DeepSynth-Eval: Objectively Evaluating Information Consolidation in Deep Survey Writing

Anonymous ACL submission

Abstract

The evolution of Large Language Models (LLMs) towards autonomous agents has catalyzed progress in Deep Research. While retrieval capabilities are well-benchmarked, the post-retrieval synthesis stage—where agents must digest massive amounts of context and consolidate fragmented evidence into coherent, long-form reports—remains under-evaluated due to the subjectivity of open-ended writing.

To bridge this gap, we introduce DeepSynth-Eval, a benchmark designed to objectively evaluate information consolidation capabilities. We leverage high-quality survey papers as gold standards, reverse-engineering research requests and constructing "Oracle Contexts" from their bibliographies to isolate synthesis from retrieval noise. We propose a fine-grained evaluation protocol using General Checklists (for factual coverage) and Constraint Checklists (for structural organization), transforming subjective judgment into verifiable metrics. Experiments across 96 tasks reveal that synthesizing information from hundreds of references remains a significant challenge. Our results demonstrate that agentic "plan-then-write" workflows significantly outperform single-turn generation, effectively reducing hallucinations and improving adherence to complex structural constraints.

1 Introduction

The evolution of Large Language Models (LLMs) (DeepSeek-AI et al., 2025) marks a paradigm shift from fundamental token completion to sophisticated task completion via tool augmentation(Yao et al., 2023). This transition has notably accelerated progress in Deep Research(OpenAI, 2025; Google, 2024; Shao et al., 2024), where agents are tasked with not only retrieving information but also synthesizing it into comprehensive, survey-style reports. While Information Retrieval (IR) has robust benchmarks such as GAIA(Mialon et al., 2023), HLE(Phan et al., 2025) and others(Wei et al., 2025; Wu et al., 2025), the latter stage—Post-Retrieval Synthesis—remains under-evaluated. In this stage, models must digest massive amounts of retrieved contexts, consolidate fragmented evidence, and organize them into a coherent narrative. Evaluating this capability is challenging. Unlike short-form QA tasks like GAIA and HLE where answers can be checked against ground-truth facts, deep survey reports are long, lack a unique reference, and vary widely in content selection, structure, and writing style. Consequently, current practices rely heavily on subjective human judgments or LLM-as-a-judge scores(Zheng et al., 2023), which are costly, unstable, and prone to reward hacking(Wang et al., 2024a). Furthermore, end-to-end evaluations often conflate retrieval noise with synthesis failures, obscuring the model's true ability to consolidate information.

To objectively evaluate this information consolidation capability, we propose to assess deep synthesis performance via survey papers. High-quality survey papers can be viewed as expert-written research reports that serve as "gold standards" of synthesis. For each survey, we first "reverse-engineer" a corresponding research request and construct the input context from its bibliography. We then manually extract a set of checklists that specify what a good report must cover under this request, including representative methods, key design choices, important datasets, and other domain-specific points. These checklists transform open-ended report evaluation into verifying whether the model's output accurately covers the required items, enabling a more objective measurement of completeness and factual correctness.

Beyond factual coverage, a hallmark of deep synthesis is structural organization—for example, consolidating methods into comparative tables or grouping them into meaningful taxonomies. However, such organization is inherently subjective and

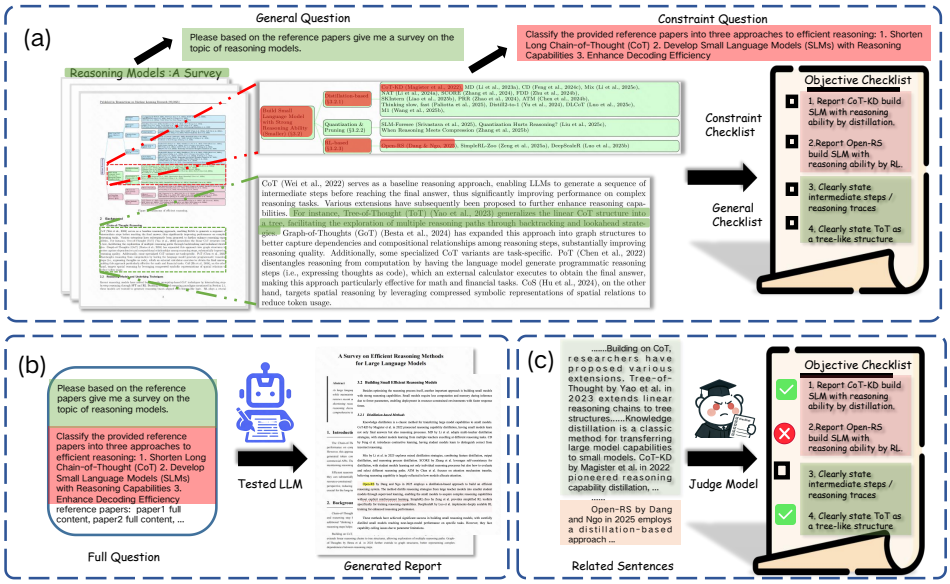

Figure 1: An illustration of DeepSynth-Eval. (a) From a reference survey, we derive a general prompt and constraint questions, and construct corresponding general/constraint checklists for evaluation. (b) The full question (prompt + constraints + reference papers) is fed to a tested LLM to generate a report. (c) A judge model verifies each checklist item on the generated report, converting subjective synthesis evaluation into objective item-wise verification.

flexible: there are diverse valid ways to categorize works or select attributes for comparison. Consequently, expecting a model to spontaneously replicate the exact structure of a reference survey is unreasonable. To resolve this ambiguity and make organizational abilities objectively comparable, we extract a canonical set of structural constraints from each survey (e.g., specific taxonomies to adopt or attributes to include in a table). By explicitly imposing these constraints in the synthesis prompt, we reduce editorial freedom, which allows us to compare the organization of content in corresponding sections. We refer to these requirements as Constraint Questions and the corresponding checklists as Constraint Checklists.

