OmniThink: Expanding Knowledge Boundaries in Machine Writing through Thinking

Anonymous ACL submission

Abstract

Machine writing with large language models often relies on retrieval-augmented generation. However, these approaches remain confined within the boundaries of the model's predefined scope, limiting the generation of content with rich information. Specifically, vanilla-retrieved information tends to lack depth, novelty, and suffers from redundancy, which negatively impacts the quality of generated articles, leading to shallow, unoriginal, and repetitive outputs. To address these issues, we propose OmniThink, a slow-thinking machine writing framework that emulates the human-like process of iterative expansion and reflection. The core idea behind OmniThink is to simulate the cognitive behavior of learners as they slowly deepen their knowledge of the topics. Experimental results demonstrate that OmniThink improves the knowledge density of generated articles without compromising metrics such as coherence and depth. Human evaluations and expert feedback further highlight the potential of OmniThink to address real-world challenges in the generation of long-form articles.

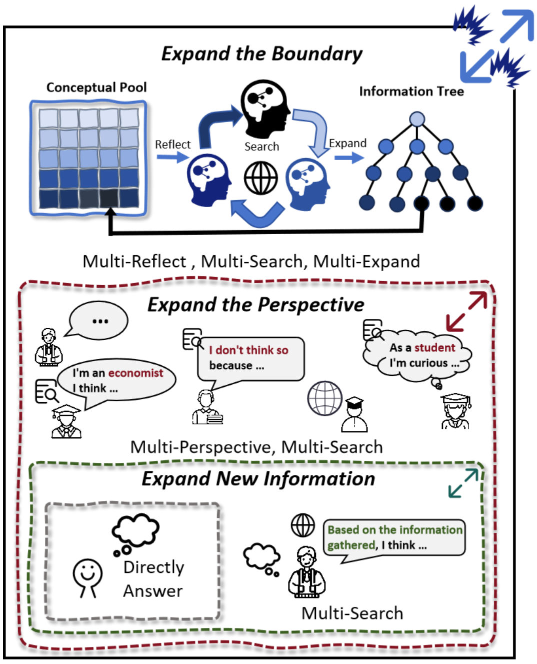

Expand the Boundary
Conceptual Pool
Information Tree
Reflect
Search
Expand
Multi-Reflect , Multi-Search, Multi-Expand
Expand the Perspective
...
I'm an economist I think ...
I don't think so because ...
As a student I'm curious ...
Multi-Perspective, Multi-Search
Expand New Information
Directly Answer
Based on the information gathered, I think ...
Multi-Search

Figure 1: Previous machine writing approaches only expand new information or perspective via RAG and role-playing. OmniThink expands knowledge boundaries through continuous reflection and exploration, attaching knowledge to an information tree and extracting it into a conceptual pool to deepen understanding and uncover more in-depth content.

1 Introduction

Writing is a continuous process of collecting information and thinking (Bean and Melzer, 2021). Recent advances in Large Language Models (LLMs) have demonstrated remarkable progress in machine writing such as open domain long-form generation (Liang et al., 2023; Yang et al., 2023; Zhao et al., 2024) or report generation on specific topics (Liu et al., 2018). To seek useful information, as shown in Figure 1, early attempts use Retrieval Augmented Generation (RAG) to expand new information on a given topic (Gao et al., 2024; Edge et al., 2024). However, vanilla RAG relies on a fixed set of search strategies (Ram et al., 2023), which lack diversity in generation, preventing a thorough exploration of the topic and resulting in a fragmented and incomplete understanding of the subject (Spink et al., 1998). To address this issue, STORM (Shao et al., 2024) and Co-STORM (Jiang et al., 2024) have proposed a role-play approach designed to expand the perspective, which means collecting information from multiple perspectives, thus broadening the information space (Shen et al., 2023; Shanahan et al., 2023; Parmar et al., 2010). Yet these approaches are still being thought within the scope of one's own role, making it difficult to generate deep content and break through one's own knowledge boundaries(Ji et al., 2025). In particular, retrieved information often lacks depth, novelty and redundancy, directly affecting the quality of generated articles, resulting in shallow, repetitive, and unoriginal outputs (Skarlinski et al., 2024).

Note that humans can naturally avoid such pitfalls in the writing process. This phenomenon can be explained through the theory of reflective practice, a concept rooted in cognitive science (Osterman, 1990). According to this theory, human writers continuously reflect on previously gathered information and personal experiences, allowing them to reorganize, filter, and refine their cognitive framework. This process prompts writers to iteratively adjust their writing direction and mental pathways, ultimately allowing human authors to generate more profound, nuanced and original content (Bruce, 1978).

Motivated by this, we propose OmniThink, a new machine writing framework that emulates the human-like cognitive process. The core idea behind OmniThink is to simulate the cognitive behavior of learners as they gradually deepen their understanding of complex topics to expand knowledge boundaries. We introduce two innovative components, information tree and conceptual pool, to simulate the process of collecting information and structuring cognition during human iterative learning. Through continuous expansion and reflection, these components are enriched. Once a diverse set of information has been gathered and structured, OmniThink transitions to the stages of outline construction and article generation. This iterative thinking process leads to the production of articles of higher quality that contain a higher knowledge density of useful, insightful, and original content. OmniThink is model-agnostic and can be integrated with existing frameworks.

We evaluate OmniThink on the WildSeek datasets (Jiang et al., 2024) based on previous metrics as well as a new metric, named knowledge density. Experimental results demonstrate that OmniThink enhances the knowledge density of generated articles without compromising key metrics such as coherence and depth. To conclude, our main contributions are as follows:

- We propose OmniThink, a novel writing framework that emulates the human slow-thinking process.

- We propose a new metric, Knowledge Density (KD), which measures the proportion of useful information in an article.

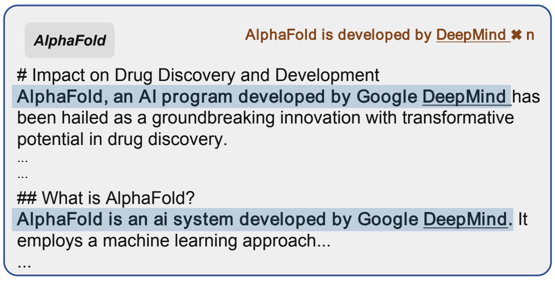
- We analyze the challenges of current long-form generation methods from a novel knowledge boundary perspective, investigate the underlying factors contributing to the effectiveness of OmniThink, and propose a new direction for future long-form generation research.

2 Background

2.1 Task Definition

We focus on the task of open-domain long-form generation for machine writing, which retrieving information from an open domain and synthesizing it into a coherent article (Fan et al., 2019; Su et al., 2022; Quan et al., 2024). Given an input topic T, the target of open-domain long-form generation is to generate a long article $\mathcal{A}$. The current standard approach involves two major steps (Zhang et al., 2019; Zheng et al., 2023): *(i)* Use a search engine $\mathcal{S}$ to retrieve information $\mathcal{I} = \mathcal{S}(\mathrm{T})$ which is related to the topic T; *(ii)* Generate an outline $O = \mathrm{Generate}(\mathcal{I}, \mathrm{T})$ based on the retrieved information $\mathcal{I}$ and input topic T. Finally, the article is generated using the outline, expressed as $\mathcal{A} = \mathrm{Generate}(O, \mathcal{I})$.

2.2 Revisiting Previous Methods



AlphaFold

AlphaFold is developed by DeepMind ✖ n

# Impact on Drug Discovery and Development
AlphaFold, an AI program developed by Google DeepMind has been hailed as a groundbreaking innovation with transformative potential in drug discovery.
...
...

## What is AlphaFold?
AlphaFold is an ai system developed by Google DeepMind. It employs a machine learning approach...
...

Figure 2: A case generated by STORM using GPT-4o on the topic of AlphaFold. We have marked the repeated expressions in the article regarding "AlphaFold is developed by DeepMind".

Previous works have made numerous efforts to improve the quality of open-domain long-form generation. Co-STORM (Jiang et al., 2024) introduces a user-participatory roundtable discussion in step *(i)* to enhance the diversity of the retrieved information. STORM (Shao et al., 2024) proposes a questioning mechanism to improve the quality and relevance of the generated outlines in step *(ii)*.

Although substantial progress has been made in open-domain long-form generation, a persistent challenge remains: the generated content frequently suffers from **redundancy** and **lacks novelty**. We present a case generated by

Feature	STORM	Co-STORM	OmniThink
Dynamic retrieval	✗	✗	✓
Structured memory	✗	✓	✓
Reflective thinking	✗	✗	✓

Table 1: Comparison of different methods. For more detailed explanations, please refer to the appendix G.

STORM (Shao et al., 2024) with GPT-4o as the backbone, as shown in Figure 2. In this article, the well-known phrase "AlphaFold was developed by DeepMind" appears multiple times, whereas it could be stated only once in the initial mention.

2.3 Limitation Analysis From A Boundary Perspective

As discussed in Section 2.1, open-domain long-form generation relies on retrieved information to composite the article. From a boundary perspective, redundancy can be analyzed in two aspects. First, when the retrieved content contains *limited factual knowledge*, the available information for generating the text is constrained, leading to redundancy in the generated article (Lewis et al., 2021). Second, even when a large amount of non-redundant factual knowledge is retrieved, the model cannot organize and structure the knowledge as humans do to effectively utilize it, resulting in a limited amount of usable information and, consequently, redundancy (Xia et al., 2024). Similarly, the lack of novelty can be attributed to either the failure to collect novel knowledge or the inability to use the retrieved novel knowledge effectively.

In summary, the challenges in open-domain long-form generation can be abstracted into two knowledge boundary issues: the Knowledge Information Boundary and the Knowledge Cognition Boundary.

3 OmniThink

We introduce a machine writing framework OmniThink, which emulates the human slow-thinking process, as shown in Figure 3.

