly thoughtful and relevant.
* **3:** Broad but shallow coverage; includes many topics but lacks prioritization, depth, or rationale.
* **2:** Touches on key areas superficially or with poor structure; lacks justification; several expected topics are missing.
* **1:** Misrepresents or omits major areas; appears outdated or arbitrarily constructed.
*Penalty:* Cap at **3** if long enumerations dominate without depth, rationale, or visual synthesis (e.g., figures, tables).
---
### **2. Relevance**
**Does the content consistently support the stated scope, objective, and framing of the survey?**
* **5:** Every section advances the survey’s declared purpose; even marginal topics are tightly integrated and justified.
* **4:** Content is strongly aligned; only occasional digressions.
* **3:** Mostly focused, but some sections feel loosely connected to the stated objective.
* **2:** Several off-topic or generic sections reduce clarity and focus.
* **1:** Large portions diverge from the intended focus or purpose.
*Penalty:* Cap at **3** if significant portions provide generic background rather than domain-specific insight.
---
### **3. Structure**
**Is the survey logically organized, well-sectioned, and progressively layered?**
* **5:** Organization is conceptually clear and layered; ideas build progressively; transitions and dependencies are handled well.
* **4:** Clear and readable structure; most sections flow logically.
* **3:** Reasonable outline, but weak layering or abrupt transitions between topics.
* **2:** Poor transitions; topics feel listed rather than structured; unclear progression.
* **1:** Disorganized or incoherent; difficult to follow.
*Penalty:* Cap at **3** if subsections follow a rigid template without conceptual grouping or progression.
---
### **4. Synthesis**
**Does the survey analyze and integrate prior work into meaningful categories, comparisons, or conceptual frameworks?**
* **5:** Synthesizes prior work into taxonomies, comparative tables, conceptual diagrams, or analytical frameworks; demonstrates interpretive insight.
* **4:** Some synthesis and comparison; related works are grouped or contrasted thoughtfully.
* **3:** Begins to group or compare work but lacks deep analysis.
* **2:** Minimal synthesis; mostly lists papers or methods independently.
* **1:** Pure enumeration; no analysis of relationships, tradeoffs, or trends.
*Reward:* Tables, diagrams, design spaces, or frameworks that aid comparative understanding.
*Penalty:* Cap at **3** if works are described without connections, comparisons, or grouping.
---
### **5. Consistency**
**Is the paper professionally written, with uniform tone, terminology, and formatting?**
* **5:** The writing is polished, coherent, and consistently professional. Terminology is clearly defined and used uniformly across sections; formatting and citation style are stable throughout. Tone is academically formal with no lapses. Glossaries or term explanations are provided or implicitly maintained.
* **4:** Generally consistent in tone and formatting with only isolated lapses (e.g., minor citation format drift, slightly inconsistent section intros). Terminology is mostly stable.
* **3:** Some inconsistencies in phrasing templates, formality, or terminology use. For example, some sections use generic introductions while others are more technical; citation styles or paragraph flow may shift noticeably.
* **2:** Frequent inconsistencies across sections in writing style, tone, or citation usage. Uses both informal and formal phrasing, or mixes different citation styles (e.g., inline vs. numbered). Terminology use is unstable or vague.
* **1:** Writing feels unprofessional or poorly edited. Style, terminology, formatting, or citation usage varies widely across the paper with no editorial control.
**Reward:** Glossaries, clearly defined and consistently used terminology, uniform structure and tone, consistent citation formatting.
**Penalty:** Cap at **3** if any of the following are present:
* Section templates feel mechanically reused without stylistic adaptation.
* Tone or phrasing switches between formal and informal.
* Inconsistent use of terminology (e.g., “framework”, “model”, “approach” used interchangeably without definition).
* Citation styles vary across sections (e.g., some use \[1], others use (Author, Year)).
---
### Output Format
Your evaluation should follow **two steps**:
---
#### **Step 1: Evaluation Notes**
For each criterion, write 1–2 sentences summarizing what the survey does well and what it lacks. Be specific. Avoid generalities.
### Review Policy
* Penalize: Template-based repetition, listing without depth, general background inflation, inconsistent phrasing.
* Reward: Selectivity, conceptual synthesis, structured comparisons, clear terminology.
* Use full range of scores (1–5), especially when reviewing papers with mixed quality.
#### **Step 2: JSON Scores**
Return a valid JSON dictionary like this:
```json
{{
 "coverage": <1,5>,
 "relevance": <1,5>,
 "structure": <1,5>,
 "synthesis": <1,5>,
 "consistency": <1,5>
}}
```
---
## Survey to Evaluate:
**Report Content:**
{content}
~~~

Figure 19: Evaluation Scoring prompt used in *SurveyGen-I*.