Building on these ideas, we construct **DeepSynth-Eval (DSE)**, a benchmark designed for the objective evaluation of **deep synthesis** capabilities. DSE comprises 96 queries derived from high-quality surveys across diverse domains. For each query, the benchmark provides the reconstructed research request, a comprehensive **Oracle Context** (constructed from the bibliography), and a set of fine-grained general and constraint checklists. Given a model-generated report, our pipeline automatically aligns the content with checklist items to compute coverage, precision, and structural constraint satisfaction scores. Compared to directly judging unstructured free-form text, this key-point verification reduces the knowledge burden on judge models, supports fine-grained and discriminative scoring, and makes the evaluation more robust and reproducible—providing a reliable foundation for training and improving deep synthesis systems.

Our main contributions are:

- **Objective Checklist Metric.** We address the challenge of evaluating long-form generation by introducing a checklist-based metric derived from gold-standard surveys. This method decomposes synthesis quality into verifiable factual coverage and structural constraint adherence, providing a robust standard for measuring deep research capabilities.
- **DeepSynth-Eval (DSE) Benchmark.** We introduce DSE, a benchmark comprising 96 tasks equipped with Oracle Contexts. This design isolates post-retrieval synthesis from retrieval noise, ensuring reproducibility and preventing ground-truth leakage.
- **Evaluation and Open Source.** We establish a rigorous baseline by benchmarking end-to-end and agentic workflows. We find that while agentic methods dominate, synthesizing context from over 100 references remains a formidable bottleneck. We open-source the entire framework to accelerate research in this direction.

2 Related Work

The paradigm of LLMs is shifting from closed-ended tasks—exemplified by benchmarks like MMLU (Hendrycks et al., 2021), GPQA (Rein et al., 2023), and Humanity's Last Exam (Wang et al., 2024b)—to complex **Deep Research**. Unlike these static evaluations, Deep Research requires integrating reading, searching, and writing. Existing works fall into three categories.

Long-Context Understanding Serving as the cognitive foundation for deep research, Long-Context Understanding has garnered significant attention; however, current evaluations are often confined to information extraction rather than logical restructuring. Existing benchmarks, such as LongBench (Bai et al., 2024), InfiniteBench (Zhang et al., 2024), and L-Eval (An et al., 2023), primarily assess the model's capacity for information retention within extended context windows. However, the task formulations in these benchmarks are largely restricted to "Needle-in-a-Haystack" scenarios or local extractive QA. In such settings, models can typically succeed by locating and reciting specific segments within the context, without the necessity for global association or cross-document reasoning to synthesize unstructured evidence dispersed across numerous documents.

Agentic Retrieval and Fact-Finding Transitioning from passive information processing to active information acquisition, Deep Search benchmarks (e.g., GAIA (Mialon et al., 2023), BrowseComp (Wei et al., 2025)) focus on evaluating an agent's ability to utilize tools for solving complex problems. However, these evaluations typically involve *low information throughput*, where retrieving a few key pages suffices. Consequently, they fall short of measuring the comprehensive ability to screen and synthesize massive information into coherent long-form reports. To address scale, WideSearch (Wong et al., 2025) targets breadth-oriented acquisition. Yet, despite the increased throughput, the task remains limited to **Structured Information Extraction** (e.g., compiling tables). This primarily tests recall in repetitive tasks, lacking the depth required for cross-document comparative analysis and the integration of heterogeneous information needed in deep research.

End-to-End Research System Evaluations Deep Research tasks inherently require agents to complete the full workflow from open-web searching to long-form report generation (Search-Read-Write). To align with this real-world application scenario, recent works have adopted an end-to-end evaluation paradigm; however, this setting introduces confounding variables that are difficult to isolate. For instance, PDR-Bench (Liang et al., 2025) and DeepResearchBench (Du et al., 2025) focus on personalization alignment and LLM-based subjective scoring for complex tasks, respectively, whereas ReportBench (Li et al., 2025) assesses the Search-Read-Write capabilities of end-to-end systems by reverse-engineering arXiv survey papers.

Despite these significant explorations in system-level assessment, this open-web evaluation methodology faces two major methodological challenges. The first challenge is the **Coupling of Retrieval and Generation**. The *dynamic drift of rankings* causes context variations over time, precipitating a *reproducibility crisis* where it is indistinguishable whether gains stem from model enhancements or better retrieval luck. Furthermore, retrieval noise (e.g., ads, broken links) prevents accurate measurement of the model's *intrinsic cognitive capability* in processing complex information. The second is the **Subjectivity of Evaluation**. Existing methods predominantly rely on LLMs to assign holistic scores (score-based) to reports, lacking objective verification of specific factual point coverage and remaining susceptible to biases inherent in the judge models.

3 Benchmark Construction

3.1 Design Philosophy

We propose to construct a benchmark utilizing high-quality survey papers as reference answers. Formally, given a written survey report R, we first reverse-engineer a query q_g for which the report R serves as the answer; we term this the **General Question** (q_g) (e.g., "Summarize recent advances in reasoning models").

Intuitively, one might evaluate the model's performance by directly comparing the consistency between the generated report $\hat{R} = \mathcal{M}(q_g)$ and the original survey R. However, in practice, directly comparing two lengthy, unstructured reports is computationally and metrically challenging. To address this, we further extract a set of **checklists** from R, where each item corresponds to a specific, verifiable fact. These checklists specify the essential content that a comprehensive report should cover under the given question q_g, thereby transforming the evaluation into a checklist verification task.

Survey writing is inherently subjective; different authors may adopt valid but distinct strategies to organize the same body of literature. For instance, benchmarks for Large Language Models (LLMs) could be categorized by the *capability domain* (e.g., reading comprehension, math, coding) or by the *task format* (e.g., multiple-choice questions vs. open-ended generation). Consequently,

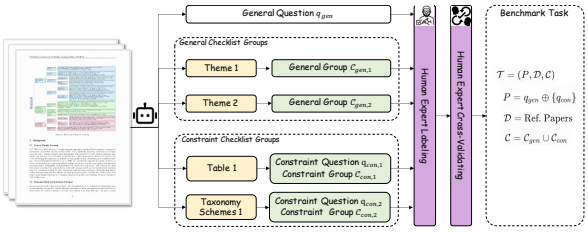

Figure 2: Overview of the DeepSynth-Eval construction process. Candidates generated by LLMs undergo human verification and editing to ensure quality.

without constraints, the structural organization of a generated report is often incomparable to the reference.

