

# O-Researcher: An Open Ended Deep Research Model via Multi-Agent Distillation and Agentic RL

OPPO AI Agent Team

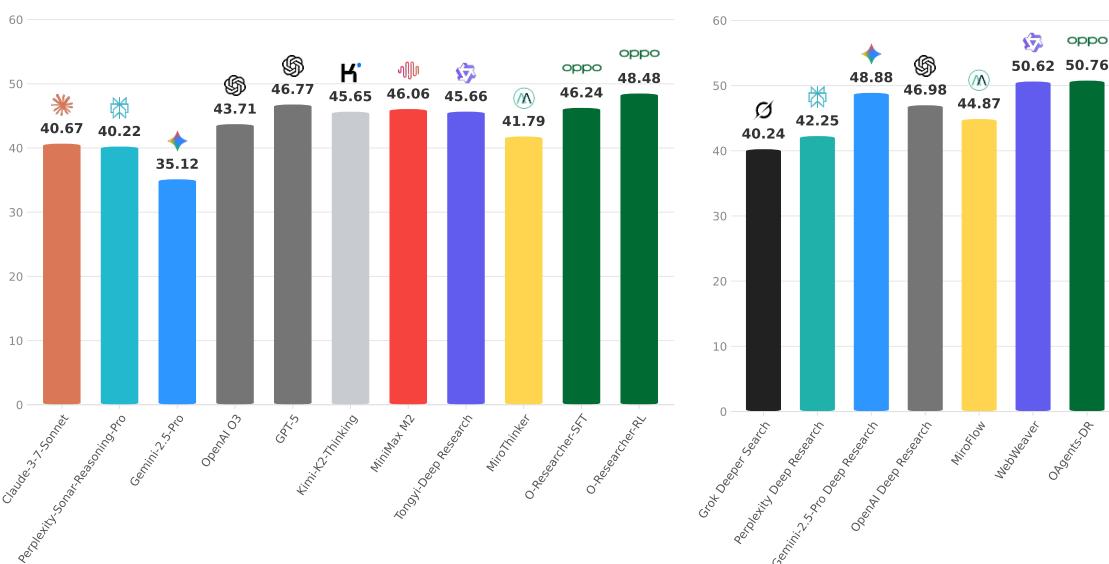
## Abstract

The performance gap between closed-source and open-source large language models (LLMs) is largely attributed to disparities in access to high-quality training data. To bridge this gap, we introduce a novel framework for the automated synthesis of sophisticated, research-grade instructional data. Our approach centers on a multi-agent workflow where collaborative AI agents simulate complex tool-integrated reasoning to generate diverse and high-fidelity data end-to-end. Leveraging this synthesized data, we develop a two-stage training strategy that integrates supervised fine-tuning with a novel reinforcement learning method, designed to maximize model alignment and capability. Extensive experiments demonstrate that our framework empowers open-source models across multiple scales, enabling them to achieve new state-of-the-art performance on the major deep research benchmark. This work provides a scalable and effective pathway for advancing open-source LLMs without relying on proprietary data or models.

**Date:** January 8, 2026

**Open Source:** O-Researcher-SFT O-Researcher-RL

**Correspondence:** Wangchunshu Zhou at [zhouwangchunshu@oppo.com](mailto:zhouwangchunshu@oppo.com)



(a) Deep Research Models

(b) Deep Research Agents

## 1 Introduction

The rapid evolution of Large Language Models (LLMs) has become a cornerstone of modern artificial intelligence, driving breakthroughs in tasks ranging from natural language understanding to complex reasoning [1; 2]. However, despite these advancements, a persistent challenge within this field is the performance disparity between powerful, closed-source models (e.g., GPT-4o, OpenAI o1 [2]) and their open-source counterparts. This gap is often attributed to the vast amounts of proprietary, high-quality training data and immense computational resources available to their developers [3]. This disparity becomes most apparent in reasoning-intensive tasks that demand extended chains of thought and rigorous problem-solving capabilities [4]. Consequently, the research community faces a significant bottleneck: how to effectively empower open-source models to achieve state-of-the-art (SOTA) performance without relying on such exclusive advantages.