3.1 Information Acquisition

While LLMs have learned vast amounts of human knowledge through training, they may struggle to capture the spontaneous processes by which humans organize useful information and update cognitive frameworks when learning new knowledge (Riva et al., 2024; Chemero, 2023). To address this, we propose two novel components: the **Information Tree** $\mathcal{T}$ and the **Conceptual Pool** $\mathcal{P}$ to simulate the human process of acquiring knowledge and updating cognitive frameworks (Wu et al., 2025b). Through interactive expansion and reflection, as shown in Figure 3, these components are iteratively enriched, expanding the knowledge boundaries of open-domain long-form generation.

Initialization The interactive process begins with the initialization of a root node based on the input topic T. OmniThink first utilizes search engines, *e.g.*, Google, or Bing, to retrieve information related to T, using the retrieved information to construct the initial root node of the information tree N_r. This initial information in N_r is then analyzed and extracted to form a preliminary conceptual pool $\mathcal{P}_0$, which serves as OmniThink's foundational cognition of the topic and guides subsequent expansion processes.

3.1.1 Expansion of Information Tree

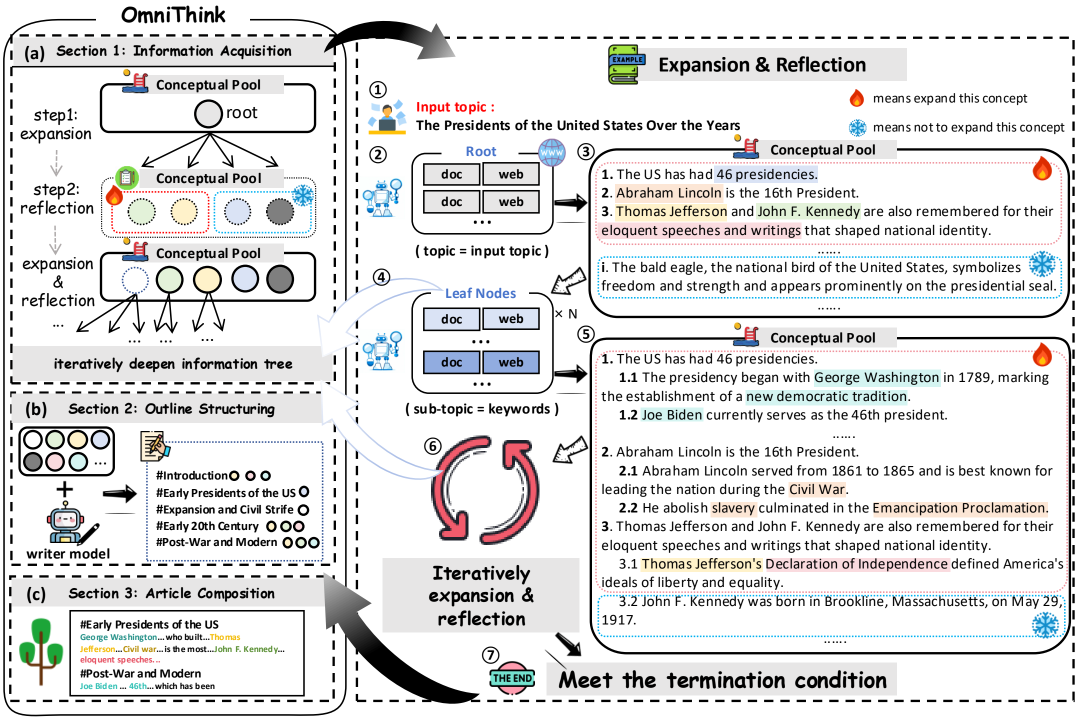
At time step m, OmniThink analyzes all leaf nodes $L_m = \{N_0, N_1, \ldots, N_n\}$ of the information tree $\mathcal{T}_m$. For nodes that need expansion, OmniThink uses the current conceptual pool $\mathcal{P}_m$ to identify areas for deeper expansion or suitable directions for expansion. For each leaf node N_i, OmniThink generates k_{N_i} sub-nodes, denoted as $\text{SUB}(N_i) = \{S_0, S_1, \ldots, S_{k_{N_i}}\}$, for expansion. Each sub-node represents a specific aspect or subtopic identified from the current node N_i. For each sub-node, OmniThink retrieves relevant information and stores it within the respective node, subsequently adding the sub-node to the appropriate position in the updated information tree $\mathcal{T}_{m+1}$ as follows:

$$\mathcal{T}_{m+1} = \text{Combine}(\mathcal{T}_m, \text{SUB}(N_0), \ldots, \text{SUB}(N_n)) \tag{1}$$

This targeted retrieval process ensures that OmniThink collects comprehensive and in-depth knowledge for each sub-node, thereby enriching the hierarchical structure of the information tree.

3.1.2 Reflection of Conceptual Pool

In this phase, OmniThink reflects the newly retrieved information in all leaf nodes $L_{m+1} = \{N_0, ...N_n\}$ to update its cognitive framework, which is represented as conceptual pool. The information from leaf nodes is analyzed, filtered, and synthesized to distill the core insights $I_{m+1} = \{\text{INS}_0, ..., \text{INS}_n\}$. These distilled insights are then incorporated into the conceptual pool $\mathcal{P}_m$, which is continuously updated and enriched throughout


OmniThink
(a) Section 1: Information Acquisition
Conceptual Pool
root
step1: expansion
step2: reflection
Conceptual Pool
Conceptual Pool
expansion & reflection
...
iteratively deepen information tree
(b) Section 2: Outline Structuring
writer model
#Introduction
#Early Presidents of the US
#Expansion and Civil Strife
#Early 20th Century
#Post-War and Modern
(c) Section 3: Article Composition
#Early Presidents of the US
George Washington...who built...Thomas Jefferson...Civil war...is the most...John F. Kennedy...
eloquent speeches...
#Post-War and Modern
Joe Biden ... 46th...which has been
Expansion & Reflection
EXAMPLE
means expand this concept
means not to expand this concept
①
Input topic :
The Presidents of the United States Over the Years
②
Root
doc
web
doc
web
www
(topic = input topic)
③
Conceptual Pool
④
Leaf Nodes
doc
web
doc
web
× N
(sub-topic = keywords)
⑤
Conceptual Pool
⑥
Iteratively expansion & reflection
⑦
THE END
Meet the termination condition

1. The US has had 46 presidencies.
2. Abraham Lincoln is the 16th President.
3. Thomas Jefferson and John F. Kennedy are also remembered for their eloquent speeches and writings that shaped national identity.
......
i. The bald eagle, the national bird of the United States, symbolizes freedom and strength and appears prominently on the presidential seal.
......

1. The US has had 46 presidencies.
1.1 The presidency began with George Washington in 1789, marking the establishment of a new democratic tradition.
1.2 Joe Biden currently serves as the 46th president.
......
2. Abraham Lincoln is the 16th President.
2.1 Abraham Lincoln served from 1861 to 1865 and is best known for leading the nation during the Civil War.
2.2 He abolish slavery culminated in the Emancipation Proclamation.
3. Thomas Jefferson and John F. Kennedy are also remembered for their eloquent speeches and writings that shaped national identity.
3.1 Thomas Jefferson's Declaration of Independence defined America's ideals of liberty and equality.
3.2 John F. Kennedy was born in Brookline, Massachusetts, on May 29, 1917.
......

Figure 3: The overview of OmniThink. As shown in the left diagram, OmniThink is mainly divided into three steps: (a) Information Acquisition, (b) Outline Structuring, and (c) Article Composition. The right diagram illustrates the specific operations during the Information Acquisition step. (① - ②) denotes the initialization of Information Acquisition, (② - ③) corresponds to the reflection, and (③ - ④) indicates the expansion.

the process as follows:

$$\mathcal{P}_{m+1} = \text{Merge}(I_{m+1}, \mathcal{P}_m) \qquad (2)$$

Using the updated conceptual pool $\mathcal{P}_{m+1}$, which represents the LLM's expanded cognition boundary on the topic, OmniThink further expands the leaf nodes of the information tree iteratively.

The iterative cycle of expansion and reflection continues until OmniThink determines that sufficient information has been acquired or the predefined maximum retrieval depth K is reached. More details about the termination conditions can be found in Appendix J. During this process, as the Information Tree and Conceptual Pool are continuously expanded, the Information Boundary and Cognition Boundary are progressively expanded.

3.2 Concept-guided Outline Structuring

The outline determines the content direction, structural hierarchy, and logical progression of an article. To create an outline that is well-guided, clearly structured, and logically coherent, it is essential to have a comprehensive and in-depth cognition of the topic. In the previous section, OmniThink maintains a conceptual pool that essentially represents the cognition boundary of the LLM. When generating the content outline, we first create a draft outline O_D, and then ask the LLM to refine and link the content from the conceptual pool $\mathcal{P}$, ultimately forming the final outline $O = \text{Polish}(O_D, \mathcal{P})$. Through this approach, the LLM is able to comprehensively cover the key points of the topic in the outline and ensure logical consistency and content coherence in the article.

3.3 Article Composition

After completing the outline O, we begin writing for each section S. At this stage, the LLM would work in parallel for each section. When writing the content of the section, we use the titles of each section and their hierarchical subsections to retrieve the most relevant K documents from the information tree by calculating the semantic similarity (Sentence-BERT (Reimers and Gurevych, 2019) embeddings). After obtaining the relevant information, the LLM is prompted to generate the section content with citations based on the retrieved information. Once all sections are generated, they will be concatenated into a complete draft article $\mathcal{A}_D = \{S_1, ..S_n\}$. Since these sections are generated in parallel and the specific content of other sections is not yet clear, we prompt the LLM to process the concatenated article, remove redundant information, and form the final article $\mathcal{A} = \{S_1^{'}, ..S_n^{'}\}$.

