DiffResearch: Harnessing Diffusion Language Models for Automated Literature Review

Alan Gazzaev [1] **Mikhail Mozikov** [2] **Alexey Gavrilov** [1] **Sergey Muravyov** [1]

Abstract

Deep Research systems built on autoregressive large language models inherit the limitations of left-to-right decoding, including outline drift, error propagation across sections, and the limited ability to revise early structural decisions once new evidence is gathered downstream. As reviews grow longer and rely on dozens of retrieved sources, these constraints may lead to losses in comprehensiveness and insight, since later findings cannot easily reshape earlier commitments. We hypothesize that a different decoding paradigm may help address this gap. We introduce DiffResearch, to our knowledge the first Deep Research framework to place a diffusion language model at the core of its writing stage, aiming to support parallel refinement of an entire review rather than purely section-by-section generation. The system combines a lightweight multi-agent scaffold including intent classification, query reformulation, planning, retrieval, writing, and judging with two operating modes. These consist of a base mode for single-pass synthesis and an iterative subquery-decomposition mode where a judge agent identifies coverage gaps and triggers additional retrieval rounds until the evidence base is sufficient. On Deep Research Bench, DiffResearch achieves an overall score of 48.03, ranking above several publicly listed systems at the time of evaluation, including openai-deepresearch (46.45), Dr. Tulu (45.49), and claude-research (45.00), with strong scores across comprehensiveness (46.95), insight (48.20), instruction following (49.41), and readability (47.40).

[1]ITMO University [2]AXXX. Correspondence to: Alan Gazzaev <alanrbtx@gmail.com>.

Accepted at the AI for Science workshop (ICML 2026).

1. Introduction

The past two years have witnessed the rapid emergence of Deep Research systems as a flagship application of large language models (LLMs). Commercial products such as OpenAI Deep Research (OpenAI, 2025), Gemini Deep Research (Google, 2024), and Perplexity Deep Research (Perplexity AI, 2025), alongside a growing body of open source frameworks (LangChain, 2025; Zheng et al., 2025), autonomously orchestrate multi step web exploration, targeted retrieval, and higher order synthesis to transform vast amounts of online information into analyst grade, citation rich reports. Among the tasks these systems are asked to perform, scientific literature review is one of the most demanding and one of the most consequential. A literature review is not a summary of retrieved sources, it requires balanced thematic coverage, faithful attribution, the identification of competing methodological camps, and a coherent narrative that situates each work within the broader research landscape. As Deep Research Bench (Du et al., 2025) and similar evaluations (Coelho et al., 2025; Yifei et al., 2025) have demonstrated, the strongest agents now approach the quality of human research analysts on PhD level tasks across dozens of domains.

Despite this progress, current Deep Research systems share a common architectural backbone that may impose meaningful constraints on the literature reviews they produce. Almost all production DR agents are built on top of autoregressive LLMs (Brown et al., 2020; Touvron et al., 2023; OpenAI, 2023), and therefore inherit the same left to right, single pass decoding paradigm that governs their underlying token generators. This shapes both how reviews are written and how the surrounding agentic loop is structured. A writing agent drafts each section sequentially, with limited ability to revise earlier passages once new evidence arrives in later sections. Errors in early structural decisions, such as a poorly scoped subtopic, an omitted research thread, or a misjudgment about which methodological lineage deserves its own section, can propagate through the rest of the review and may be difficult to correct without regeneration. The cost is most visible on long, multi faceted surveys, where global coherence and balanced coverage are precisely what distinguish a useful literature review from a

list of paper summaries. The same sequential bottleneck can also constrain the orchestration layer: when an agent must emit multiple search queries to fan out retrieval over distinct facets of the literature, autoregressive decoding produces them one token at a time, and downstream retrieval calls are typically dispatched only after generation completes.