A critical pathway to overcoming this bottleneck is the generation of high-quality, diverse, and scalable instructional data for supervised fine-tuning (SFT) and reinforcement learning (RL) [5; 6]. While existing methods often rely on human annotation or distillation from larger teacher models, these approaches are either prohibitively expensive, limited in scale, or risk inheriting the limitations of the teacher [7]. Crucially, standard distillation often captures only the final answer, failing to transfer the intricate “thought process” required for complex problem-solving [4]. Thus, there is a pressing need for an automated, end-to-end framework that can synthesize vast and sophisticated research-grade data to fuel the next generation of open-source LLMs.

In this paper, we propose a novel framework that addresses this fundamental challenge through a multi-agent driven synthetic data generation workflow. Our core insight is that a collaborative ecosystem of specialized AI agents can autonomously simulate complex human reasoning processes—akin to a rigorous peer-review mechanism—to create high-fidelity, multi-turn instruction-response pairs [8; 9]. This synthetic data serves as the foundation for a robust two-stage training strategy, which integrates a novel reinforcement learning method designed to align model outputs with precise quality and correctness metrics [4; 10].

Our contributions are summarized as follows:

- We raise a novel multi-agent driven workflow to automatically synthesize end-to-end deep research data generation. This system leverages a structured collaboration between multiple LLM agents to decompose, debate, and verify complex tasks, resulting in a scalable pipeline for producing premium training corpora that surpasses the quality of standard model-generated text [5; 8].
- Based on the synthesized data from our proposed workflow, we design a two-stage training strategy that incorporates a novel reinforcement learning method. This strategy first employs supervised fine-tuning on the synthetic data to establish a strong knowledge base, followed by a reinforcement learning phase that further refines the model’s performance and self-correction capabilities [4].
- We demonstrate that our method empowers open-source models in multiple sizes to achieve new SOTA performance. Through extensive experiments on major deep research benchmarks, we show that models trained with our framework not only significantly close the gap to closed-source models but also establish new state-of-the-art results for models of comparable size in the open-source domain.

## 2 Related Work

Large Reasoning Models (LRMs) [11; 12; 13] have achieved impressive results even in challenging domains like math and code. Web agents have also become a significant research direction, with rapid development driven by a series of systems [14; 15; 16; 17; 18; 19] aiming for web-scale reasoning and information-seeking. Nevertheless, their capabilities remain fundamentally constrained by their internal closed-world knowledge boundaries [20]. To mitigate these limitations, agentic systems [21; 22; 23; 24] have been introduced to augment LRMs by enabling interaction with external APIs, search engines, and computational tools [25]. In December 2024, Google introduced its initial research-oriented implementation, Gemini Deep Research [26], focusing on the agentic system with multi-step reasoning and knowledge integration. Building on these advancements, the task of deep research report generation [27; 28; 29; 30] emerged as a specialized benchmark designed to systematically exploit agentic capabilities. Recent works have further extended this to personalized research evaluations [30] and open sandboxes like DeepResearchGym [31], testing abilities to execute long-horizon planning and synthesize heterogeneous evidence into faithful analyses [20; 32; 33].

The field is advancing rapidly, driven by proprietary DRAs such as OpenAI Deep Research [34], Tongyi Deep Research [35; 36], Grok DeeperSearch [37], and AutoGLM-Research [38]. Parallel open-source efforts include OAgents [39], Jina-AI node-DeepResearch [40], Agentic Reasoning [25], DeepResearcher [27], OpenResearcher [28], and WebLeaper [19]. Collectively, these systems center around planning, query formulation, knowledge discovery, and report generation [20; 32].

Mainstream DRAs [25; 29] often employ monolithic architectures or specific agentic strategies. For instance, Agentic Reasoning [25] incorporates a Mind-Map agent for robust planning, while Flash-Searcher [41] introduces DAG-based parallel execution to enhance information seeking efficiency. In terms of knowledge synthesis, methods range from structure-controlled generation [42] to multimodal reporting [29]. Beyond prompt-based agents, recent advances in reinforcement learning [12; 43; 44] and agent distillation, such as Chain-of-Agents [45] and A<sup>2</sup>FM [46], strive to unlock stronger capabilities by optimizing reasoning trajectories and tool-use behaviors end-to-end.