4 Experiments

4.1 Dataset and Baseline

We use WildSeek as evaluation dataset to verify the effectiveness of our method, following previous work (Jiang et al., 2024). WildSeek includes 100 data points across 24 different domains with each data consisting of a specific topic and a user's intend. We select representative baselines for comparison, including RAG, oRAG, and STORM (Shao et al., 2024) and Co-STORM (Jiang et al., 2024). The baseline results are reproduced on the basis of STORM[1].

4.2 Knowledge Density Metric

Previous works mostly focus on whether the article is relevant and correct, but do not consider whether the article is sufficiently concise and free of redundancy (Li et al., 2024; Que et al., 2024; Liu et al., 2024). Many generated articles contain a lot of redundant information, which is very inconsistent with human writing. To quantify this, we introduce the Knowledge Density (KD) for the generated article, which is defined as the ratio of meaningful content to the overall volume of text (Xu and Reitter, 2017) as:

$$KD = \frac{\sum_{i=1}^{N} \mathcal{U}(k_i)}{L} \quad (3)$$

where N is the total number of atomic knowledge units identified within the document. The function $\mathcal{U}(k_i)$ indicates whether the i-th unit information k_i is unique. L represents the total length of the text.

In the appendix H, we empirically demonstrate the effectiveness of the KD metric. Readers encountering low KD content often experience fatigue, frustration, or disengagement due to redundant or irrelevant details. In contrast, high-density content provides a streamlined experience, enabling efficient knowledge transfer.

4.3 Evaluation Setup

We use Prometheus2 (Kim et al., 2024)[2] to automaticly score articles on a scale of 0 to 5, evaluating Relevance, Breadth, Depth, and Novelty. Furthermore, we measure information diversity (Jiang et al., 2024) (cosine similarity differences between web pages) and knowledge density (discussed in detail in §4.2) for information richness. Detailed procedures are provided in the Appendix B. In addition, we also conduct a detailed human evaluation. The implementation details and evaluation results can be found in Appendix C.

4.4 Implementation Details

We build OmniThink based on the DSpy framework (Khattab et al., 2023), and Appendix A.2 contains the corresponding prompts we used. During generation, we set the *temperature* at 1.0 and *top_p* at 0.9. We use Bing's API with the parameter for the number of web pages returned per query set to 5. For the computation of knowledge density, we utilize Factscore[3] with `GPT-4o-08-06` as the backbone to decompose atomic knowledge (Min et al., 2023). After decomposition, we proceed to use GPT-4o-08-06 for the deduplication of the split atomic knowledge. To avoid the impact of search engine changes over time. More implementation details are presented in Appendix A.1.

4.5 Main Results

Article Generation. Table 2 presents the evaluation results on WildSeek dataset. Within the framework of four grading criteria (Relevance, Breadth, Depth, and Novelty) OmniThink excels across all metrics, particularly standing out in Novelty. This achievement can be attributed to OmniThink's Information Tree and Conceptual Pool, which are continuously enriched, enabling OmniThink to expand the boundaries of existing knowledge.

OmniThink utilizes the Conceptual Pool for multidimensional deep thinking on the retrieved information during the retrieval process, enabling subsequent searches to access deeper levels of external knowledge, thereby enhancing the diversity of information.

In terms of knowledge density, OmniThink employs a continuous and dynamic retrieval strategy, storing a wealth of information in the Information Tree. This allows OmniThink to draw upon a broader range of resources during the content generation phase, positioning OmniThink at a distinct advantage in the knowledge density metric compared to existing benchmark methods.

[1] https://github.com/stanford-oval/storm

[2] https://github.com/prometheus-eval/prometheus-eval

[3] https://github.com/shmsw25/FActScore

Backbones	Methods	Rubric Grading				Information Diversity	Knowledge Density
		Relevance	Breadth	Depth	Novelty		
			Conversational Models				
GPT-4o	RAG	4.65	4.55	4.59	4.22	0.1042	22.11
	oRAG	2.38	3.63	2.56	2.27	0.0963	19.70
	STORM	4.34	4.21	4.21	3.80	0.6342	19.33
	Co-STORM*	4.37	4.66	4.65	3.89	0.6285	19.53
	OmniThink	**4.77**	**4.71**	**4.66**	**4.31**	**0.6642**	**22.31**
Qwen-Plus	RAG	2.63	2.82	2.93	2.21	0.0927	10.32
	oRAG	2.42	2.52	2.66	2.22	0.1032	11.31
	STORM	2.72	2.81	3.00	2.72	0.6417	10.28
	Co-STORM*	3.26	3.10	3.07	2.73	0.5332	11.52
	OmniThink	**4.00**	**3.92**	**4.06**	**3.38**	**0.7230**	**11.66**
			Reasoning Models				
O1-preview	RAG	3.99	4.13	4.02	3.44	0.1065	10.49
	oRAG	2.49	3.03	2.89	2.55	0.1222	10.51
	STORM	3.26	3.22	3.44	2.56	0.6121	10.82
	Co-STORM*	3.41	3.29	3.23	2.97	0.6347	10.33
	OmniThink	**4.20**	**4.20**	**4.32**	**3.60**	**0.6752**	**10.87**
DeepSeek-R1	RAG	4.12	4.33	4.55	4.44	0.1044	11.32
	oRAG	4.56	4.49	4.39	4.37	0.1123	10.44
	STORM	2.42	2.93	3.14	2.86	0.6640	11.57
	Co-STORM*	4.62	4.54	4.78	4.47	0.5332	11.66
	OmniThink	**4.70**	**4.78**	**4.78**	**4.59**	**0.6653**	**11.72**

Table 2: Results of article quality evaluation. * means that this method is different from the original experimental setting, primarily in the human-machine collaboration component. Instead of simulating human involvement through an agent, as done in the original paper (Jiang et al., 2024), we remove the human participation step. The variance of evaluation can be found in Appendix B.3.

Method	Content Guidance	Hierarchical Clarity	Logical Coherence
oRAG	3.93	3.95	3.97
STORM	3.92	3.99	3.99
Co-STORM*	3.45	3.27	3.41
OmniThink	**4.00**	**4.02**	**3.99**

Table 3: Results of outline quality evaluation.

Outline Generation. We evaluate outline quality from the perspectives of structural soundness, logical consistency, and generative guidance. More evaluation details can be found in the Appendix B.1. From Table 3, we notice that OmniThink achieves superior performance. This improvement can be attributed to the unique design of OmniThink's Conceptual Pool, which enables the LLMs to develop a more comprehensive and diverse understanding of the target topic during outline generation.

5 Analysis

5.1 Ablation Study

Information tree and Conceptual pool Ablation. For the Information Tree, we remove the hierarchical structure and instead have the OmniThink reflect over all retrieved content directly, followed by another retrieval. In contrast, to evaluate the Conceptual Pool, we disable reflection and allow the Information Tree to grow continuously until the maximum depth of Information tree is reached. As shown in Figure 6(a) and Figure 6(b), the performance of OmniThink degrades when either the Information Tree or the Conceptual Pool is removed.

Expansion and Reflection Ablation. We compare OmniThink with a version that does not implement expansion and reflection. As shown in Figure 6(c), w/o E&R performs worse in all metrics than the complete system, particularly in terms of Information Diversity and Novelty.

5.2 Boundary Analysis

As discussed in Section 2.3, we divide the boundary into Information Boundary and Cognition Boundary. In this section, we explore in detail whether OmniThink has truly expanded these boundaries.

Information Boundary. To investigate whether OmniThink has truly expanded the Information Boundary, we map the retrieval information of OmniThink, STORM, and Co-STORM to a two-dimensional plane as their Information Boundary to visualize the scope. As shown in Figure 4, OmniThink has the largest retrieval scope, indicating that it has indeed expanded the Information Boundary through the information tree and conceptual pool. More implementation details can be found in Appendix E.

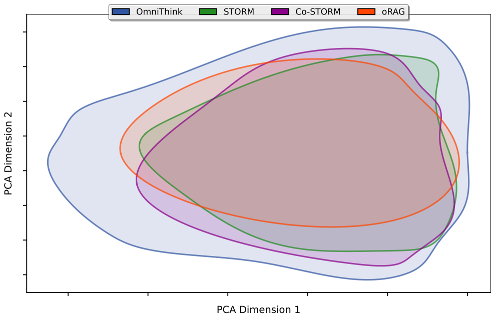


Figure 4: The information scope of OmniThink, Co-STORM, STORM and oRAG.

Cognition Boundary. For the Cognition Boundary, since Expansion and Reflection cannot be separated, we set a new baseline, oRAG-Plus, where we increase the number of web pages retrieved by oRAG-Plus to match that of OmniThink. From Figure 5, it can be observed that without the guidance of the Conceptual Pool, even with a large amount of information, the LLM still fails to utilize it effectively. In fact, some of the results of oRAG-Plus are even lower than those of oRAG, which may be due to the lack of sufficient cognition to utilize the retrieved information, with excessive web content acting as noise to the model.

5.3 Expansion & Reflection Analysis

Cognitive boundary mainly constrain the potential for innovation. To further analyze how the expansion and reflection processes shape various aspects of the final article through the conceptual pool and information tree, we design an indirect yet ingenious experiment. As shown in Figure 6(b), we use lower-performing models to complete the expansion and reflection processes, with the decline in various metrics serving as an indicator of their impact on the article. The details of the experimental design can be found in Appendix F. We observe that reflection is much more important for novelty. As discussed in Section 5.2, OmniThink indeed expands the knowledge boundary. Reflection endows the model with the ability not only to re-evaluate and introspectively consider existing knowledge but also to integrate this information in a way that promotes the emergence of more diverse and expansive ideas, which is similar to our definition of the cognition boundary. Expanding the cognition boundary through Reflection significantly enhances the model's innovation in generating articles. Therefore, we believe that it is the cognition boundary that limits the model's writing innovation.