To address this, we extract specific structural requirements from the reference survey and append them as **Constraint Questions** (q_c) to the general prompt. For example, we might explicitly require the model to classify efficient reasoning methods into three specific categories: *(1) Shortening Chain-of-Thought, (2) Developing Small LLMs with reasoning capabilities, and (3) Enhancing decoding efficiency.* In this manner, we can objectively track whether the system correctly organizes representative works into the corresponding classes.

We term the extracted structural requirements as **Constraint Questions** (q_c) and the corresponding evaluation points as **Constraint Checklists**. Note that these checklists are valid for evaluation only when q_c is provided. By imposing q_c, we effectively reduce the editorial freedom of the synthesis task, rendering the results comparable. Furthermore, this design allows for a more rigorous assessment of the model's instruction-following capabilities within complex writing tasks.

Overall, given the composite input comprising the general question q_g and a set of constraint questions q_c, the model synthesizes a report by processing the reference bibliography of the original survey. Finally, an evaluation system verifies the generated report $\hat{R}$ against the comprehensive set of checklists $\mathcal{C} = \mathcal{C}_{gen} \cup \mathcal{C}_{con}$ to assess the satisfaction of each requirement.

3.2 Automated Construction Pipeline

The construction of this writing benchmark itself could be very labor consuming. In this paper, we utilize the LLM to generate candidates and with human labor to review and cross validate the dataset.

To construct the benchmark efficiently while maintaining high quality, we employ a multi-stage generation pipeline utilizing LLMs.

General Question Generation (q_{gen}). This step is straightforward. We treat the survey title and abstract as the core intent and employ an LLM to directly reverse-engineer the corresponding research request. This produces the general question q_{gen} (e.g., "Provide a comprehensive survey on ..."), which serves as the foundation of the task.

General Checklist Generation ($\mathcal{C}_{gen}$). Directly generating a comprehensive set of fine-grained checklists from a lengthy document is challenging for current models, often resulting in low recall or hallucinations. To mitigate this, we adopt a hierarchical **Topic Anchor** strategy. First, we prompt the LLM to synthesize several distinct *topics* based on the survey's section headers and content (e.g., "Core Algorithms", "Evaluation Metrics", "Future Directions"). Then, using these topics as anchors, we invoke the LLM to generate specific checklist items for each topic individually. This divide-and-conquer approach yields K groups of general checklists, denoted as $\mathcal{C}_{gen} = \bigcup_{k=1}^{K} \mathcal{C}_{gen,k}$.

Constraint Pair Generation ($q_{con}, \mathcal{C}_{con}$). For structural constraints, we employ a **Structure Anchor** approach to generate coupled pairs of constraint questions and checklists.

1. **Anchor Extraction:** The model first scans the survey to identify key organizational elements, such as specific taxonomies (classification schemes) or summary tables.
2. **Guidance Generation (q_c):** Based on an extracted anchor (e.g., a table summarizing reasoning datasets), the model generates a **Constraint Question** q_c that acts as explicit guidance. For instance, if the anchor is a dataset table, q_c might be: *"List a table summarizing recent reasoning datasets, specifying their data scale, question source, and answer source."*
3. **Checklist Extraction:** Simultaneously, the model extracts the corresponding cell values or classification criteria from the anchor to form a group of **Constraint Checklists** $\mathcal{C}_{con}$. This process yields M groups of constraint checklists, denoted as $\mathcal{C}_{con} = \bigcup_{m=1}^{M} \mathcal{C}_{con,m}$.

3.3 Human Verification

In the construction of DeepSynth-Eval, the automated workflow initially yielded 322 candidate tasks. To ensure benchmark reliability, we implemented a rigorous two-stage human verification process. The annotation team consisted of six experts specializing in computer science. Through the first round of manual screening and editing, we

retained 129 instances. Subsequently, following a second round of expert cross-validation and final refinement, we curated a final set of 96 high-quality instances to constitute the benchmark.

During the screening, we identified and discarded problematic instances characterized by common generation issues, such as: (1) Over-specification, where constraint questions were so detailed that the corresponding checklists became trivially inferable; and (2) Contextual Ambiguity, particularly in table-based extractions, where visual symbols (e.g., ✓ or ×) were extracted without sufficient context to permit verification.

3.4 Task Formalization

Formally, we define a deep synthesis evaluation task in DeepSynth-Eval as a tuple $\mathcal{T} = (P, \mathcal{D}, \mathcal{C})$, where:

- P denotes the **Unified Input Prompt**. It is constructed by concatenating the general question q_{gen} with a set of M constraint questions $\{q_{con}^{(1)}, \ldots, q_{con}^{(M)}\}$. Formally:
$$P = q_{gen} \oplus q_{con}^{(1)} \oplus \cdots \oplus q_{con}^{(M)}$$
where $\oplus$ represents the string concatenation operation.
- $\mathcal{D}$ represents the **Oracle Context**, containing the full text of the reference bibliography associated with the original survey. This serves as the closed-world knowledge boundary for the task.
- $\mathcal{C}$ is the **Evaluation Standard**, composed of the union of general checklist groups and constraint checklist groups: $\mathcal{C} = \mathcal{C}_{gen} \cup \mathcal{C}_{con}$.

3.5 Evaluation Metric

During inference, the model $\mathcal{M}$ receives the unified prompt P and the context $\mathcal{D}$ to synthesize a report $\hat{R} = \mathcal{M}(P, \mathcal{D})$. The quality of $\hat{R}$ is then quantified by assessing its satisfaction of the checklist set $\mathcal{C}$. Specifically, for each checklist item $c \in \mathcal{C}$, we employ a judge model to verify its presence in $\hat{R}$, computing a final score based on the satisfaction rate.

Specifically, the judge model evaluates each requirement and assigns a reward r based on three distinct verification states:

$$r = \begin{cases} 1, & \text{if mentioned correctly (Correctness);} \\ 0, & \text{if not mentioned (Omission);} \\ -1, & \text{if mentioned incorrectly (Hallucination).} \end{cases}$$

Fault-Tolerant Scoring. Given the open-ended nature of survey writing, exhaustive enumeration of every possible detail is often unnecessary; capturing the majority of representative works usually suffices for a high-quality report. To reflect this, we introduce a **saturation threshold** θ_k for each checklist group k.