In this work, we introduce *DiffResearch*, a Deep Research framework purpose built for literature review, with a diffusion LLM as its core report generator. DiffResearch combines a standard agentic scaffold (Wang et al., 2024; Xi et al., 2023): an intent agent that selects between web and academic sources, a query reformulation agent, a planning agent that drafts the review outline, and a retrieval module that gathers papers and web pages with a diffusion based writing stage that drafts and iteratively refines the full review in parallel rather than section by section. An optional judge agent (Zheng et al., 2023) evaluates whether the final review is faithful to the plan and sufficiently complete. Building on this base, we introduce a second operating mode in which the original query is decomposed into multiple subqueries that target distinct facets of the literature. Because the decomposition agent is itself a diffusion LLM, the subqueries are generated within a single denoising process and become available together at the end of the generation pass. This allows the system to dispatch retrieval calls for every subquery in parallel as soon as generation completes, with each subquery retrieved and synthesized independently, and the judge agent triggers additional rounds of subquery generation and retrieval until coverage of the relevant subfields is deemed sufficient. We evaluate the system on Deep Research Bench (Du et al., 2025), a benchmark of 100 PhD level research tasks across 22 fields, which has become a standard yardstick for end to end research agents and whose tasks closely mirror the demands of real literature review work.

Our main contributions are as follows:

- **A diffusion native Deep Research framework for literature review.** We present DiffResearch, to our knowledge the first Deep Research system that uses a diffusion LLM as its primary report generator and aligns the agentic pipeline with literature review as its target task.
- **An iterative subquery decomposition mode.** We propose a second operating mode in which a diffusion based decomposition agent generates multiple subqueries during a single denoising pass, allowing the system to dispatch retrieval calls for every subquery in parallel as soon as generation completes. Each subquery can be retrieved and synthesized independently.
- **Empirical evaluation on Deep Research Bench.** We evaluate DiffResearch on the 100 PhD level tasks of Deep Research Bench and compare its report quality against publicly listed Deep Research systems.

2. Related Work

Deep Research systems have rapidly evolved as an important application of LLM based agents. Commercial systems such as OpenAI Deep Research (OpenAI, 2025), Gemini Deep Research (Google, 2024), and Perplexity Deep Research (Perplexity AI, 2025), along with open source frameworks (LangChain, 2025; Zheng et al., 2025), integrate multi step reasoning, retrieval, and synthesis to produce long form, citation grounded outputs. Benchmarks such as Deep Research Bench (Du et al., 2025), Deep Research Gym (Coelho et al., 2025), and ResearchQA (Yifei et al., 2025) evaluate these systems on complex, multi domain tasks and highlight their progress toward expert level performance.

Most existing Deep Research systems rely on autoregressive LLMs (Brown et al., 2020; Touvron et al., 2023; OpenAI, 2023). These models generate text sequentially and impose structural constraints on both writing and planning. Prior work has explored agentic scaffolds that decompose complex tasks into modular components such as planning, retrieval, and evaluation (Wang et al., 2024; Xi et al., 2023), as well as judge models that assess output quality (Zheng et al., 2023). Retrieval augmented generation RAG has also been widely studied as a mechanism to ground LLM outputs in external knowledge sources (Lewis et al., 2020; Gao et al., 2023b). However, these systems typically inherit the limitations of sequential decoding, particularly for long form generation tasks that require global coherence and iterative refinement.

In parallel, diffusion based language models have emerged as an alternative to autoregressive generation (Li et al., 2022; Austin et al., 2021; Lou et al., 2024). These models generate text through iterative denoising, refining an entire sequence in parallel across a small number of steps. Recent systems such as LLaDA (Nie et al., 2025), Dream (Ye et al., 2025), and Mercury 2 (Inception Labs et al., 2025) suggest that diffusion LLMs can approach the quality of autoregressive models and may offer potential advantages in generation speed and global coherence. Prior work has also shown that diffusion models naturally support structure first generation and iterative error correction (Li et al., 2022; Gong et al., 2023).

Despite these advances, diffusion LLMs have primarily been evaluated in standalone generation settings such as chat and coding (Nie et al., 2025; Ye et al., 2025). Their integration into agentic pipelines that involve planning, retrieval, and iterative refinement remains underexplored. In particular, to the best of our knowledge, no prior work has systematically studied how diffusion based generation interacts with litera-

ture review tasks, where global structure, balanced coverage, and faithful citation are essential. DiffResearch addresses this gap by introducing a diffusion native Deep Research framework that integrates retrieval, planning, and iterative refinement within a unified system.