Example

1 INTRODUCTION TO TEXT-TO-IMAGE GENERATION
1.1 Historical Development of Text-to-Image Generation

The evolution of text-to-image generation has been significantly influenced by advancements in diffusion models, which synthesize high-quality images from textual descriptions. Early models, such as Generative Adversarial Networks (GAN) [1★], established a foundational framework for image generation but encountered challenges such as mode collapse and training instability. In contrast, diffusion models, particularly Denoising Diffusion Probabilistic Models (DDPM) [2★], utilize a forward process that incrementally adds noise to data, followed by a reverse process that reconstructs the original data. This iterative denoising approach effectively generates images that closely align with text prompts, as demonstrated by models like Stable Diffusion [3★] and Imagen [4★], which leverage large-scale datasets and advanced architectures to enhance fidelity and semantic alignment.

Recent innovations have further refined diffusion models, particularly through the integration of transformer architectures. Models such as PixArt-α [5] and the Hybrid Autoregressive Transformer (HART) [6] illustrate that substituting traditional U-Net backbones with transformers significantly enhances scalability and efficiency. These transformer-based models employ advanced attention mechanisms, facilitating improved handling of complex prompts and superior image quality. The incorporation of cross-attention layers in diffusion models enables more effective integration of text and image features, resulting in better alignment between generated images and their corresponding textual descriptions [7★, 8★]. This transition towards transformer architectures underscores the critical role of attention in achieving high-quality outputs.

Moreover, structured approaches to compositional generation have addressed challenges related to multi-object synthesis and attribute binding. Techniques such as Generative Semantic Nursing (GSN) [7] and Bounded Attention [9] mitigate semantic leakage and enhance the fidelity of generated images. These methods manipulate attention maps to ensure correct associations between attributes and their respective objects, thereby improving the coherence of the generated content. Additionally, frameworks like ControlNet [10★] and CountGen [11★] introduce mechanisms for spatial control and accurate object counting, further expanding the capabilities of text-to-image models in generating complex scenes. These structured approaches highlight the necessity of addressing semantic relationships in image generation, which is crucial for producing coherent and contextually relevant outputs.

While diffusion models play a pivotal role in enhancing image generation, their effectiveness is significantly augmented by the integration of Pretrained Language Models (PLM), which provide robust text encoding capabilities. The trajectory of text-to-image generation continues to evolve, with ongoing research focused on optimizing inference processes and enhancing model efficiency. Approaches such as Training-Free Layout Control [12] and the use of predicate logic to guide attention [13] pave the way for more intuitive interactions with generative models. As these models become increasingly sophisticated, they promise to unlock new applications across various domains, from creative arts to technical documentation, while addressing the inherent challenges of accurately translating textual prompts into visual representations.

[1] I. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2021. Generative Adversarial Networks. (2021).

[2] Jonathan Ho, Ajay Jain, and P. Abbeel. 2020. Denoising Diffusion Probabilistic Models. (2020).

[3] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and B. Ommer. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021), 10674–10685.

[4] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. S. Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. ArXiv abs/2205.11487 (2022).

[5] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James T. Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. 2023. PixArt-α: Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis. (2023).

[6] Haotian Tang, Yecheng Wu, Shang Yang, Enze Xie, Junsong Chen, Junyu Chen, Zhuoyang Zhang, Han Cai, Yao Lu, and Song Han. 2024. HART: Efficient Visual Generation with Hybrid Autoregressive Transformer. ArXiv abs/2410.10812 (2024).

[7] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and D. Cohen-Or. 2023. Attend-and-Excite: Attention-Based Semantic Guidance for Text-to-Image Diffusion Models. (2023).

[8] Yasi Zhang, Peiyu Yu, and Yingnian Wu. 2024. Object-Conditioned Energy-Based Attention Map Alignment in Text-to-Image Diffusion Models. (2024).

[9] Omer Dahary, Or Patashnik, Kfir Aberman, and Daniel Cohen-Or. 2024. Be Yourself: Bounded Attention for Multi-Subject Text-toImage Generation. (2024).

[10] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. (2023).

[11] Lital Binyamin, Yoad Tewel, Hilit Segev, Eran Hirsch, Royi Rassin, and Gal Chechik. 2024. Make It Count: Text-to-Image Generation with an Accurate Number of Objects. (2024).

[12] Minghao Chen, Iro Laina, and A. Vedaldi. 2023. Training-Free Layout Control with Cross-Attention Guidance. (2023).

[13] Kota Sueyoshi and Takashi Matsubara. 2023. Predicated Diffusion: Predicate Logic-Based Attention Guidance for Text-to-Image Diffusion Models. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023), 8651–8660.

Figure 20: An example of generated example with *SurveyGen-I*. References marked with an asterisk (*) indicate citations that have been explicitly traced and verified.