However, a significant gap remains in effectively synthesizing high-quality data to fine-tune open-source models for state-of-the-art deep research. Although TaskCraft [47] has introduced scalable mechanisms for automating complex task generation via width-based extension, and multi-agent distillation works like Chain-of-Agents [45] and A<sup>2</sup>FM [46] have made strides in transfer learning, they do not fully capture the iterative, exploratory essence of the complete deep research process—from initial query planning to final report synthesis. By explicitly modeling this workflow to generate training corpora, we aim to democratize access to powerful research assistants, reducing dependency on proprietary APIs and enabling autonomous, evidence-based reasoning in open-source models.

### 3 Method

Our objective is to develop an end-to-end, open-source model for complex research tasks that minimizes reliance on intricate prompt engineering and hand-designed workflows. We propose a three-stage training pipeline: (1) Design a power and generation deep research report generation workflow and synthesize high-quality trajectory data generation for Supervised Fine-Tuning (SFT), (2) Conduct SFT on an open source language model to empower it to learn the research process, and (3) Conduct reinforcement learning to further enhance the model’s tool-use deep research report generation capacity.

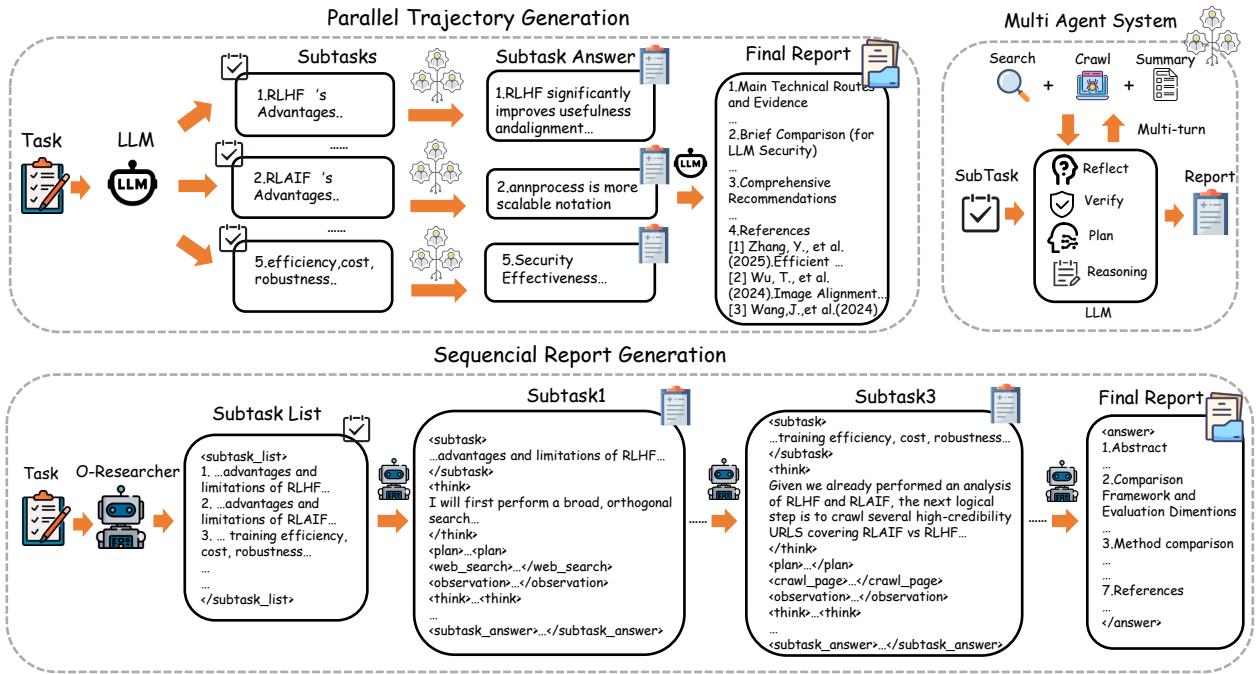
#### 3.1 High-Quality Trajectory Generation for SFT

The foundation of our approach is a meticulously curated dataset of deep research tool-integrated reasoning trajectories. The process involves two key steps: data synthesis and trajectory synthesis.