3. Methods

In this section we describe the architecture of DiffResearch, the retrieval backends that ground its outputs, the two operating modes the system supports, and the motivation for building a literature-review system around a diffusion LLM.

3.1. System Overview

DiffResearch is a multi-agent framework in which every agent is an instance of a diffusion LLM. The system is organized as a small, explicit pipeline rather than as a monolithic chain. We deliberately avoid heavy orchestration abstractions of the kind found in LangChain (Chase, 2022) or similar frameworks; in our experience these abstractions obscure the data flow between agents and make it difficult to reason about latency budgets, retry behavior, and the exact prompt seen by the model at each step. Instead, DiffResearch exposes a thin interface in which each agent is a Python class that inherits from a common backbone: OpenAIAgent for autoregressive baselines or LLaDAAgent for diffusion-native deployment, and implements a single *generate* method.

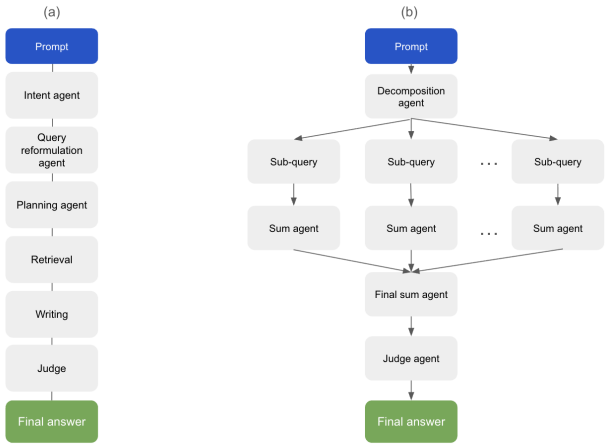

Figure 1. (a): The first mode of operation of the system for writing a literature review. Suitable for both open and closed dLLMs. (b): Parallel Diffusion Mode: An optimized mode for open-source dLLMs that leverages parallel token denoising to execute multiple concurrent queries and is designed to improve throughput for long-form literature review generation through high-throughput inference.

Each subclass encapsulates one well-defined responsibility through its own task-specific prompt: nine agents in total, covering query formatting, planning, plan-checking, relevance filtering, information extraction, summarization, complexity assessment, query decomposition, and judging. Outputs are kept minimal and machine-parseable: binary flags (0/1) for routing decisions made by the relevance and complexity agents, newline-separated query lists from the decomposition, judge, and plan-check agents, and free-form text only where downstream consumption requires it (extraction, summarization, planning).

Per-agent decoding parameters are exposed through the constructor: for example, the RelevanceAgent and ComplexityAgent operate with a short generation budget (gen length=32, steps=32) appropriate to their binary-flag outputs, while the SummarizationAgent is configured with a substantially larger budget (gen length=512, steps=128) to accommodate full-length review drafts. Adding a domain-specific reranker or an agent that extracts methodological taxonomies from retrieved papers requires only subclassing the appropriate backbone, defining a generate method with the relevant prompt, and wiring the new class into the pipeline.

The base pipeline consists of five core stages: intent classification, query reformulation, planning, retrieval, and writing, with judging used as an optional sixth stage.

(1) Intent classification. The user's natural-language query is passed to an intent agent that decides between two retrieval channels: a web channel for queries whose answers depend on grey literature, blog posts, technical reports, or rapidly changing engineering practice; and an academic channel for queries whose answers should be grounded primarily in peer-reviewed publications. The intent agent emits a structured label together with a short rationale; the rationale is logged but not propagated downstream, so that the routing decision does not bias subsequent agents.

(2) Query reformulation. A reformulation agent rewrites the original query into one or more retrieval-friendly search strings (Wang et al., 2023; Jagerman et al., 2023), tuned to the syntax and conventions of the channel selected at the previous stage. The agent is instructed to preserve the user's original information need verbatim in a separate field so that downstream agents always have access to the unrewritten query.