Let N_k be the total number of items in group k, and S_k be the cumulative reward sum obtained by the generated report (aggregating rewards of $1, 0$ and -1). The normalized score for group k is calculated as:

$$\text{Score}_k = \min\left(1, \frac{S_k}{\theta_k}\right) \tag{1}$$

where $\theta_k \leq N_k$. This mechanism implies that once the model satisfies a sufficient number of requirements (i.e., $S_k \geq \theta_k$), it receives a full score for that group. For instance, in a checklist group listing $N_k = 10$ representative methods, we may set the threshold at $\theta_k = 8$, meaning that correctly covering 8 out of 10 key methods is considered "good enough" to achieve the maximum score.

Aggregated Metrics. To provide a fine-grained analysis of model performance, we report metrics across three dimensions: **General Score** ($\mathcal{S}_{gen}$), which measures the coverage of core research content; **Constraint Score** ($\mathcal{S}_{con}$), which evaluates adherence to structural instructions; and **Overall Score** ($\mathcal{S}_{all}$), which offers a holistic assessment.

Formally, let $\text{Score}_{gen,k}$ and $w_{gen,k}$ denote the score and weight of the k-th general group, and similarly for the constraint groups. The metrics are computed as weighted averages:

$$\mathcal{S}_{gen} = \frac{\sum_{k=1}^{K} w_{gen,k} \cdot \text{Score}_{gen,k}}{\sum_{k=1}^{K} w_{gen,k}} \times 100\%, \tag{2}$$

$$\mathcal{S}_{con} = \frac{\sum_{m=1}^{M} w_{con,m} \cdot \text{Score}_{con,m}}{\sum_{m=1}^{M} w_{con,m}} \times 100\%, \tag{3}$$

$$\mathcal{S}_{all} = \frac{\sum_{i=1}^{K+M} w_i \cdot \text{Score}_i}{\sum_{i=1}^{K+M} w_i} \times 100\%. \tag{4}$$

Here, $\mathcal{S}_{all}$ aggregates the union of all K general and M constraint groups, where w_i corresponds to the weight of the i-th group within this unified collection.

3.6 Statistics

We present benchmark statistics based on the checklist items and citations of each task.

Distribution of Checklist Items per Task: As illustrated in Figure 3, the dataset maintains a high

standard of evaluation granularity: on average, each task entails **199.7** checklist items, comprising **128.2** general items and **71.5** constraint items. This dense distribution underscores the complexity of the tasks, as models are required to satisfy a vast array of atomic facts and constraints to achieve a high score, effectively precluding the success of generic responses. By transforming the holistic evaluation of lengthy reports into fine-grained checklist matching, we convert an inherently subjective task into a quantifiable metric, significantly enhancing the objectivity of the report assessment.

Distribution of Total Reference Lengths: Figure 4 illustrates the distribution of total reference lengths per task, measured in words. Notably, the cumulative text length for a significant portion of tasks in DeepSynth-Eval exceeds 1 million words. This magnitude underscores the rigorous nature of the benchmark, placing substantial demands on the information processing, long-context retention, and summarization capabilities of the evaluated models.

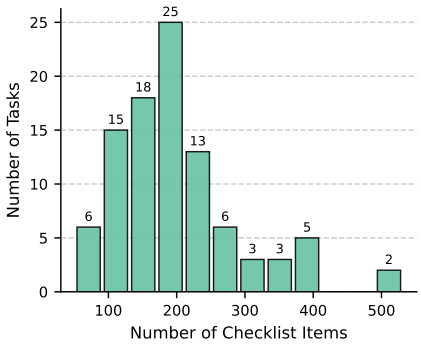

Figure 3: Distribution of checklist item counts

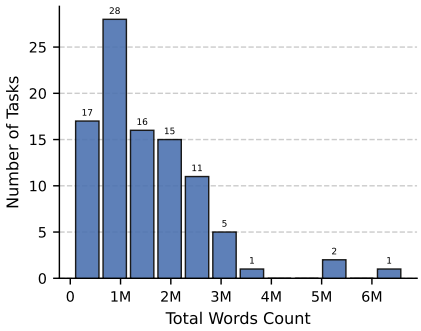

Figure 4: Distribution of total citation lengths

4 Experiments and Evaluation

4.1 Evaluation Settings

We aim to evaluate the deep synthesis capability of models, formalized as $\hat{R} = \mathcal{M}(P, \mathcal{D})$. As observed in Figure 4, the total length of the reference papers $\mathcal{D}$ for a significant portion of tasks exceeds 1 million words. Since the context window of most mainstream LLMs falls below 256k tokens—with only a few extending to 1 million—directly feeding the full context often results in context overflow.

Nevertheless, we believe that direct reading of such massive contexts is a challenging task that holds significant value for benchmarking emerging long-context capabilities and memory methasim. Unlike standard reading comprehension, which often relies on retrieving specific "needles in a haystack"(Liu et al., 2024; Hsieh et al., 2024), our report generation task requires high information throughput, making it a more challenging testbed for global context understanding.

To evaluate the deep synthesis ability of mainstream models within current constraints, we designed two representative scaffolds for report writing.

E2E Single-turn workflow. Instead of reading the full text of all reference papers $\mathcal{D}$, we utilize a pre-processing stage to summarize each document relevant to the writing request. We then assemble the Oracle Context by concatenating these summaries, each paired with a stable numeric citation ID. Finally, we append a unified instruction P—constructed by merging the general request with the extracted constraint questions—and the model generates the full survey report in a single pass, without an explicit intermediate planning stage or iterative refinement.