**Producing Trajectory with Parallel Execution.** As depicted in Figure 2, we designed a parallel execution workflow to generate research reports, simulating the rigorous division of labor in research teams. Drawing on the width-based task extension methodology proposed in TaskCraft [47], we employ a hierarchical strategy to manage complex queries. A given query is first decomposed by a planner into a set of orthogonal sub-queries, effectively transforming a long-horizon problem into manageable, parallelizable atomic tasks. Different models then independently conduct multiple rounds of tool-integrated reasoning—executing a continuous *Plan-Execute-Observe* loop—to produce detailed sub-query reports. These intermediate reports are then aggregated by a dedicated summarizer model to form the final answer. Meanwhile, the trajectories underlying all sub-reports are collected and merged into a single SFT reasoning trace, ensuring the training data captures the full depth of the parallel reasoning process.

**Query Synthesis and Trajectory Collection.** To construct a high-quality corpus for supervised fine-tuning, we curated a seed set of 5,000 queries derived from both mature open-source datasets (including Zhihu-KOL [48], WideSearch [49], and ELIS [50]) and LLM-synthesized topics covering underrepresented domains. These queries were specifically selected for their open-ended nature and high complexity. Following the initial generation by our agentic workflow, we applied rigorous filtering to remove low-quality or short-path trajectories. This process ultimately yielded 3,500+ premium instruction-response pairs, which serve as the foundation for our SFT stage.

To construct high-fidelity training data, we employ a diversity-driven synthesis strategy [45; 46] followed by a rigorous multi-stage rejective sampling pipeline. For each curated query, our agentic framework—comprising a planner, a



**Figure 2** The report generation process of O-Researcher. A query is broken down into multiple sub-queries, which are then independently and parallelized by different agents through tool-integrated reasoning to generate sub-query reports. These sub-query reports are then aggregated through the summarizer agent to generate the final report. All traces and reports of different sub-queries are concatenated as the supervised-training data for this query.

tool-user, and a summarizer—first decomposes the query into sub-tasks. Each sub-task undergoes multi-step *Thought-Tool–Observation* interactions [41]. to generate a corresponding sub-report. Finally, a fusion model consolidates these decomposed sub-reports into a coherent final answer, accompanied by its complete reasoning trace.

**Quality Assurance Pipeline** To ensure the generation of high-fidelity training data, we have designed a comprehensive, multi-stage rejective sampling pipeline. This pipeline acts as a funnel, starting with a diverse set of candidates and progressively filtering them based on increasingly sophisticated criteria, organized into the following stages:

- **Diversity-Driven Generation:** The pipeline begins by generating three distinct candidate trajectories for each query. This initial oversampling is crucial for exploring diverse reasoning paths and increasing the likelihood of producing at least one high-quality output.
- **Rule-Based Hard Rejection:** The generated candidates first undergo a battery of deterministic checks to eliminate structurally flawed or overly simplistic trajectories. A trajectory is immediately rejected if it fails any of the following criteria:
  - **Completeness:** It must contain all necessary tool invocations and correctly closed formatting tags (e.g., <web\_search> and </web\_search>).
  - **Context Length:** The total token count must remain within the effective context window ( $\leq 64k$  tokens).
  - **Complexity Thresholds:** To ensure sufficient depth, the trajectory must demonstrate at least 10 reasoning steps and 5 distinct tool-use actions.
  - **Consistency:** The output must pass strict format validity checks and maintain language consistency with the input query.
- **Model-Based Semantic Filtering:** Trajectories that survive deterministic checks proceed to semantic evaluation by a Qwen3-based LLM-as-a-Judge. This model assesses higher-order qualities beyond rule-based

constraints—including logical coherence, tool-use relevance, and evidential grounding—to identify and select the most optimal trajectory for the final dataset.

- **Human-in-the-Loop Verification:** As the final layer of validation, the highest-rated trajectories undergo topic-stratified human spot-checking. If a sample is flagged as low-quality, it triggers a regeneration loop, where the original query is re-processed until a valid trajectory is produced and verified.