(3) Planning. A planning agent (Wei et al., 2022; Yao et al., 2023) consumes the original query and produces a structured outline for the literature review. The outline specifies a list of main sections (e.g., Introduction, Background, Thematic Areas, Methodology Comparison, Key Findings, Research Gaps, Conclusion), two to four sub-questions or

points to be addressed within each section, and an explicit list of key concepts, methods, or datasets that must be covered. The plan is the central artifact of the system: it is passed alongside the retrieved corpus to the writing agent as an explicit drafting target, and it serves as the reference against which the plan-check agent evaluates completeness once the review is produced.

(4) Retrieval. Given the plan and the reformulated queries, the retrieval module collects papers and web pages from the selected channel. Retrieval is performed by deterministic API calls to dedicated backends, described in the next subsection, followed by content extraction. The retrieved corpus is then passed to the writing stage along with the plan.

(5) Writing. A writing agent drafts the literature review conditioned on the plan and the retrieved corpus. Because the writing agent is itself a diffusion LLM call, the review is produced through diffusion-style refinement rather than token-by-token autoregression.

(6) Judging. A sixth, optional stage is the judge agent, which compares the produced review against the plan and against the retrieved corpus. The judge emits a structured verdict along three axes: *plan adherence* (does the review cover every subtopic listed in the plan?), *citation faithfulness* (Gao et al., 2023a; Liu et al., 2023) (is each claim supported by a retrieved source?), and *coverage completeness* (are there facets of the topic that the retrieved corpus addresses but the plan and review missed?). In the base mode the judge's output is reported to the user as a quality signal but does not trigger further action.

3.2. Retrieval Backends

The retrieval module is the part of DiffResearch that grounds every claim in the final review, and we therefore treat it as a first-class component rather than an implementation detail. Retrieval is dispatched to one of two channels based on the intent agent's decision.

Web channel: Serper API. For queries routed to the web channel, DiffResearch issues queries through the Serper API (Serper, 2024), a programmatic interface to Google Search. Serper returns ranked organic results together with structured metadata (title, URL, snippet, publication date when available). Each result URL is then fetched and passed through a content-extraction step that strips boilerplate and normalizes the page into plain text suitable for downstream agents. We use Serper rather than direct scraping because it provides stable result formatting, handles rate limiting and CAPTCHA cleanly, and exposes the same ranking signals across runs, which matters for reproducibility of evaluation results.

Academic channel: Semantic Scholar and arXiv. For queries routed to the academic channel, DiffResearch issues parallel calls to two complementary backends. The Semantic Scholar API (Kinney et al., 2023) provides broad coverage of the peer-reviewed literature across disciplines, with structured metadata (authors, venue, year, citation counts, abstracts) and citation-graph information that we use for follow-on retrieval of highly cited or recently citing work. The arXiv API (arXiv, 2024) provides timely access to preprints, which is essential for fast-moving fields where the most relevant work has not yet appeared in Semantic Scholar's index. We deduplicate across the two sources by DOI and arXiv identifier, and merge the results into a single ranked list before passing the corpus to the writing stage. Full-text PDFs are fetched from arXiv where available; for Semantic Scholar entries without an open-access full text, we fall back to the abstract.

Splitting the academic channel across these two backends is a deliberate design choice. Semantic Scholar alone underweights very recent work; arXiv alone misses the long tail of older or non-CS literature that Semantic Scholar indexes well. Running both in parallel and merging is cheap relative to the cost of an LLM call and meaningfully broadens coverage on PhD-level survey topics that span subfields with different publication norms.

In all cases, retrieval itself is performed by deterministic API calls: there is no LLM in the retrieval loop. This keeps the retrieval step fast, cacheable, and easy to ablate; an LLM is only reintroduced when the retrieved corpus is handed to the planning, writing, or judge agents.