Agentic Multi-turn workflow. The Agentic Multi-turn workflow utilizes the same summarized Oracle Context but decomposes generation into coordinated stages—*Planning*, *Iterative Chapter Writing*, and *Global Polishing*—to strengthen organization and grounding. First, the model produces an *intellectual skeleton*—a high-level plan specifying section taxonomy, comparison axes, and content allocation—which is subsequently parsed into structured sub-tasks for each section. Second, for each sub-task, the model executes a 'read-then-write' loop starting with *selective deep reading*: it identifies critical papers based on task requirements and generates detailed reading notes by consulting their full text within the Oracle corpus, ensuring fine-grained details are captured without context overflow. Subsequently, the model writes each section conditioned on the skeleton, detailed notes, and previous sections, adhering to rules that encourage grouped comparisons and tabular consolidation. Finally, the model aggregates and polishes the sec-

Workflow	Model	Evaluation Metrics(%)			
		Overall	General	Constraint	Precision
-	Reference Survey	96.1	95.5	98.9	99.6
E2E Single-turn	Qwen3-30B-A3B-Thinking-2507	6.3	4.8	11.4	69.5
	Qwen3-235B-A22B-Thinking-2507	24.8	24.7	23.3	79.9
	DeepSeek-V3.2	23.4	22.2	27.0	79.8
	GPT-5.2	28.3	26.4	36.1	85.5
Agentic Multi-turn	Qwen3-30B-A3B-Thinking-2507	17.3	17.7	16.0	90.1
	Qwen3-235B-A22B-Thinking-2507	35.5	37.5	27.5	92.9
	DeepSeek-V3.2	30.4	31.4	27.0	91.4
	GPT-5.2	33.3	34.8	26.2	95.3
	GPT-5.2 (59/96 tasks subset)	37.0	39.0	28.4	95.0

Table 1: Main results under our two controlled synthesis workflows on DeepSynth-Eval.

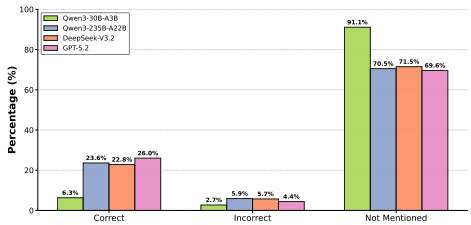

Figure 5: Checklist requirement status breakdown across four E2E models.

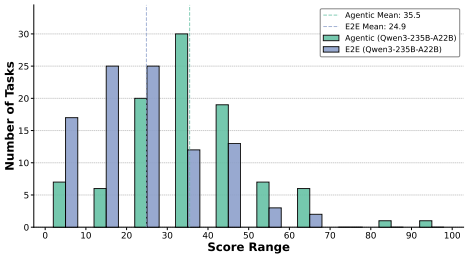

Figure 6: Score distribution of E2E versus Agentic workflows on Qwen3-235B-A22B.

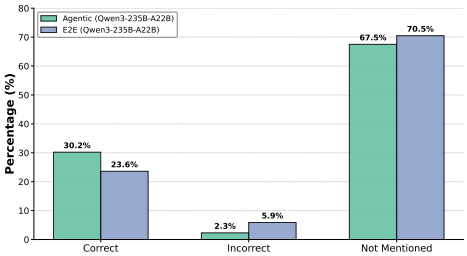

Figure 7: Checklist requirement status breakdown for E2E versus Agentic workflows on Qwen3-235B-A22B.

tions to ensure global coherence. Importantly, the key innovation lies in the dynamic, task-driven paper selection, which significantly improves grounding compared to single-turn generation.

Detailed textual descriptions of the implementation steps for both workflows are provided in Appendix A. In the evaluation, DeepSeek v3.2 is employed as the underlying model for checklist evaluation and summary reading in both workflows.

4.2 Results Analysis

Table 1 reports the main results under two controlled synthesis workflows—E2E single-turn and agentic multi-turn—with all models given the same Oracle Context. Our goal is to figure out how well a model can *synthesize* the Oracle corpus into a structured survey that satisfies both content coverage and formatting constraints. We focus on two aspects: (i) model-to-model differences under the single-turn baseline, which isolates post-retrieval consolidation in a single pass, and (ii) the effect of introducing multi-turn planning under the same model, which tests whether staged generation improves synthesis quality.

Judge Validity. We first validate the judge system using the Reference Survey (Row 1). It achieves a 96.1% Overall score, with the minor gap (<5%) primarily attributable to textual ambiguity and judge error. Notably, the near-perfect Precision (99.6%) confirms the judge's high fidelity, indicating that correct content is almost never erroneously penalized. These results establish a reliable upper bound for our evaluation.

Model comparison under the E2E single-turn workflow. Under the E2E single-turn setting, we observe large performance variance across models, revealing substantial differences in their ability to transform the same Oracle Context into a well-structured synthesis. GPT-5.2 achieves the strongest Overall score of 28.3 and leads by a clear margin on Constraint (36.1), while also obtaining the highest Precision (85.5). This pattern suggests that GPT-5.2 is comparatively better at *executing explicit structural instructions* and *grounding mentioned content* within a single pass. We further examine the checklist requirement status breakdown to understand where these gains come from. As shown in Figure 5, GPT-5.2 attains the highest frac-

tion of correctly satisfied requirements and the lowest incorrect rate among the E2E models, indicating fewer unsupported completions when mapping Oracle evidence to checklist items. Qwen3-235B and DeepSeek-V3.2 form a middle tier with similar Overall scores (24.8 and 23.4), but without a corresponding advantage on Constraint (23.3 and 27.0), indicating that their main bottleneck lies in *organizing and covering required content* rather than in factuality once content is produced. In contrast, Qwen3-30B largely collapses in the single-turn regime (Overall 6.3; General 4.8), consistent with Figure 5 where most requirements fall into the not-mentioned category, highlighting a strong capacity dependence for long-context consolidation.

Workflow effect: agentic multi-turn improves synthesis quality. Switching from single-turn generation to the agentic multi-turn workflow consistently improves Overall and General across all evaluated models: Qwen3-30B increases from 6.3→17.3, Qwen3-235B from 24.8→35.5, and GPT-5.2 from 28.3→33.3. We note that GPT-5.2 often fails to rewrite the full text during polishing, instead referring to the previous draft (Appendix D); excluding these invalid responses reveals a higher Overall score of 37.0 (Table 1).

We then perform a finer-grained distributional analysis over tasks to go beyond mean scores. As shown in Figure 6, the agentic workflow shifts a larger portion of tasks into higher score bins than E2E, indicating more consistent improvements across tasks rather than gains driven by a small subset. To further inspect subtle workflow differences, we analyze the checklist requirement status composition under the same model. Figure 7 shows that the agentic workflow increases the fraction of correctly satisfied requirements while reducing incorrect ones, suggesting that staging the process as *plan→reading→write* improves grounding and reduces unsupported completions. Overall, these results suggest that the agentic multi-turn workflow is generally beneficial for long-form synthesis: it helps models better allocate attention over full reference passages, produce more faithful content, and consequently achieve higher checklist pass rates in aggregate.