This exhaustive funneling approach ensures that only deep-research trajectories meeting our highest standards are retained for the subsequent SFT and RL phases. Further details are provided in the Appendix.

**Structured Data Representation** To transform raw interaction data into effective training signals, we serialize the trajectories using a coherent XML-style schema. The training sequences are structured to explicitly expose the model’s step-wise reasoning and tool-use behaviors, organized into the following tag categories:

- **Workflow Control Tags:** Define the high-level structure of the problem-solving process.
  - <subtask\_list>: Decomposes the main query into a sequence of manageable sub-problems.
  - <subtask>: Marks the beginning of the execution for a specific sub-problem.
- **Cognitive Tags:** Encapsulate the internal reasoning and planning processes.
  - <think>: Contains the internal reasoning monologue, analysis, and strategic thinking before an action is taken.
  - <plan>: Outlines a concrete, step-by-step action plan derived from the preceding thought process.
- **Action Tags:** Represent executable tool calls that the model learns to invoke to gather external information.
  - <web\_search>: Invokes the web search tool with a set of queries.
  - <crawl\_page>: Invokes the web crawler tool with specific URLs.
- **Feedback Tag:** Contains the raw output returned by a tool after an action is executed.
  - <observation>: Captures the direct results from a tool call (e.g., search snippets, webpage content).
- **Response Tags:** Mark the synthesized conclusions at different stages of the process.
  - <subtask\_answer>: Provides the conclusive answer for a single subtask.
  - <suggested\_answer>: Marks the final, comprehensive report that consolidates all subtask answers.

This format forces the model to adhere to a structured *Thinking–Acting–Observing–Answering* loop, which is critical for cultivating robust and verifiable research capabilities. The prompt template is shown as follows.

#### Available Functions

You may only use the following 8 functions to answer the question. Each function must be enclosed within its corresponding tags.

- **subtask\_list** Break the main question into independent subtasks. Start with <subtask\_list> and end with </subtask\_list>.
- **subtask** Marks the specific subtask being executed. Start with <subtask> and end with </subtask>.
- **think** Internal reasoning before plan/tool. Start with <think> and end with </think>.
- **plan** Break the subtask into detailed micro steps. Start with <plan> and end with </plan>.
- **tool** Invoke an external tool.
- **observation** Holds the tool output.
- **subtask\_answer** Provide the intermediate answer for a subtask. Start with <subtask\_answer> and end with </subtask\_answer>.

- **suggested\_answer** Integrate all subtask answers into the final solution.

#### Available Tools

- <web\_search> Queries separated by |, append &serp\_num=N. Example: <web\_search>AI trends | LLM safety&serp\_num=20</web\_search>
- <crawl\_page> Fetch deeper information from URLs.
- You may call <web\_search> multiple times as needed.

#### Tool Usage Guide

- If retrieved information is irrelevant, refine your queries and repeat <web\_search>.
- Continue searching until you have high-confidence evidence.

#### Trail Notes

- **Workflow:** Start with <subtask\_list>. Then repeatedly perform: think → plan → tool → observation, until sufficient info is gathered, then generate <subtask\_answer>.
- **Information Gathering:** Use tools multiple times when necessary.
- **Tag Restrictions:** Special tags must not appear in free text, especially inside <think>.

#### Function Association Instructions

- Always begin with <subtask\_list>.
- Then start the first <subtask>.
- Inside each subtask, <think> must appear before plan or tool.
- Output <subtask\_answer> once enough information is gathered.
- After all subtasks, output <suggested\_answer>.

#### Answering Tips

- Final <suggested\_answer> must be entirely in English.
- Must include: Introduction, Body, Conclusion, References.
- Every key fact must include a citation like [1] .