3.3. Why Diffusion May Matter Here

Our motivation for using a diffusion LLM is that literature reviews require decisions about global structure, thematic balance, and cross-paper synthesis. We hypothesize that parallel refinement may be better aligned with these requirements than strictly left-to-right generation. Autoregressive writing agents commit to early tokens before later evidence is integrated, which in long-form survey writing can manifest as *outline drift*: the introduction frames the review around one taxonomy, but by the time the writer reaches later sections, the corpus may pull the narrative toward a different organization, and the early framing may no longer be consistent with the body. A diffusion writer instead refines the entire review in parallel (Li et al., 2022; Nie et al., 2025), potentially allowing global structure to stabilize before local wording is finalized. Section boundaries, the choice of which methodological lineage to highlight, and the balance between competing schools of thought are global decisions that should ideally be made jointly across the whole re-

view, rather than committed to one section at a time, which parallel denoising is designed to support.

A second potential consequence of the diffusion paradigm is felt in the orchestration layer rather than the writing stage. When the decomposition agent generates a set of subqueries, they are produced within a single denoising pass and become available together, allowing the pipeline to dispatch retrieval calls in parallel as soon as generation completes. An autoregressive decomposition agent, by contrast, emits subqueries one token at a time, and downstream retrieval is typically dispatched after generation completes. This makes the iterative subquery mode introduced in the next section practical at a granularity that may be difficult to justify with an autoregressive backbone of comparable size, although the magnitude of this advantage depends on the relative cost of generation and retrieval in a given deployment setting.

3.4. Operating Modes

DiffResearch supports two operating modes that share the same agents but differ in how the pipeline is composed.

Base mode. The pipeline runs the five base stages (intent, reformulation, planning, retrieval, and writing) once, with the judge agent attached as an optional post-hoc evaluator. This mode is appropriate for queries that are narrow enough to be answered from a single retrieval pass and where the user values latency over exhaustive coverage.

Iterative subquery mode. For queries whose target literature spans multiple subfields or methodological camps, the base mode may under-cover certain facets, since a single reformulation cannot capture every relevant search direction. In iterative mode we insert a decomposition agent (Khot et al., 2023; Press et al., 2023) between the intent and reformulation stages. The decomposition agent splits the original query into a set of subqueries, each targeting one facet of the topic.

4. Results and Discussion

We evaluated DiffResearch on Deep Research Bench (Du et al., 2025), a benchmark consisting of 100 PhD-level research tasks (50 of them in Chinese and 50 in English), spanning 22 distinct fields, designed to assess end-to-end Deep Research Agents on report generation and information retrieval. The task distribution is calibrated against a statistical analysis of over 96,000 real-world user queries, so performance on the benchmark is intended to track model performance on the kind of literature-review queries that real users issue. We followed the benchmark's official RACE evaluation framework, which scores generated reports along four dimensions: comprehensiveness (Comp.), insight (Insight), instruction-following (Inst.), and readability (Read.), and reports an aggregate overall score.

All agents in the pipeline are instances of Mercury 2. We use the same model for every stage (intent classification, query reformulation, planning, retrieval-side filtering, writing, and judging). This removes one source of model heterogeneity across stages, but it does not by itself isolate the effect of diffusion decoding from the surrounding agentic scaffold.

Table 1 reports DiffResearch's performance on Deep Research Bench alongside the publicly listed entries on the benchmark leaderboard at the time of evaluation, including proprietary frontier systems and open-source baselines.

DiffResearch achieves competitive performance on the benchmark, scoring 48.03 overall and ranking above several systems on the public leaderboard snapshot used in this study. Among proprietary commercial Deep Research products, DiffResearch scores above Claude Research by 3.03 points, Kimi Researcher by 3.39 points, and Doubao Deep Research by 3.69 points, while remaining within close range of OpenAI Deep Research (46.45) and UESTC-MBSE-RAAA-DeepResearch (46.13). Among open-source and openly licensed systems, DiffResearch scores above LangChain Open Deep Research by 4.59 points, NVIDIA AIQ Research Assistant by 7.51 points, and Tongyi Deep Research 30B-A3B by 7.57 points. We view the comparison against proprietary systems as informative, but not as a controlled isolation of any single architectural factor: DiffResearch is released under an Apache-2.0 license and runs on commodity API access, yet scores above several closed Deep Research entries on this benchmark snapshot.