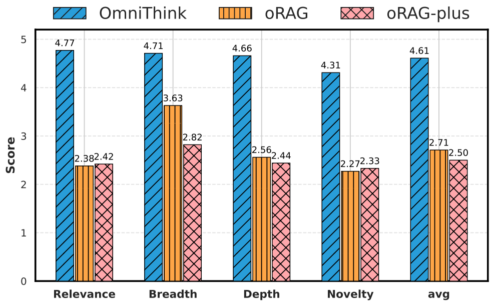


Figure 5: The Comparison of results between OmniThink, oRAG, and oRAG-plus.

Information boundary limits the effective organization of information on the topic. We notice that expansion is more important than reflection in Knowledge Density, Breadth, and Depth. The rationale behind this is that expansion inherently sets the trajectory for the model's subsequent information retrieval. By establishing more precise and effective directions for the model's retrieval process, it becomes more adept at harnessing the retrieved information to expand the information boundary. This integration not only enhances the relevance of the content but also increases the knowledge density, as the text becomes more comprehensive and nuanced. Consequently, a better expansion strategy leads to a more sophisticated planner, capable of navigating the complexities of information retrieval and utilization with greater finesse.

More knowledge boundaries need to be identified and defined. Previous experiments have shown that expansion and reflection extend the information boundary and cognition boundary, which improves the quality of the articles. We increase the depth of expansion and reflection to explore how far they can extend the knowledge boundary. From Figure 6(c), we observe that as the depth increases, the growth rate of knowledge density and information diversity significantly slows down. This indicates that the information boundary and cognition boundary are no longer the primary limi-

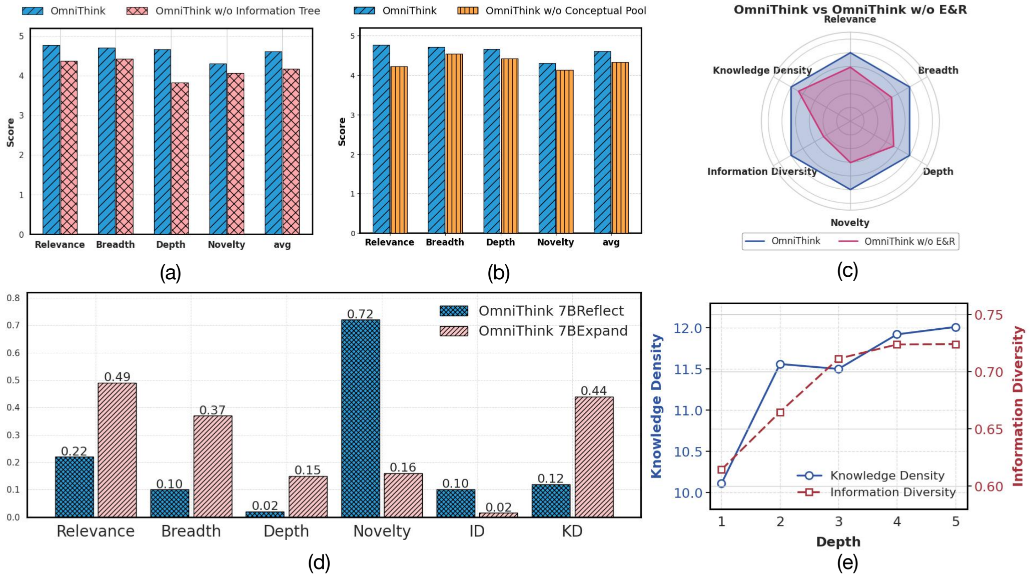


Figure 6: (a) The Ablation of Conceptual Pool; (b) The Ablation of Information Tree; (c) The Ablation of OmniThink, OmniThink w/o E&R represents a version of OmniThink without expansion and reflection ; (d) The comparison of the impact of expansion and reflection on various metrics, OmniThink 7BReflect indicates the use of Qwen2.5-7b-instruct for Reflection. More details can be found in Appendix F ; (e) The result of depth analysis.

tations on article quality, and other boundaries need to be identified and defined.

6 Related Work

6.1 Information Seeking in NLP

Previous studies on information-seeking focused on designing question-answering (QA) systems (Wu et al., 2025a). Early open-domain QA methods generally assumed that users could fulfill their information needs through a single query (Chen et al., 2017; Levy et al., 2021). Subsequent studies have recognized that, in real-world scenarios, users often struggle to satisfy their information needs with a single query (Chen et al., 2017; Levy et al., 2021). To address this limitation, researchers have explored *multi sub-query* retrieval methods, where a single query is decomposed into multiple sub-queries to retrieve distinct pieces of information (Mao et al., 2024; Chen et al., 2011; Peng et al., 2019). The information collected is then aggregated to provide a comprehensive answer. Building on these developments, recent advances in open-domain long-form generation require reasoning across multiple information sources (Fan et al., 2019; Ujwal et al., 2024; Wei et al., 2024; Tan et al., 2024). This line of open-domain long-form generatio underscores the importance of integrating information from multiple perspectives.

6.2 Machine Writing

Due to the high costs associated with manual writing, machine writing has garnered significant research interest in recent years (Zhou et al., 2023; Pham et al., 2024; Wang et al., 2024a,b,c). The emergence of LLMs and Retrieval-Augmented Generation (RAG) has opened new possibilities for automated writing (Liang et al., 2024; Balepur et al., 2023; de la Torre-López et al., 2023). To ensure authenticity and real-time relevance, current RAG-based automated writing systems primarily rely on retrieved content to generate articles. For example, STORM (Shao et al., 2024) introduces a role-playing question-and-answer approach to author Wikipedia-like articles, while Co-STORM (Jiang et al., 2024) proposes a user-participated information retrieval paradigm.

7 Conclusion and Future Work

We propose OmniThink, a machine writing framework that emulates the human-like process of iterative expansion and reflection. Automatic and human evaluations demonstrate that OmniThink can generate well-founded, high-quality long articles. OmniThink is model-agnostic and can be integrated with existing frameworks. In the future, we will explore more advanced machine writing methods that combine deeper reasoning with human-computer interaction.

Limitations

Although the proposed OmniThink has demonstrated its advantages in both automatic and human evaluations, several limitations remain. Firstly, the current work is limited to search and text generation, while a vast amount of multimodal information in the open domain remains unused. Secondly, we have not considered personalized language styles in text production. As a result, the generated texts tend to be academic in nature, which may not be as suitable for general users' reading preferences. We plan to address these limitations in future work.

References

Nishant Balepur, Jie Huang, and Kevin Chen-Chuan Chang. 2023. Expository text generation: Imitate, retrieve, paraphrase. Preprint, arXiv:2305.03276.

John C Bean and Dan Melzer. 2021. Engaging ideas: The professor's guide to integrating writing, critical thinking, and active learning in the classroom. John Wiley & Sons.

Bertram C Bruce. 1978. A cognitive science approach to writing. Center for the Study of Reading Technical Report; no. 089.

Anthony Chemero. 2023. Llms differ from human cognition because they are not embodied. Nature Human Behaviour, 7(11):1828–1829.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. Preprint, arXiv:1704.00051.

Gang Chen, Yongwei Wu, Jia Liu, Guangwen Yang, and Weimin Zheng. 2011. Optimization of sub-query processing in distributed data integration systems. Journal of Network and Computer Applications, 34(4):1035–1042.

José de la Torre-López, Aurora Ramírez, and José Raúl Romero. 2023. Artificial intelligence to automate the systematic review of scientific literature. Computing, 105(10):2171–2194.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. arXiv preprint arXiv:2404.16130.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. arXiv preprint arXiv:1907.09190.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. Preprint, arXiv:2312.10997.

Yixin Ji, Juntao Li, Hai Ye, Kaixin Wu, Jia Xu, Linjian Mo, and Min Zhang. 2025. Test-time computing: from system-1 thinking to system-2 thinking. Preprint, arXiv:2501.02497.

Yucheng Jiang, Yijia Shao, Dekun Ma, Sina J. Semnani, and Monica S. Lam. 2024. Into the unknown unknowns: Engaged human learning through participation in language model agent conversations. Preprint, arXiv:2408.15232.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. Preprint, arXiv:2310.03714.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. Preprint, arXiv:2405.01535.

Sharon Levy, Kevin Mo, Wenhan Xiong, and William Yang Wang. 2021. Open-domain question-answering for covid-19 and other emergent domains. Preprint, arXiv:2110.06962.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. Preprint, arXiv:2005.11401.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, Yuxuan Lai, Chongyang Tao, and Shuai Ma. 2024. Leveraging large language models for nlg evaluation: Advances and challenges. Preprint, arXiv:2401.07103.

Xiaobo Liang, Zecheng Tang, Juntao Li, and Min Zhang. 2023. Open-ended long text generation via masked language modeling. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 223–241, Toronto, Canada. Association for Computational Linguistics.

Yi Liang, You Wu, Honglei Zhuang, Li Chen, Jiaming Shen, Yiling Jia, Zhen Qin, Sumit Sanghai, Xuanhui Wang, Carl Yang, and Michael Bendersky. 2024. Integrating planning into single-turn long-form text generation. Preprint, arXiv:2410.06203.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. Preprint, arXiv:1801.10198.

Xiang Liu, Peijie Dong, Xuming Hu, and Xiaowen Chu. 2024. Longgenbench: Long-context generation benchmark. Preprint, arXiv:2410.04199.

Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2024. RaFe: Ranking feedback improves query rewriting for RAG. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 884–901, Miami, Florida, USA. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. Preprint, arXiv:2305.14251.

Karen F Osterman. 1990. Reflective practice: A new agenda for education. Education and urban society, 22(2):133–152.

Bidhan L Parmar, R Edward Freeman, Jeffrey S Harrison, Andrew C Wicks, Lauren Purnell, and Simone De Colle. 2010. Stakeholder theory: The state of the art. Academy of Management Annals, 4(1):403–445.