Concluding Remarks. Our analysis highlights three key insights. First, intrinsic model capacity is foundational, with stronger models consistently prevailing under identical workflows. Second, scaffolding design is pivotal; the substantial gains from the Agentic workflow demonstrate that *how* we structure generation is as critical as *which* model is used. Finally, with SOTA scores still below 40%, synthesizing grounded reports from over 100 documents remains a formidable open challenge.

4.3 Case Analysis

To complement checklist pass rates, we conduct a qualitative case analysis to identify recurring failure modes and to understand how workflow design affects post-retrieval synthesis behaviors.

We analysed the failure modes under the E2E single-turn workflow. Across low-scoring models, a common pattern is *under-elaboration*: instead of consolidating as many relevant points as possible from the Oracle Context, the model produces a high-level but sparse narrative that omits many checklist items. This behavior disproportionately harms the General score, as required coverage is missed due to insufficient detail and weak synthesis effort. Moreover, under constrained requirements, these models not only omit required organizational elements (e.g., missing categories/attributes) but also more frequently introduce *unsupported statements* when attempting to satisfy constraints, leading to apparent "hallucination"-like outputs (i.e., claims or comparisons not grounded in the provided materials).

5 Conclusion

We introduce **DeepSynth-Eval** to address the lack of objective benchmarks for long-form report generation. Instead of relying on subjective holistic scoring, we ground our evaluation in atomic, verifiable checklists derived from expert surveys. On average, each evaluation task involves 198.5 checklist items, providing a granular lens to quantify the quality of open-ended generation. Beyond the dataset, we establish a foundational evaluation framework by implementing and testing diverse baselines, ranging from standard end-to-end workflows to sophisticated agentic systems. Experiments demonstrate that even with agentic workflows, state-of-the-art (SOTA) models achieve scores of only around 37%, highlighting the challenging nature of this task and the significant room for improvement. Furthermore, this verifiable checklist mechanism paves the way for designing precise reward signals in long-form generation, offering a promising avenue for future reinforcement learning optimization.

Limitations

Currently, our dataset is predominantly composed of computer science surveys, a choice driven by data accessibility and annotation feasibility. However, the proposed checklist-based evaluation methodology is domain-agnostic and holds the potential to be extended to broader scientific disciplines.

Acknowledgments

We acknowledge the use of Gemini 3 to assist with linguistic polishing and refinement of the manuscript. The original content and ideas were authored by the researchers. All AI-suggested modifications were manually reviewed and verified by the authors to ensure strict consistency with the factual information and original intent.

References

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2023. L-eval: Instituting standardized evaluation for long context language models. *Preprint*, arXiv:2307.11088.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. *Preprint*, arXiv:2308.14508.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. 2025. Deepresearch bench: A comprehensive benchmark for deep research agents. *Preprint*, arXiv:2506.11763.

Google. 2024. Try deep research and our new experimental model in Gemini, your AI assistant. https://blog.google/products/gemini/google-gemini-deep-research/. Accessed: 2026-01-01.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Minghao Li, Ying Zeng, Zhihao Cheng, Cong Ma, and Kai Jia. 2025. Reportbench: Evaluating deep research agents via academic survey tasks. *Preprint*, arXiv:2508.15804.

Yuan Liang, Jiaxian Li, Yuqing Wang, Piaohong Wang, Motong Tian, Pai Liu, Shuofei Qiao, Runnan Fang, He Zhu, Ge Zhang, Minghao Liu, Yuchen Eleanor Jiang, Ningyu Zhang, and Wangchunshu Zhou. 2025. Towards personalized deep research: Benchmarks and evaluations. *Preprint*, arXiv:2509.25106.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *Preprint*, arXiv:2311.12983.

OpenAI. 2025. Introducing deep research. https://openai.com/index/introducing-deep-research/. Accessed: 2026-01-01.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, and 1093 others. 2025. Humanity's last exam. *Preprint*, arXiv:2501.14249.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *Preprint*, arXiv:2311.12022.

Yijia Shao, Yucheng Jiang, Theodore Kanell, Peter Xu, Omar Khattab, and Monica Lam. 2024. Assisting in writing Wikipedia-like articles from scratch with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6252–6278, Mexico City, Mexico. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024a. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *Preprint*, arXiv:2504.12516.

Ryan Wong, Jiawei Wang, Junjie Zhao, Li Chen, Yan Gao, Long Zhang, Xuan Zhou, Zuo Wang, Kai Xiang, Ge Zhang, Wenhao Huang, Yang Wang, and Ke Wang. 2025. Widesearch: Benchmarking agentic broad info-seeking. *Preprint*, arXiv:2508.07999.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025. WebWalker: Benchmarking LLMs in web traversal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10290–10305, Vienna, Austria. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

A Detailed Workflow Implementation

In this section, we provide a step-by-step textual description of the two synthesis workflows presented in Section 4.1, as illustrated in Figure 8.

A.1 Monolithic E2E Single-turn Workflow

The E2E workflow (Figure 8a) is designed to test the model's ability to perform long-context consolidation in a single pass. The process consists of two main steps:

1. **Oracle Context Construction:** First, we parallelize the processing of all bibliography entries. Each reference paper d_i is compressed into a dense prose summary s_i using a dedicated summarization prompt. These summaries are then concatenated to form the *Oracle Context*. Crucially, each summary is prefixed with a stable numeric citation identifier (e.g., "[1]", "[2]") to allow the model to ground its claims explicitly.
2. **One-Shot Generation:** We construct a unified prompt that combines the reconstructed research request, the extracted constraint checklists, and the *Oracle Context*. The model is then instructed to generate the full survey report in a single inference pass. No intermediate reasoning steps or external retrieval actions are permitted; the model must rely entirely on its internal attention mechanism to organize and synthesize the provided summaries.

A.2 Agentic Multi-turn Workflow

The Agentic workflow (Figure 8b) decomposes the synthesis task to mimic the *plan-read-write* process of a human researcher. It operates in three phases to ensure better organization and grounding:

Phase 1: Global Planning (Intellectual Skeleton). Instead of writing immediately, the model acts as an "Architect." It analyzes the *Oracle Context* (containing summaries) to construct an *Intellectual Skeleton*. This skeleton is a high-level plan that defines the section taxonomy, identifies key comparison axes, and allocates specific content to each section. This plan is then parsed into a structured list of sub-tasks (e.g., JSON objects), where each task corresponds to writing one section of the report.