The per-dimension breakdown clarifies where DiffResearch performs strongest. DiffResearch's strongest absolute dimension is instruction-following (49.41), narrowly behind Dr. Tulu (49.56) and essentially tied with OpenAI Deep Research (49.39), while ahead of most other commercial products. This is consistent with our design: the explicit plan produced by the planning agent and verified by the judge agent gives the writing stage an unambiguous target, which may contribute to instruction-following performance. The strongest relative gains are on the two dimensions most diagnostic of literature-review quality. On insight, DiffResearch reaches 48.20, ahead of OpenAI Deep Research (43.73) by 4.47 points and ahead of several other proprietary systems by 5–8 points, with UESTC-MBSE-RAAA-DeepResearch as the closest competitor (48.34). On readability, DiffResearch reaches 47.40, ahead of OpenAI Deep Research (47.22) and ahead of most other systems on the leaderboard, with Kimi Researcher in third at 45.59.

The combination of strong insight and strong readability is consistent with our hypothesis that a diffusion writer may benefit long-form synthesis. Insight scores reward syntheses that draw non-trivial connections across the retrieved corpus,

Table 1. Performance of DiffResearch on Deep Research Bench compared to publicly listed leaderboard entries at the time of evaluation. Scores are reported on the four RACE dimensions (Comp. = comprehensiveness, Insight, Inst. = instruction-following, Read. = readability) and as an aggregate Overall score. License is reported as listed on the public leaderboard.

MODEL	OVERALL	COMP.	INSIGHT	INST.	READ.	LICENSE
DIFFRESEARCH (OURS)	**48.03**	**46.95**	**48.20**	**49.41**	**47.40**	APACHE-2.0
OPENAI-DEEPRESEARCH (OPENAI, 2025)	46.45	46.46	43.73	49.39	47.22	PROPRIETARY
UESTC-MBSE-RAAA-DEEPRESEARCH	46.13	43.77	48.34	47.21	43.78	PROPRIETARY
DR. TULU (SHAO ET AL., 2025)	45.49	44.08	44.65	49.56	42.30	APACHE-2.0
CLAUDE-RESEARCH (ANTHROPIC, 2025)	45.00	45.34	42.79	47.58	44.66	PROPRIETARY
KIMI-RESEARCHER (MOONSHOT AI, 2025)	44.64	44.96	41.97	47.14	45.59	PROPRIETARY
DOUBAO-DEEPRESEARCH (BYTEDANCE, 2025)	44.34	44.84	40.56	47.95	44.69	PROPRIETARY
LANGCHAIN-OPEN-DEEP-RESEARCH (LANGCHAIN, 2025)	43.44	42.97	39.17	48.09	45.22	UNKNOWN
NVIDIA-AIQ-RESEARCH-ASSISTANT (NVIDIA, 2025)	40.52	37.98	38.39	44.59	42.63	UNKNOWN
TONGYI-DEEPRESEARCH-30B-A3B (TDR TEAM, 2025)	40.46	39.46	34.44	46.22	44.27	UNKNOWN

and readability scores reward globally coherent prose; both plausibly benefit from a writer that can revise global structure rather than committing section by section. However, because the comparison is against full systems rather than writer-isolated variants, we do not interpret these results as a causal attribution to diffusion decoding alone. A controlled comparison between diffusion and autoregressive writers under identical plans, retrieved corpora, prompts, and evaluation settings is needed to determine how much of the observed improvement is due specifically to the generator architecture.

4.1. Qualitative Writer-Isolated Check

As a small-scale qualitative writer-isolated check, we compared two reports generated for the same literature-review topic using two models: the autoregressive `google/gemma-4-26B-A4B-it` and the diffusion-based `google/diffusiongemma-26B-A4B-it`. Both models were given the same query, "diffusion LLM," and asked to produce a literature-review report. Both outputs followed the requested high-level structure and produced readable surveys. However, the diffusion-written report exhibited a more explicit review-level organization, including a clearer objective, thesis statement, summary of findings, and final verdict. It also framed the comparison between diffusion and autoregressive models more cautiously, explicitly noting that diffusion LLMs remain less competitive on raw perplexity and long-form semantic coherence while still being promising for controllable generation and iterative editing.