Peng Peng, Qi Ge, Lei Zou, M Tamer Özsu, Zhiwei Xu, and Dongyan Zhao. 2019. Optimizing multi-query evaluation in federated rdf systems. IEEE Transactions on Knowledge and Data Engineering, 33(4):1692–1707.

Chau Minh Pham, Simeng Sun, and Mohit Iyyer. 2024. Suri: Multi-constraint instruction following for long-form text generation. arXiv preprint arXiv:2406.19371.

Shanghaoran Quan, Tianyi Tang, Bowen Yu, An Yang, Dayiheng Liu, Bofei Gao, Jianhong Tu, Yichang Zhang, Jingren Zhou, and Junyang Lin. 2024. Language models can self-lengthen to generate long texts. Preprint, arXiv:2410.23933.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, Junran Peng, Zhaoxiang Zhang, Songyang Zhang, and Kai Chen. 2024. Hellobench: Evaluating long text generation capabilities of large language models. Preprint, arXiv:2409.16191.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. Preprint, arXiv:2302.00083.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Giuseppe Riva, Fabrizia Mantovani, Brenda K. Wiederhold, Antonella Marchetti, and Andrea Gaggioli. 2024. Psychomatics – a multidisciplinary framework for understanding artificial minds. Preprint, arXiv:2407.16444.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. Nature, 623(7987):493–498.

Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. 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 and Short Papers).

Zejiang Shen, Tal August, Pao Siangliulue, Kyle Lo, Jonathan Bragg, Jeff Hammerbacher, Doug Downey, Joseph Chee Chang, and David Sontag. 2023. Beyond summarization: Designing ai support for real-world expository writing tasks. Preprint, arXiv:2304.02623.

Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. 2024. Language agents achieve superhuman synthesis of scientific knowledge. Preprint, arXiv:2409.13740.

Amanda Spink, Howard Greisdorf, and Judy Bateman. 1998. From highly relevant to not relevant: examining different regions of relevance. Information processing & management, 34(5):599–621.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Read before generate! faithful long form question answering with machine reading. Preprint, arXiv:2203.00343.

Haochen Tan, Zhijiang Guo, Zhan Shi, Lu Xu, Zhili Liu, Yunlong Feng, Xiaoguang Li, Yasheng Wang, Lifeng Shang, Qun Liu, and Linqi Song. 2024. Proxyqa: An alternative framework for evaluating long-form text generation with large language models. Preprint, arXiv:2401.15042.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Utkarsh Ujwal, Sai Sri Harsha Surampudi, Sayantan Mitra, and Tulika Saha. 2024. " reasoning before responding": Towards legal long-form question answering with interpretability. In Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, pages 4922–4930.

Qiyao Wang, Shiwen Ni, Huaren Liu, Shule Lu, Guhong Chen, Xi Feng, Chi Wei, Qiang Qu, Hamid Alinejad-Rokny, Yuan Lin, and Min Yang. 2024a. Autopatent:

A multi-agent framework for automatic patent generation. Preprint, arXiv:2412.09796.

Tiannan Wang, Jiamin Chen, Qingrui Jia, Shuai Wang, Ruoyu Fang, Huilin Wang, Zhaowei Gao, Chunzhao Xie, Chuou Xu, Jihong Dai, et al. 2024b. Weaver: Foundation models for creative writing. arXiv preprint arXiv:2401.17268.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min Zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. 2024c. Autosurvey: Large language models can automatically write surveys. In The Thirty-eighth Annual Conference on Neural Information Processing Systems.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, et al. 2024. Long-form factuality in large language models. arXiv preprint arXiv:2403.18802.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025a. Webwalker: Benchmarking llms in web traversal. Preprint, arXiv:2501.07572.

Junde Wu, Jiayuan Zhu, and Yuyuan Liu. 2025b. Agentic reasoning: Reasoning llms with tools for the deep research. Preprint, arXiv:2502.04644.

Peng Xia, Kangyu Zhu, Haoran Li, Hongtu Zhu, Yun Li, Gang Li, Linjun Zhang, and Huaxiu Yao. 2024. RULE: Reliable multimodal RAG for factuality in medical vision language models. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 1081–1093, Miami, Florida, USA. Association for Computational Linguistics.

Yang Xu and David Reitter. 2017. Spectral analysis of information density in dialogue predicts collaborative task performance. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 623–633, Vancouver, Canada. Association for Computational Linguistics.

Kevin Yang, Dan Klein, Nanyun Peng, and Yuandong Tian. 2023. Doc: Improving long story coherence with detailed outline control. Preprint, arXiv:2212.10077.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, and Xueqi Cheng. 2019. Outline generation: Understanding the inherent content structure of documents. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 745–754.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. A survey of large language models. Preprint, arXiv:2303.18223.

Wenqing Zheng, SP Sharan, Ajay Kumar Jaiswal, Kevin Wang, Yihan Xi, Dejia Xu, and Zhangyang Wang. 2023. Outline, then details: Syntactically guided coarse-to-fine code generation. In International Conference on Machine Learning, pages 42403–42419. PMLR.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Peng Cui, Tiannan Wang, Zhenxin Xiao, Yifan Hou, Ryan Cotterell, and Mrinmaya Sachan. 2023. Recurrentgpt: Interactive generation of (arbitrarily) long text. Preprint, arXiv:2305.13304.

A OmniThink Details

A.1 Implementation

We build OmniThink based on the DSpy framework (Khattab et al., 2023), and STORM. Appendix A.2 contains the corresponding prompts we used. During article generation, we set the *temperature* at 1.0 and *top_p* at 0.9. The search engine employed is Bing's API, with the parameter for the number of web pages returned per query configured to 5. To retrieve information based on the outline, we use SentenceBERT (Reimers and Gurevych, 2019) embeddings to calculate cosine similarity, thereby retrieving the three most similar web pages each time. For the computation of knowledge density, we utilize Factscore[4] with GPT-4o-08-06 as the backbone to decompose atomic knowledge (Min et al., 2023). After the decomposition, we proceed to use GPT-4o-08-06 for the deduplication of the split atomic knowledge.

A.2 Full Prompts in OmniThink

In §3, we introduce the specific process of OmniThink, which is implemented using zero-shot prompting based on GPT-4o-2024-08-06. Lists 1, 2, 3, 4 and 5, respectively document the complete prompts for OmniThink's Expand, Reflect, Write Outline, Write Article, and Polish Article stages. These prompts are designed to guide the model through iterative stages of content generation, ensuring coherence and depth in the produced text.

The structured process leverages dynamic adjustments based on intermediate outputs, reflecting a balanced integration of retrieval and generation capabilities. This systematic approach highlights OmniThink's ability to adaptively construct well-organized and contextually relevant articles across diverse topics.

B Automatic Evaluation Details

To further ensure reliability, we conducted multiple evaluation rounds using different prompts covering various aspects of outline coherence, structural logic, and topic relevance. This multi-faceted evaluation helps mitigate potential biases and enhances the robustness of the scoring results.

B.1 Outline Evaluation

Since Prometheus2 (Kim et al., 2024) does not perform targeted optimization on the outline, we decided to use a more powerful model to score the outline. To ensure the results are consistent, we set the temperature to 0. Specifically, we use the Prometheus2 framework but replace the underlying evaluation model with GPT-4o-08-06. The scoring criteria for outline quality evaluation and discourse quality evaluation can be found in Lstlisting 9. In addition, since Co-STORM does not have an intermediate outline generation step, we had to extract the outline from the final article for evaluation, which might be the reason for the relatively lower outline scores observed form Co-STORM.

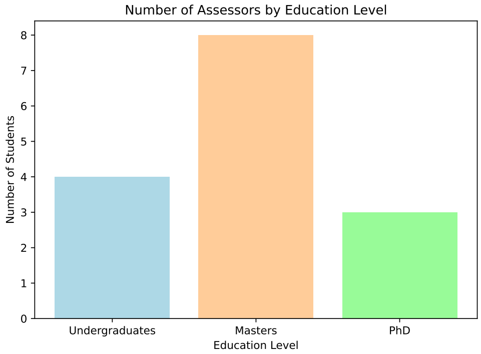


Figure 7: The educational background distribution of assessors.

B.2 Article Evaluation

Following Co-STORM (Jiang et al., 2024), we utilized the Prometheus-7b-v2.0 model for evaluation. Prometheus (Kim et al., 2024) is an open-source scoring model used to assess lengthy texts based on user-defined criteria. Its default temperature value is 1.0, and the top_p value is 0.9. Due to the model's limited context window, we exclude reference sections from the article evaluation and trim the input text to fewer than 2000 words to fit within the model's context window. This is consistent with STORM's approach (Shao et al., 2024), where the shortest section is removed each time until the article length meets the specified requirement. The scoring criteria for article quality evaluation can be found in Listing 10.

B.3 Variance of Article Evaluation

As shown in the table 4, we present the variance of three evaluation runs using the previously saved checkpoints on Prometheus-7B-v2.0.Thanks to the solid alignment of Prometheus-7B-v2.0, the variances are relatively small.

[4] https://github.com/shmsw25/FActScore

Method	Relevance	Breadth	Depth	Novelty
RAG	0.0027	0.0060	0.0092	0.0073
oRAG	0.0043	0.0071	0.0111	0.0132
STORM	0.0027	0.0052	0.0021	0.0085
Co-STORM	0.0032	0.0066	0.0036	0.0106
OmniThink	0.0011	0.0027	0.0042	0.0095

Table 4: Variance of three evaluation on Prometheus-7B-v2.0

C Human Evaluation

C.1 Human Evaluation Details

We randomly select 20 topics and compare articles generated by our method with those from the Co-STORM (the comprehensive best-performing baseline based on automatic evaluation), scoring them on the same four aspects. The participants in the evaluation voluntarily provided their highest educational qualification to demonstrate their ability to impartially assess the article. As shown in Figure 7, all of our human evaluators have an undergraduate degree or higher, with 53% having a graduate degree. As discussed in §C, to compare the merits of OmniThink and Co-STORM, each human evaluator was given a scoring criterion and a pair of articles. They were required to compare and assign scores, with the scoring criteria being the same as Lstlisting 10. We compiled the average scores given by the human evaluators for OmniThink and Co-STORM and compared their wins and losses.