Phase 2: Serial Iterative Writing Loop. The workflow executes a loop for each sub-task derived from the skeleton. For each section k, the process involves three specialized steps:

1. **Relevance Selection:** The model evaluates the requirements of the current section against the bibliography to identify a subset of critical papers (represented by IDs) that are necessary for drafting this specific part.
2. **Selective Deep Reading:** For the identified critical papers, the agent retrieves their full text (simulated within the Oracle corpus). It then generates detailed reading notes focused specifically on the current section's topic. This step allows the model to access fine-grained details that might have been lost in the initial summaries, without overflowing the context window with all papers at once.
3. **Context-Aware Writing:** The model writes the section. The generation is conditioned on three inputs: (1) the *Intellectual Skeleton* (for structure), (2) the *Deep Reading Notes* (for details), and (3) the *Writing History* (the draft of previous sections). This ensures the new section is logically coherent with the preceding text.

Phase 3: Final Polish. Once all sections are drafted, they are concatenated. An "Editor" agent performs a final pass to ensure global consistency, unify terminology, and format the document before outputting the final report.

B External System Evaluation

The controlled workflows in Table 1 isolate post-retrieval synthesis by holding the Oracle Context fixed. However, practical Deep Research deployments are typically end-to-end: systems actively search, browse, and write in a coupled loop. To contextualize our controlled findings and to understand how checklist-based synthesis evaluation manifests in real deployments, we additionally evaluate representative professional Deep Research systems and an external agentic pipeline. We emphasize that these results serve as complementary reference points rather than the primary benchmark ranking.

Fairness controls. Evaluating external systems introduces two confounds: uncontrolled retrieval may obscure synthesis failures, and the gold survey itself (or derivative pages) may be retrieved, leading to leakage and inflated scores. To mitigate these issues, we provide an explicit allowlist of reference URLs derived from the bibliography and

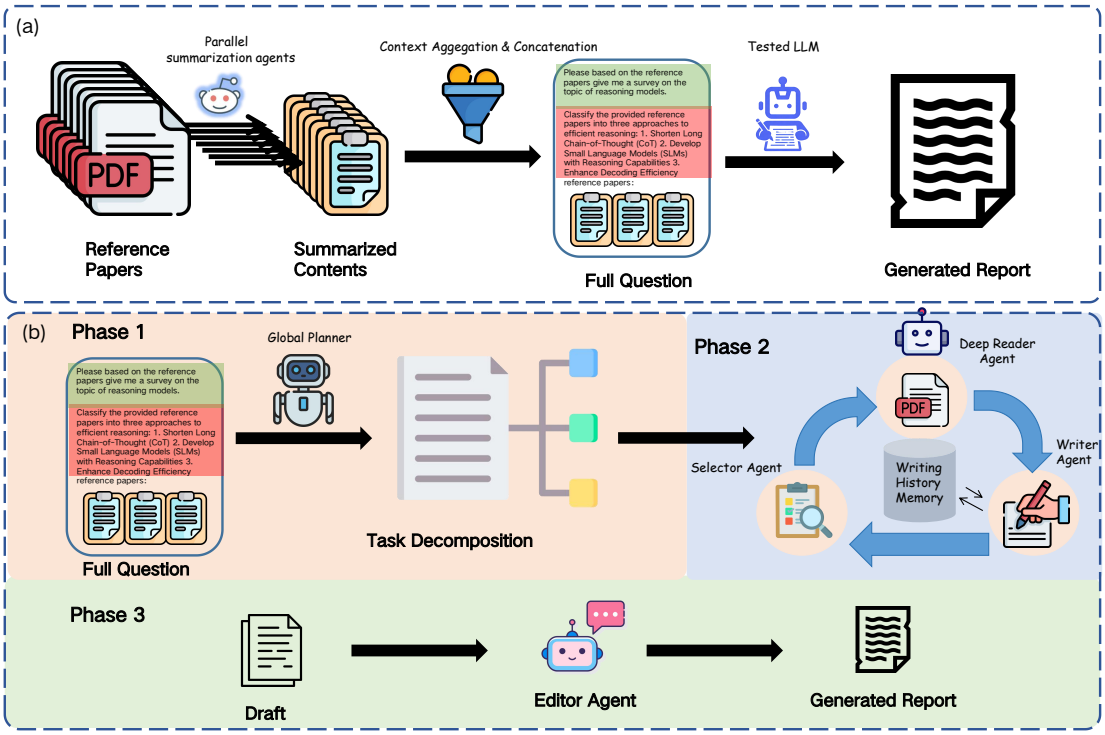

Figure 8: **Schematic overview of the controlled synthesis workflows.** (a) **E2E Single-turn workflow**: A streamlined pipeline that first summarizes all references into an *Oracle Context* with stable citation IDs, then concatenates the request and constraints to generate the full report in a single pass via long-context inference. (b) **Agentic multi-turn workflow**: A staged generation process. **Phase 1** generates an *Intellectual Skeleton* to plan taxonomy and content allocation. **Phase 2** executes a loop for each section: the model performs *selective deep reading* by identifying critical papers and consulting their full text to generate specific notes, followed by writing the section conditioned on the skeleton, notes, and previous writing history to ensure grounded organization.

instruct systems to use only these sources for evidence and citations, thereby preventing the original gold survey from being used as a reference. We further impose a strict temporal cutoff: only papers published before the survey date are allowed, and any later materials must be ignored even if surfaced by search.

Systems and evaluation. We evaluate two professional Deep Research systems accessed through their official interfaces (*Doubao DeepResearch*, *OpenAI DeepResearch*) and one external agentic pipeline (*OpenAI Agentic*). We interpret the results as contextual comparisons rather than direct, full-set rankings against our controlled workflows.

Results and discussion. Table 2 summarizes the external-system performance on the evaluation subset. Among the tested systems, *OpenAI DeepResearch* achieves the highest Overall score (28.7) and particularly strong constraint-checklist performance (Constr: 41.8), with *Doubao DeepResearch* showing a similar pattern (Overall: 26.4; Constr: 38.9). In contrast, the external agentic pipeline (*OpenAI Agentic*) attains a comparable Overall score (25.6) but notably lower Constr (28.3), suggesting that generic agentic tool use does not automatically translate into stronger constraint-driven organization.We observe that the evaluated end-to-end Deep Research systems do not consistently outperform our controlled workflows, despite having access to browsing and tool orchestration.