The autoregressive report was fluent and topic-focused, but it made stronger claims about diffusion models as successors to autoregressive LLMs and showed weaker citation bookkeeping, including references cited in the text that were absent from the bibliography. The diffusion-written report was not uniformly superior: it occasionally broadened the discussion toward adjacent diffusion applications outside language modeling. Nevertheless, this small paired comparison is consistent with our hypothesis that diffusion-style writing may help produce more globally organized and balanced literature-review outputs. We treat this observation as illustrative rather than conclusive, and leave a larger writer-isolated ablation across more tasks and model backbones to future work.

Two aspects of these results deserve emphasis. First, the gain over several proprietary systems is consistent across nearly every dimension in the public leaderboard snapshot used here. Second, the systems that DiffResearch ranks above are not weak baselines: they include production Deep Research offerings of major frontier labs, each of which may have access to internal retrieval infrastructure, larger writer models, and substantial engineering investment. That a relatively small open-source system built on a public diffusion LLM API and three off-the-shelf retrieval backends can score above a substantial fraction of the entries on this leaderboard suggests that generator architecture and agentic loop design may be important factors for the literature-review task, alongside raw model scale.

5. Conclusion

We presented DiffResearch, to our knowledge the first Deep Research framework to place a diffusion language model at the core of its writing stage and to exploit parallel denoising in its orchestration layer. On Deep Research Bench, the system scores above a substantial fraction of publicly listed entries, with its strongest results concentrated on insight and readability, the two dimensions most directly tied to long-form synthesis. We view this as preliminary evidence that generator architecture may be an important factor in long-form retrieval-grounded writing, alongside model scale, retrieval quality, and agentic design. At the same time, a controlled writer-isolated comparison is needed to determine how much of the observed improvement is due specifically to diffusion decoding. We hope that releasing DiffResearch under an Apache-2.0 license lowers the barrier to further work at the intersection of diffusion LLMs and

agentic retrieval pipelines.

6. Limitations

A central limitation of our evaluation is that the main leaderboard comparison does not isolate the contribution of diffusion decoding from the surrounding agentic scaffold. The observed gains may arise from the diffusion writer, from the planning and retrieval pipeline, or from their interaction. We therefore treat the diffusion-specific explanation as a hypothesis rather than as a fully established causal claim. A more complete study should compare diffusion and autoregressive writers under identical plans, retrieved corpora, prompts, and evaluation settings.

We treat the diffusion generator as a black box: we do not exploit hidden states, intermediate denoising trajectories, or partial-noise representations, all of which could in principle support stronger forms of in-generation citation verification or plan-conditioned guidance. Our judge and plan-check agents operate only on the final review and are not embedded inside the denoising loop; tighter integration, in which a verifier conditions individual denoising steps, is left to future work.

The latency argument we make for the iterative subquery mode is most defensible when generation cost is modest relative to retrieval. On hardware or model configurations where local diffusion inference becomes the bottleneck, the marginal-cost claim weakens and the case for the iterative mode rests more heavily on its quality contribution alone. We have not benchmarked DiffResearch on consumer-grade hardware or under tight latency budgets, and we expect the practical viability of the iterative mode to vary across deployment settings.

Our evaluation is restricted to Deep Research Bench. While the benchmark is calibrated against a large corpus of real-world user queries and spans 22 fields, the RACE rubric is one specific operationalization of literature-review quality, and the leaderboard reflects a snapshot in time rather than an exhaustive comparison. Additional evaluations, particularly those involving expert human raters in specific domains, or benchmarks that test long-horizon multi-document synthesis directly would strengthen the claims made here.

Acknowledgements

This work supported by the Ministry of Economic Development of the Russian Federation (IGK 000000C313925P4C0002), agreement No139-15-2025-010

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Anthropic. Claude takes research to new places. https://www.anthropic.com/news/research, 2025. Accessed: 2026-06-17.

arXiv. arXiv API access. https://info.arxiv.org/help/api/index.html, 2024. Accessed: 2026-06-17.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

ByteDance. Doubao Deep Research. https://www.doubao.com/, 2025. Accessed: 2026-06-17.