C.2 Human Evaluation Results

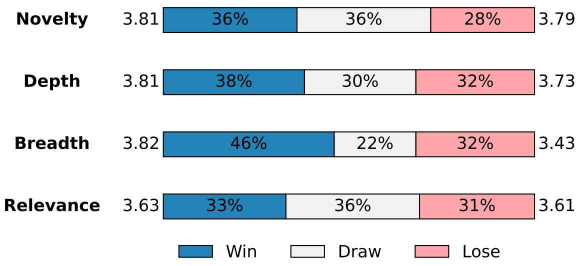


Figure 8: Comparison of OmniThink and Co-STORM results under human evaluation. The values on the left side represent the average score from OmniThink human evaluators, while the values on the right side represent the average score from Co-STORM human evaluators.

To better understand the strengths and weaknesses of OmniThink, we engage 15 well-educated volunteers to conduct a human evaluation. In Figure 8, we present the results of human scoring. The findings indicate that OmniThink's average performance surpasses that of the current strongest baseline across various dimensions, with a notable 11% improvement in the Breadth metric compared to Co-STORM. However, in terms of the Novelty metric, although automated evaluation shows an 11% enhancement, human assessment reveals only a marginal advantage. This discrepancy suggests that the current automated evaluation may not yet be fully aligned with human judgment, highlighting a direction for future improvement in the evaluation of long texts.

It should also be noted that despite OmniThink's overall superior performance in various dimensions, approximately 30% of the articles are considered equally excellent to the baseline by human evaluators. This could be attributed to the increasing difficulty for humans to discern subtle differences as the foundational writing capabilities of large models improve. Consequently, there is an urgent need to develop more rigorous and fine-grained evaluation methods to assess model performance more accurately.

D Further Analysis

D.1 Unique URL Analysis

To further investigate whether OmniThink surpasses these predefined boundaries, we conduct an unique url experiment. The goal is to examine whether OmniThink can retrieve more unique URLs compared to other methods, thus enabling the generation of more diverse and innovative content. Table 5 show that OmniThink retrieves significantly more unique URLs compared to other methods, such as Co-STORM, STORM, and oRAG. This indicates that OmniThink can access a broader range of diverse web content, which in turn enables the generation of more innovative and in-depth articles.

Method	OmniThink	Co-STORM	STORM	oRAG
Unique URLs	120.63	10.49	16.56	2.15

Table 5: Average number of unique URLs retrieved by each method.

D.2 Processing Time Analysis

We have recorded the time required for each method to run in the main table. Based on cost considerations, we use Google Search and Qwen-Plus. We ran 10 cases for each and calculated the

Method	OmniThink	Co-STORM	STORM
time(s)	322	289	289

Table 6: Average time taken by each method.

average time taken. As shown in Table 6, the current state of long text generation has encountered a certain bottleneck. We bypassed the scaling of complex text writing pipelines and instead focused on scaling from the data perspective to enhance text quality. We embraced the current trend of multiple rounds of reflection, led by DeepResearch. Therefore, we believe that these processing time costs are worthwhile.

E Information Boundary Experiments Details

In the information boundary analysis, our data comes from the results in Table 1, based on GPT-4o as the backbone. we extract the snippets content of each retrieved webpage from the search engine, then use Sentence-BERT to extract their representations. After reducing the dimensions to a 2D plane using PCA, we apply normalization and calculate the centroid for each category. Outliers, defined as points beyond 1.5 times the standard deviation, are exclude, and the convex hull formed by the remaining points is computed.

F Expansion & Reflection Experiments Details

Given the interdependent nature of expansion and reflection in OmniThink, it is impractical to assess their individual impacts in isolation. To address this challenge, we adopt an indirect yet systematic approach to evaluate their collective influence on the final articles' quality. During the information acquisition phase, we substitute the model used for expansion with a lower-performing model and measured the extent of performance decline in the generated article's metrics, which served as an indicator of the impact of the expansion process on these metrics. Specifically, based on the experimental results for qwen-plus-2024-08-06, we replace the models used for the expansion and reflection processes from Qwen-Plus to Qwen2.5-7b-instruct (Team, 2024) and observe the decline in various evaluation results. This transition allows us to observe and document the subsequent changes in a range of evaluation metrics, providing insights into the expansion and reflection process's influence on the articles' overall assessment.

G Comparison of features across different methods

Dynamic retrieval In previous methods, STORM and Co-STORM primarily retrieve web pages through ongoing dialogue, largely relying on the maximum number of conversations, without dynamically adjusting the retrieval of web content according to the difficulty and depth of the problem. OmniThink achieves dynamic retrieval based on the problem's difficulty by constantly reflecting on whether further retrieval is necessary with the current content.

Structured memory STORM stores web content merely through dialogue, while Co-STORM records a mind map during the conversation process. OmniThink not only records retrieved web pages in a progressive knowledge manner but also uses a conceptual pool to document changes in the LLM's understanding of the topic.

Reflective thinking In STORM and Co-STORM, continuous dialogue mainly occurs through role-playing, without reflection on the retrieved content. OmniThink achieves better results by continuously reflecting on the retrieved content to fill the conceptual pool.

H Effectiveness of Knowledge Density

We designed an interesting experiment to demonstrate the effectiveness of the KD evaluation metric.

First, we constructed 50 unique atomic facts across different topics and asked GPT-4o to generate a 500-word article based on these facts. Then, we gradually reduced the number of atomic facts while keeping the article length unchanged, in order to simulate articles with varying levels of knowledge density. To ensure stylistic consistency, all generations were produced using GPT-4o, so that the articles remained largely consistent in expression apart from differences in knowledge density. We invited three human volunteers and three language model evaluators (GPT-4o, DeepSeek-R1, and O3-mini-high) to assign preference scores to the articles generated with different amounts of atomic knowledge. The experimental results are shown in Table 7.

Evaluator	20 Facts	30 Facts	40 Facts	50 Facts
GPT-4o	1.0	2.0	3.0	3.6
DeepSeek-R1	1.0	2.6	2.8	3.6
O3-mini-high	1.0	2.4	3.2	3.4
Humans	1.0	2.1	3.1	3.8

Table 7: Preference scores assigned by human and LLM evaluators for articles generated with varying numbers of atomic facts.

I Case Study

In Figure 11, we present an example of AGI generated by OmniThink. It is generated using GPT-4o as the backbone. We can see that OmniThink's language is more concise compared to other methods, and it contains more information per unit of text length.

In addition, we present an example of AGI generated by the Reasoning model in Figure 12. We can observe that the OmniThink using the Reasoning model cites significantly more content per chapter, indicating that the model has improved its ability to utilize information through reflection.

J Decision Process of Expansion

Algorithm 1 Decision Process of Expansion

```
Input: Tree $\mathcal{T}$, Max Depth $D$, Conceptual Pool $\mathcal{P}$
Output: Updated $\mathcal{T}$ and $\mathcal{P}$
while depth($\mathcal{T}$) $< D$ do
    for each leaf node $N_i$ in $\mathcal{T}$ do
        $R_i \leftarrow$ LLM.decide_next($\mathcal{P}, N_i$)
        if $R_i$ requires expansion then
            Extract keywords and retrieve info
            Create sub-nodes and add to $\mathcal{T}$
        end if
    end for
    Update $\mathcal{P}$ with new insights
    if early stopping condition met then
        break
    end if
end while
Return $\mathcal{T}, \mathcal{P}$
```

In practice, we first check whether each leaf node of the information tree has reached a predefined maximum depth. If it has not, we feed the content and type of that node, along with the current conceptual pool, to the LLM as a prompt. The LLM is instructed to decide whether the node requires further expansion. If expansion is needed, the model generates potential sub-node categories and corresponding retrieval keywords based on the conceptual pool; otherwise, if the node is deemed sufficiently complete, the model produces no output.

To operationalize this, we extract the sub-node categories and keywords from the model's response using regular expressions. These elements are then employed to query web search engines or retrieval systems. The retrieved content forms the basis of new information nodes, which are added to the current information tree to iteratively refine and expand the knowledge structure.

Algorithm 1 is a brief pseudocode illustrating the overall expansion process. We first check whether the information tree has reached a predefined maximum depth. If not, the LLM is queried to decide the next steps for each leaf node. New information is retrieved accordingly and integrated into the tree. The conceptual pool is also dynamically updated during the expansion process.

K Clarification of Reflection

In this paper, our reflection refers to the process where the LLM reflects on the retrieved information based on its current Conceptual Pool, evaluating which parts of the information can enrich the existing Conceptual Pool. The usable information is then extracted as insights and added to the Conceptual Pool.