System	Overall	General	Constr	Prec. (%)
Doubao DeepResearch	26.4	23.3	38.9	87.4
OpenAI Agentic	25.6	24.0	28.3	83.3
OpenAI DeepResearch	28.7	25.1	41.8	85.4

Table 2: Performance of external systems on a subset of DSE tasks. Scores are checklist pass rates (%). Due to interface and execution constraints, these systems are evaluated on 39 tasks; results are reported separately from the main table and should be interpreted as contextual comparisons.

Why external Deep Research systems may underperform our controlled workflows. In our manual inspection, we observe that once the retrieval scope is constrained by the URL allowlist and the temporal cutoff, external Deep Research systems do not consistently utilize the full set of

provided references: reports often cite only a subset of the allowlisted papers, and some key references required by the checklists are never retrieved or never appear in the final citations. Correspondingly, the generated reports exhibit more frequent coverage gaps (missing required methods/datasets/attributes), which directly lowers checklist satisfaction.

We hypothesize that this underperformance is driven by two practical factors induced by the fairness constraints. First, restricted retrieval can become *incomplete*: even with an allowlist, the system may fail to open, read, or surface all relevant papers (e.g., due to interface limits, ranking/truncation, or budgeted browsing), causing evidence coverage gaps that propagate to the final synthesis. Second, to prevent leakage and ensure protocol compliance, our prompts impose additional instructions (e.g., strict source restrictions and citation rules), which may introduce an instruction-following burden that reduces flexibility and consumes generation capacity, leading to more conservative or less detailed synthesis.

Together, these observations highlight a key advantage of DSE's controlled Oracle-Context evaluation: it isolates consolidation and organization ability from retrieval completeness and interface-induced instruction overhead, enabling a more faithful measurement of post-retrieval synthesis.

C Prompt for Judge Model

Figure 9 presents the prompt supplied to the judge model, which guides the model in assessing the quality of the results generated by model under test based on given checklists.

D Case study of GPT 5.2

You are an expert evaluator.

=== Task ===
Your task is to assess whether the provided survey content meets the specific requirements listed in the criteria.

=== Input Data: Survey Content ===
{survey_content}

=== Input Data: Criteria ===
{criteria_text}

=== Evaluation Options ===
For each requirement listed in the **Criteria**, select exactly one of the following statuses:

- `mentioned_correct`: The requirement is present and accurate.
- `not_mentioned`: The requirement is absent.
- `mentioned_incorrect`: The requirement is present but incorrect or flawed.

=== Instructions ===
1. Analyze Step-by-Step: Read the survey and compare it against each requirement in the criteria list one by one.
2. Determine Status: Decide the appropriate status string for each requirement.
3. Verify Length: Ensure the final list contains exactly {len(subgroup["requirements"])} items.
4. Format Output: Output the result as a JSON list of strings.

=== Output Format ===
Only return a standalone JSON code block, with all the results in it.

=== Example format ===

```json
["mentioned_correct", "not_mentioned", "mentioned_incorrect"]
```

Figure 9: Prompt for judge model

Generated Report

Retrieval-Augmented Generation: Foundations, Methods, Multimodal Extensions, Systems, and Evaluation

1. Introduction and Formalization

Retrieval-Augmented Generation (RAG) provides a principled way to couple the parametric knowledge stored inside large language models with non-parametric, dynamically updatable external knowledge through retrieval and context curation. We formalize a generic RAG pipeline as y = G(x, C(R(x, K))), where x denotes the user query, R is a retriever that selects evidence from a knowledge base K, C is a context curation function that filters, orders, compresses, or structures the retrieved content for the generator, and G is the generator that produces the final output conditioned on both x and curated context. This formulation encompasses common operational regimes, including open-book and closed-book usage; single-hop and multi-hop question answering; and iterative procedures in which retrieval and generation interleave or recur. Treating retrieval outputs as latent variables that are marginalized at decoding time connects RAG to probabilistic formulations introduced in early hybrid models, while instantiations like Fusion-in-Decoder characterize specific architectural choices for integrating multiple retrieved passages into sequence-to-sequence generation [133, 125].

...

To guide readers, we include three artifacts. A taxonomy figure maps methods across the axes of retrieval type, integration strategy, adaptivity, efficiency, robustness, modality, structure, and domain, clarifying how designs cluster and where gaps remain [225, 283]. A pipeline schematic contrasts retrieve-then-generate, generate-then-retrieve, and iterative synergy workflows, highlighting where retriever training, query refinement, and context curation intervene [133, 84, 211]. A timeline from 2020 to 2024 situates key milestones—dense retrieval and REALM, probabilistic RAG and Fusion-in-Decoder, black-box and in-context augmentation, few-shot Atlas, and neural indexing—underscoring the rapid maturation and recent emphasis on adaptivity, risk certification, and multimodal expansion [165, 198, 133, 125, 77, 37, 205, 98, 20, 80]. Within this scope, the survey contributes a unified formalization, a structured taxonomy, and a synthesis of design trade-offs and evaluation constructs, with emphasis on recent advances that define the current frontier.

2. Retrieval Foundations

[Content unchanged from draft Section II.]

3. Integration Strategies and Generation Architectures

[Content unchanged from draft Section III.]

4. Robustness, Hallucination Mitigation, and Risk

[Content unchanged from draft Section IV.]

5. Efficiency, Latency, and System Co-Design

[Content unchanged from draft Section V.]

6. Multimodal RAG (Images, Audio, Video)

[Content unchanged from draft Section VI.]

7. RAG Meets Diffusion Models and Generative Modeling

Diffusion and score-based generative models provide the backbone on which retrieval-augmented conditioning can act, offering a spectrum of formulations that trade sampling speed, likelihood, and controllability. The core DDPM framework establishes forward and reverse Markov chains with a simplified noise-prediction objective that yields high-fidelity images and interpretable denoising trajectories [26]. Subsequent refinements improve both training and sampling: learned reverse-process variances with a hybrid variational objective and cosine noise schedules enable near-optimal sampling in tens of steps and competitive likelihoods, markedly improving practical efficiency [5]; the SDE-based unification generalizes DDPMs and denoising score matching...