Coelho, J., Ning, J., He, J., Mao, K., Paladugu, A., Setlur, P., Jin, J., Callan, J., Magalhães, J., Martins, B., and Xiong, C. DeepResearchGym: A free, transparent, and reproducible evaluation sandbox for deep research. *arXiv preprint arXiv:2505.19253*, 2025.

Du, M., Xu, B., Zhu, C., Wang, X., and Mao, Z. DeepResearch Bench: A comprehensive benchmark for deep research agents. *arXiv preprint arXiv:2506.11763*, 2025.

Gao, T., Yen, H., Yu, J., and Chen, D. Enabling large language models to generate text with citations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023a.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023b.

Gong, S., Li, M., Feng, J., Wu, Z., and Kong, L. DiffuSeq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations (ICLR)*, 2023.

Google. Try Deep Research and our new experimental model in Gemini, your AI

assistant. https://blog.google/products-and-platforms/products/gemini/google-gemini-deep-research/, 2024. Accessed: 2026-06-17.

Inception Labs, Khanna, S., Kharbanda, S., Li, S., Varma, H., Wang, E., Birnbaum, S., Luo, Z., Miraoui, Y., Palrecha, A., Ermon, S., Grover, A., and Kuleshov, V. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

Jagerman, R., Zhuang, H., Qin, Z., Wang, X., and Bendersky, M. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*, 2023.

Khot, T., Trivedi, H., Finlayson, M., Fu, Y., Richardson, K., Clark, P., and Sabharwal, A. Decomposed prompting: A modular approach for solving complex tasks. In *International Conference on Learning Representations (ICLR)*, 2023.

Kinney, R., Anastasiades, C., Authur, R., Beltagy, I., Bragg, J., Buraczynski, A., Cachola, I., Candra, S., Chandrasekhar, Y., Cohan, A., et al. The Semantic Scholar open data platform. *arXiv preprint arXiv:2301.10140*, 2023.

LangChain. Open Deep Research. https://github.com/langchain-ai/open_deep_research, 2025. Accessed: 2026-06-17.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Li, X. L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Liu, N. F., Zhang, T., and Liang, P. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

Lou, A., Meng, C., and Ermon, S. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning (ICML)*, 2024.

Moonshot AI. Kimi-Researcher: End-to-end RL training for emerging agentic capabilities. https://moonshotai.github.io/Kimi-Researcher/, 2025. Accessed: 2026-06-17.

Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., and Li, C. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

NVIDIA. AI-Q NVIDIA blueprint. https://github.com/NVIDIA-AI-Blueprints/aiq, 2025. Accessed: 2026-06-17.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

OpenAI. Introducing deep research. https://openai.com/index/introducing-deep-research/, 2025. Accessed: 2026-06-17.

Perplexity AI. Introducing Perplexity Deep Research. https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research, 2025. Accessed: 2026-06-17.

Press, O., Zhang, M., Min, S., Schmidt, L., Smith, N. A., and Lewis, M. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

Serper. Serper API: Google search API. https://serper.dev/, 2024. Accessed: 2026-06-17.

Shao, R., Asai, A., Shen, S. Z., Ivison, H., Kishore, V., Zhuo, J., Zhao, X., Park, M., Finlayson, S. G., Sontag, D., Murray, T., Min, S., Dasigi, P., Soldaini, L., Brahman, F., Yih, W.-t., Wu, T., Zettlemoyer, L., Kim, Y., Hajishirzi, H., and Koh, P. W. DR Tulu: Reinforcement learning with evolving rubrics for deep research. *arXiv preprint arXiv:2511.19399*, 2025.

TDR Team. Tongyi DeepResearch technical report. *arXiv preprint arXiv:2510.24701*, 2025.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Wang, L., Yang, N., and Wei, F. Query2doc: Query expansion with large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., and Wen, J.-R. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 2024.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., and Kong, L. Dream 7B: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Yifei, L. S., Chang, A., Malaviya, C., and Yatskar, M. ResearchQA: Evaluating scholarly question answering at scale across 75 fields with survey-mined questions and rubrics. *arXiv preprint arXiv:2509.00496*, 2025.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., et al. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zheng, Y., Fu, D., Hu, X., Cai, X., Ye, L., Lu, P., and Liu, P. DeepResearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025.