L Pseudo-code of Expansion & Reflection

Algorithm 2 Expansion and Reflection

```
Input: Topic T, Depth $K$
Output: Information Tree $\mathcal{T}$, Conceptual Pool $\mathcal{P}$
{Initialization}
Initialize Information Tree $\mathcal{T}_0$ with root node $N_r$
Retrieve initial information using search engines
Organize and analyze information to form Conceptual Pool $\mathcal{P}_0$
{Expansion and Reflection}
for each time step $m = 0$ to $K - 1$ do
    $L_m \leftarrow$ Leaf Nodes of $\mathcal{T}_m$
    Store $L_m$ in Conceptual Buffer $\mathcal{P}_b$
    for each node $N_i$ in $L_m$ do
        if Needs Expansion($N_i$) then
            Determine expansion areas using $\mathcal{P}_m$
            Generate sub-nodes SUB($N_i$) $= \{S_0, S_1, \ldots, S_{k_{N_i}}\}$
            for each sub-node $S_j$ in SUB($N_i$) do
                Retrieve information for $S_j$
                Add $S_j$ to $\mathcal{T}_{m+1}$
            end for
        end if
    end for
    $L_{m+1} \leftarrow$ Leaf Nodes of $\mathcal{T}_{m+1}$
    Analyze, filter, and synthesize information from $L_{m+1}$ to obtain insights $I_{m+1}$
    Update Conceptual Pool $\mathcal{P}_{m+1} \leftarrow$ Merge($I_{m+1}, \mathcal{P}_m$)
    if Sufficient information acquired then
        break
    end if
end for
Return Final Article $\mathcal{A}$
```

```python
class ExtendConcept(dspy.Signature):
    """
    You are an analytical robot. I will provide you with a subject, the information I have searched about it, and our
      preliminary concept of it. I need you to generate a detailed, in-depth, and insightful report based on it, further
      exploring our initial ideas.

    First, break down the subject into several broad categories, then create corresponding search engine keywords for each
      category.

    Note: The new categories should not repeat the previous ones.

    Your output format should be as follows:
    -[Category 1]
    --{Keyword 1}
    --{Keyword 2}
    -[Category 2]
    --{Keyword 1}


    --{Keyword 2}
    """
    info = dspy.InputField(prefix='The information you have collected from the webpage:', format=str)
    concept = dspy.InputField(prefix='The summary of the previous concepts:', format=str)
    category = dspy.InputField(prefix='The broader categories you need to further expand:', format=str)
    keywords = dspy.OutputField(format=str)
```

Listing 1: Prompts used for expanding in OmniThink.

```python
class GenConcept(dspy.Signature):
    """
    Please analyze, summarize, and evaluate the following webpage information.
    Think like a person, distill the core point of each piece of information, and synthesize them into a comprehensive opinion.

    Present your comprehensive opinion in the format of 1. 2. ...
    """
    info = dspy.InputField(prefix='The webpage information you have collected:', format=str)
    concepts = dspy.OutputField(format=str)
```

Listing 2: Prompts used for reflecting in OmniThink.

```python
class PolishPageOutline(dspy.Signature):
    """
    Improve an outline for a report page. You already have a draft outline that covers the general information. Now you want
      to improve it based on the concept learned from an information-seeking to make it more informative.
    Here is the format of your writing:
    1. Use "#" Title" to indicate section title, "##" Title" to indicate subsection title, "###" Title" to indicate
      subsubsection title, and so on.
    2. Do not include other information.
    3. Do not include topic name itself in the outline.
    """

    draft = dspy.InputField(prefix="Current outline:\n ", format=str)
    concepts = dspy.InputField(prefix="The information you learned from the conversation:\n", format=str)
    outline = dspy.OutputField(prefix='Write the page outline:\n', format=str)


class WritePageOutline(dspy.Signature):


    """
    Write an outline for a report page.
    Here is the format of your writing:
    1. Use "#" Title" to indicate section title, "##" Title" to indicate subsection title, "###" Title" to indicate
      subsubsection title, and so on.
    2. Do not include other information.
    3. Do not include topic name itself in the outline.
    """

    topic = dspy.InputField(prefix="The topic you want to write: ", format=str)
    outline = dspy.OutputField(prefix="Write the report page outline:\n", format=str)
```

Listing 3: Prompts used for writing the outline in OmniThink.

```
class WriteSection(dspy.Signature):
    """Write a Wikipedia section based on the collected information.

    Here is the format of your writing:
        1. Use "#" Title" to indicate section title, "##" Title" to indicate subsection title, "###" Title" to indicate
          subsubsection title, and so on.
        2. Use [1], [2], ..., [n] in line (for example, "The capital of the United States is Washington, D.C.[1][3]."). You DO
          NOT need to include a References or Sources section to list the sources at the end.
        3. The language style should resemble that of Wikipedia: concise yet informative, formal yet accessible.
    """

    info = dspy.InputField(prefix="The Collected information:\n", format=str)
    topic = dspy.InputField(prefix="The topic of the page: ", format=str)
    section = dspy.InputField(prefix="The section you need to write: ", format=str)
    output = dspy.OutputField(
        prefix="Write the section with proper inline citations (Start your writing with # section title. Don't include the page
            title or try to write other sections):\n",
        format=str)
```

Listing 4: Prompts used for writing section in OmniThink.

```
class PolishPage(dspy.Signature):
    """
    You are a faithful text editor that is good at finding repeated information in the article and deleting them to make sure
      there is no repetition in the article.
    You won't delete any non-repeated part in the article.
    You will keep the inline citations and article structure (indicated by "#", "##", etc.) appropriately.
    Refine the statement to avoid vague and ambiguous expressions, making it more concise and clear.
    Do your job for the following article.
    """

    article = dspy.InputField(prefix="The article you need to polish:\n", format=str)
    page = dspy.OutputField(
        prefix="Your revised article:\n",
        format=str)
```

Listing 5: Prompts used for polishing article in OmniThink.

Criteria Description	**Guidance for Content Generation**: Does the outline effectively guide content generation, ensuring comprehensive coverage of the topic?
Score 1 Description	The outline fails to guide content generation, omitting significant aspects of the topic or providing insufficient direction.
Score 2 Description	The outline provides limited guidance, covering some key areas but lacking depth or completeness in addressing the topic.
Score 3 Description	The outline provides moderate guidance for content generation, addressing most key areas but leaving some gaps or ambiguities.
Score 4 Description	The outline effectively guides content generation, covering all significant aspects with clear direction, though minor refinements could enhance comprehensiveness.
Score 5 Description	The outline is exemplary in guiding content generation, thoroughly addressing all aspects of the topic with clear, detailed direction and no significant gaps.
Criteria Description	**Hierarchical Clarity**: Does the outline clearly define a hierarchy of topics and subtopics, with a logical, diverse structure that is easy to understand?
Score 1 Description	The outline exhibits no discernible hierarchical structure. Topics and subtopics are jumbled together without logical separation or clear levels, making it nearly impossible to follow or identify any organization.
Score 2 Description	The outline attempts to establish a hierarchy but fails to maintain logical consistency. Main topics and subtopics are frequently misclassified, and the structure is overly rigid or disjointed. Subtopics may be missing, misplaced, or redundant, making it hard to grasp the intent of the structure.
Score 3 Description	The outline has a recognizable hierarchical structure but lacks diversity in organization style. While main topics are somewhat clear, subtopics occasionally overlap, are misaligned, or follow a repetitive format. This restricts flexibility and introduces mild confusion in certain areas.
Score 4 Description	The outline displays a clear, logical, and diverse hierarchical structure. Main topics are distinct, and subtopics are properly nested. While most elements are well-placed, there may be minor redundancies or opportunities to introduce more diverse formats for subtopics. Slight adjustments could achieve better precision and variety in style.
Score 5 Description	The outline showcases an exceptional, flawless hierarchical structure. Each main topic is distinct, and subtopics are logically nested with absolute clarity and stylistic diversity. The outline demonstrates flexibility in structure and organization, adapting its style where appropriate for the content and logic. No further refinement is necessary.
Criteria Description	**Logical Coherence**: Does the outline logically organize topics and subtopics, ensuring a smooth and natural flow of ideas with clear logical transitions?
Score 1 Description	The outline is highly disjointed and incoherent. Topics and subtopics appear in a random, unordered manner, with no logical flow or sense of progression. Major conceptual gaps and illogical jumps are present throughout the structure.
Score 2 Description	The outline shows some attempt at logical organization, but it contains frequent inconsistencies, abrupt shifts, or logical missteps. Topics and subtopics are misaligned or lack proper transitions, making the reader work hard to follow the structure.
Score 3 Description	The outline demonstrates a basic level of logical coherence. Most topics follow a general sequence, but some sections feel forced, with weak or unclear transitions. There are small jumps in logic, causing slight confusion or loss of flow at certain points.
Score 4 Description	The outline exhibits a strong sense of logical flow, with ideas presented in a mostly smooth and connected manner. Transitions between topics and subtopics are clear, but a few minor adjustments could make the flow more seamless or natural. The logic is sound, but room for refinement exists.
Score 5 Description	The outline achieves exceptional logical coherence. Each topic and subtopic follows a deliberate, thoughtful progression, with clear, natural, and intuitive transitions. The reader experiences a seamless flow of ideas, and no adjustments are required to improve logical consistency or flow.

Figure 9: Outline scoring rubrics on a 1-5 scale for the Prometheus model.

Criteria Description	**Broad Coverage**: Does the article provide an in-depth exploration of the topic and have good coverage?
Score 1 Description	Severely lacking; offers little to no coverage of the topic's primary aspects, resulting in a very narrow perspective.
Score 2 Description	Partial coverage; includes some of the topic's main aspects but misses others, resulting in an incomplete portrayal.
Score 3 Description	Acceptable breadth; covers most main aspects, though it may stray into minor unnecessary details or overlook some relevant points.
Score 4 Description	Good coverage; achieves broad coverage of the topic, hitting on all major points with minimal extraneous information.
Score 5 Description	Exemplary in breadth; delivers outstanding coverage, thoroughly detailing all crucial aspects of the topic without including irrelevant information.
Criteria Description	**Novelty**: Does the report cover novel aspects that relate to the user's initial intent but are not directly derived from it?
Score 1 Description	Lacks novelty; the report strictly follows the user's initial intent with no additional insights.
Score 2 Description	Minimal novelty; includes few new aspects but they are not significantly related to the initial intent.
Score 3 Description	Moderate novelty; introduces some new aspects that are somewhat related to the initial intent.
Score 4 Description	Good novelty; covers several new aspects that enhance the understanding of the initial intent.
Score 5 Description	Excellent novelty; introduces numerous new aspects that are highly relevant and significantly enrich the initial intent.
Criteria Description	**Relevance and Focus**: How effectively does the report maintain relevance and focus, given the dynamic nature of the discourse?
Score 1 Description	Very poor focus; discourse diverges significantly from the initial topic and intent with many irrelevant detours.
Score 2 Description	Poor focus; some relevant information, but many sections diverge from the initial topic.
Score 3 Description	Moderate focus; mostly stays on topic with occasional digressions that still provide useful information.
Score 4 Description	Good focus; maintains relevance and focus throughout the discourse with minor divergences that add value.
Score 5 Description	Excellent focus; consistently relevant and focused discourse, even when exploring divergent but highly pertinent aspects.
Criteria Description	**Depth of Exploration**: How thoroughly does the report explore the initial topic and its related areas, reflecting the dynamic discourse?
Score 1 Description	Very superficial; provides only a basic overview with significant gaps in exploration.
Score 2 Description	Superficial; offers some detail but leaves many important aspects unexplored.
Score 3 Description	Moderate depth; covers key aspects but may lack detailed exploration in some areas.
Score 4 Description	Good depth; explores most aspects in detail with minor gaps.
Score 5 Description	Excellent depth; thoroughly explores all relevant aspects with comprehensive detail, reflecting a deep and dynamic discourse.

Figure 10: Report scoring rubrics on a 1-5 scale for the Prometheus model.

AGI(GPT-4o)

# Introduction
Artificial General Intelligence (AGI) refers to artificial intelligence that can understand, learn, and apply intelligence across various tasks like a human[1]. Unlike narrow AI, designed for specific tasks, AGI can theoretically perform any intellectual task a human can. AGI has been a topic of scientific research and philosophical debate, focusing on machines exhibiting human−like cognition and autonomous learning[1]. The pursuit of AGI involves interdisciplinary research in computer science, cognitive science, neuroscience, and other fields to replicate human−like understanding and reasoning in machines[1]. Researchers aim to create systems that process information and adapt to new environments and tasks with minimal human intervention. Realizing AGI could revolutionize industries, enabling machines to perform complex problem−solving and decision−making tasks autonomously[1]. While AGI holds significant promise, it also poses substantial ethical, technical, and safety challenges related to its impact on employment, privacy, and existential risk, which remain active discussion topics[1]. Despite these challenges, advancements in machine learning, neural networks, and computational power continue to propel AGI development [1].

# Definitions and Characteristics
Artificial General Intelligence (AGI) is a form of artificial intelligence that understands, learns, and applies intelligence across various tasks, akin to human cognitive capabilities[1]. Unlike narrow AI, which performs specific tasks, AGI solves general problems without domain restrictions[1]. The concept became prominent in 2007 when AI researcher Ben Goertzel, inspired by DeepMind cofounder Shane Legg, popularized the term in a book[1]. AGI embodies qualities like adaptability and general problem−solving, setting it apart as a versatile entity compared to existing AI models, which operate within defined domains[1].

# Historical Context and Philosophical Underpinnings
The concept of AGI has been a subject of interest since the early days of computing and AI research. The idea of machines with human−like intelligence dates back to pioneers like Alan Turing, who in 1950 questioned "Can machines think?" and introduced the Turing Test to measure a machine's intelligent behavior[2]. AGI differs from narrow AI as it aims to simulate human cognitive abilities across varied tasks, positioning it at the intersection of technology, cognitive science, and ethics, thereby raising questions about intelligence, consciousness, and human cognition.

Advancements in AI, especially in generative models, have reignited discussions about AGI. Modern AI tools apply distinct embedding strategies to engage with data in text, images, and sound, reflecting early philosophical inquiries into human mind structures and potential mechanical replication.

Consequently, pursuing AGI is not just technological but also philosophical, prompting ongoing discussions about implications of creating machines that might match or surpass human intelligence.

# Key Issues in Development
Developing AGI presents critical challenges and ethical considerations. A primary issue is defining and replicating human cognitive processes in machines, as explored in Kurzweil's work on understanding human thought intricacies [3]. The potential for machines to exhibit human−like empathy and compassion is also under examination, shown in chatbot developments designed for self−compassion[4].

Another issue is the ethical implications of human−robot relationships, investigating boundaries of emotional interactions[5]. Moreover, safety and ethical considerations are crucial, especially in autonomous vehicles using advanced AI, where ethical frameworks must guide real−world decisions[6]. The socio−ethical impacts of AGI, like its influence on employment, privacy, and norms, require continuous evaluation. The Lighthill Report from the 1970s highlighted skepticism about AI's feasibility and potential, echoing today as developers and ethicists analyze AGI's implications[7]. Understanding these issues is essential for responsible AGI technology development and deployment.

# Current Technological Developments

Recent AGI advances focus on aligning AI systems with human values and ethics. A pivotal study offers insights into designing AI that integrates societal norms responsibly, ensuring AGI development adheres to safety and responsibility principles as it gains autonomy and decision−making capabilities[8].

# Future Directions
AGI development aligns with evolving technological paradigms, with potential in multisensory processing. Criticisms of visual dominance in models inspire this approach, paralleling architectural theory's multisensory considerations beyond visual[2][8]. This could lead to AGI processing data through multiple sensory inputs, enhancing contextual awareness.

Exploring sensory modality interactions, supporting a holistic sensory input perspective, AGI could achieve comprehensive environmental understanding, akin to enriching human experiences[13].

Critiques of visual dominance suggest AGI needs nuanced cognitive flexibility, adapting dynamically to varied contexts, researching psychosocial and philosophical intelligence aspects[14]. AGI's future involves multidisciplinary collaborations synthesizing sensory science, architecture, and social theory insights, making AGI a profound perception and intelligence exploration.

Figure 11: A case of AGI generated by OmniThink with GPT-4o.

AGI(DeepSeek-R1)

# Introduction

The development of ∗∗Artificial General Intelligence (AGI)∗∗systems capable of human−level reasoning and autonomy has raised significant ethical and legal questions, particularly in military contexts.

By 2050, autonomous weapon systems (AWS) powered by AGI could dominate battlefields, posing risks to human rights and humanitarian principles by challenging frameworks of ∗∗international humanitarian law (IHL)∗∗ and ∗∗ international human rights law (IHRL)∗∗.

Key issues include indiscriminate harm, accountability gaps, and loss of human control over lethal decisions[1].

Critics warn that AGI−driven AWS might exacerbate human dignity violations through invasive surveillance, uncontrolled conflict escalation, and the ethical dilemma of delegating life−and−death decisions to machines lacking empathy[1].

As nations like China prioritize human rights in tech governance, urgent regulatory clarity and safeguards are needed, requiring international cooperation to align AGI development with humanitarian goals[1].

# Historical Evolution

The EU's 2020 digital strategies ∗A European Strategy for Data∗, ∗Shaping Europe's Digital Future∗, and the ∗White Paper on Artificial Intelligence∗ redefined ∗∗technological sovereignty∗∗ by emphasizing reduced reliance on external technologies and Europe's role in shaping global AGI standards[2].

## Technological Autonomy

A core pillar of the EU's strategy focused on ∗∗technological autonomy∗∗, particularly in critical infrastructure like 5 G networks. The ∗Shaping Europe's Digital Future∗ document stressed indigenous capability development to avoid dependency, extending to AGI through investments in research and foundational technologies[2].

## Regulatory Frameworks

The EU further prioritized ∗∗regulatory sovereignty∗∗, building on the GDPR to propose laws like the ∗Data Act∗ and ∗Digital Services Act∗. These aimed to counter data monopolies, ensure ethical AI deployment, and create a unified data space with harmonized rules, positioning Europe as a global standard−setter in AGI governance[2].

# Technical Implementation Pathways

## Quantum Machine Learning and Algorithms

Quantum machine learning (QML) leverages variational quantum algorithms (VQAs) for tasks like quantum autoencoding, enabling efficient optimization in high−dimensional spaces critical for AGI[3][4]. Hybrid frameworks, such as quantum neural networks (QNNs) and quantum generative adversarial networks (QGANs), integrate classical and quantum components to model complex data[5].

## Error Correction and Noise Mitigation

Robust error correction, including quantum error correction (QEC) protocols and dynamical decoupling, is essential to maintain coherence in quantum systems. Noise disrupts optimization landscapes (Fig. 5), complicating training a challenge partially addressed by error−mitigated VQAs, though scalability remains unresolved[6][7][8].

## Hybrid Quantum−Classical Architectures

Hybrid systems combine classical machine learning with quantum subroutines, such as parameterized quantum circuits for decision−making under uncertainty. Quantum−enhanced reinforcement learning uses superposition to explore action spaces efficiently, though scalability depends on hardware advancements[9][10][11][12].

## Challenges and Limitations

Key hurdles include noise mitigation, resource optimization for hybrid systems, and AGI−compatible QEC protocols. Fault−tolerant quantum computation may be necessary for AGI−relevant advantages, requiring further exploration of quantum−classical algorithm interplay[7][8][13].

# Core Challenges in AGI Development

## Economic Value Measurement

Quantifying AGI's economic impact demands frameworks like the ∗∗Anthropic Economic Index∗∗, which assesses adaptability, decision−making, and task performance against human benchmarks[11][14]. Critics stress the need to account for labor disruption and ethical risks in autonomous economic decisions[11].

# Evaluation Methodologies

AGI evaluation requires human oversight to ensure compliance with humanitarian law. The International Committee of the Red Cross (ICRC) emphasizes predictability, reliability, and accountability in AGI decision−making, particularly for military systems[13][14]. Their frameworks advocate transparency, human judgment in critical functions, and multidisciplinary assessments to balance innovation with ethical safeguards[3][12][14].

# Contemporary Research Landscape

Current AGI research emphasizes hybrid cognitive models, exemplified by ∗∗OpenNARS for Research 3.0+∗∗, an open−source project integrating non−axiomatic logic and probabilistic inference for human−like reasoning under uncertainty. Hosted on GitCode, it enables modular experimentation with hybrid AI architectures, prioritizing transparency and scalability[15].

Figure 12: A case of AGI generated by OmniThink with DeepSeek-R1.