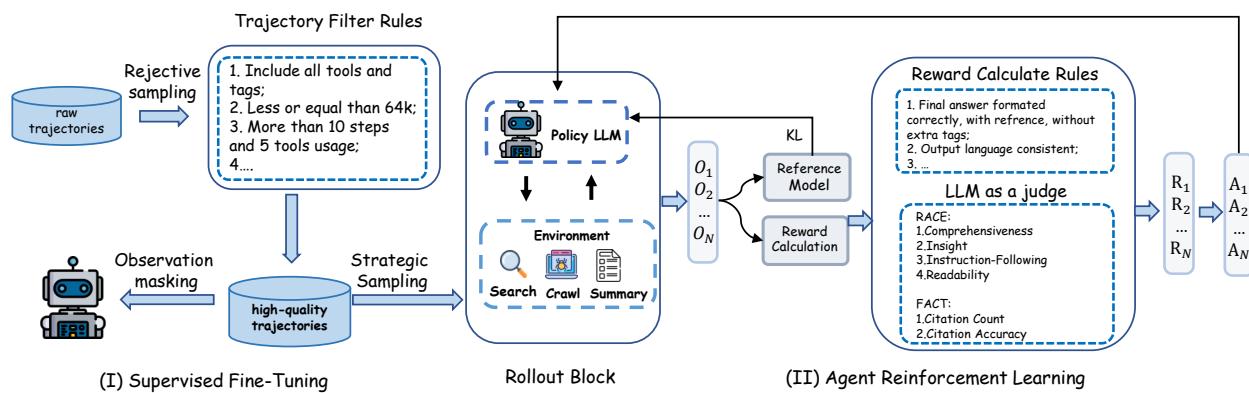
#### References Section

Each reference entry must follow:

- [Number]. URL – Webpage Title

## 3.2 Reinforcement Learning from AI Feedback (RLAIF)

To further enhance the model's capability to produce high-quality, novel, and comprehensive research reports, we implemented a reinforcement learning stage utilizing Group Relative Policy Optimization (GRPO) [51].



**Figure 3** The deep research model training process of our work, which has an SFT stage and an RL stage.

**Preference Data Curation.** As depicted in Figure 3, we began by synthetically generating a diverse set of research questions across multiple domains using an auxiliary large language model (LLM). To construct a preference dataset that is both challenging and informative for reinforcement learning, we filtered these questions based on the performance variance of our supervised fine-tuned (SFT) model. Specifically, for each question, we generated eight distinct responses and evaluated them. Questions that resulted in consistently high scores (indicating trivial difficulty) or consistently low scores (indicating intractable difficulty) were discarded. This filtering approach isolates queries within a "sweet spot" of difficulty, thereby maximizing the learning signal for the policy model during training.

**Reward Function Design.** Our reward function is designed to balance report quality, tool utilization efficiency, and format compliance. It is formulated as a weighted combination of three primary components:

$$R = w_1 R_{\text{base}} + w_2 R_{\text{tool}} + w_3 R_{\text{format}}. \quad (1)$$

We set  $w_1 = 0.6$ ,  $w_2 = 0.2$  and  $w_3 = 0.2$ , placing stronger emphasis on high-quality report generation while still encouraging appropriate tool-use behavior.

**Base Quality Reward** ( $R_{\text{base}}$ ). The base quality reward is obtained from an LLM-as-a-Judge that evaluates each (question, generated report) pair along four dimensions: comprehensiveness, insight, instruction-following, and readability. Each dimension is computed via a weighted sum of its criteria, and  $R_{\text{base}}$  is the average across dimensions.

**Tool-Usage Reward** ( $R_{\text{tool}}$ ). To encourage appropriate evidence collection, we define

$$N_{\text{calls}} = \min(\text{web\_search}, \text{crawl\_page}), \quad N_{\text{min}} = 2, \quad N_{\text{max}} = 8.$$

The tool-usage reward is:

$$R_{\text{tool}} = \begin{cases} 0, & N_{\text{calls}} < N_{\text{min}}, \\ -1, & N_{\text{calls}} > N_{\text{max}}, \\ \frac{N_{\text{calls}} - N_{\text{min}}}{N_{\text{max}} - N_{\text{min}}}, & \text{otherwise.} \end{cases}$$

This design rewards reasonable tool usage while penalizing both insufficient and excessive invocation.

**Formatting Reward** ( $R_{\text{format}}$ ). The formatting reward enforces structural correctness. It verifies two strict conditions: (1) all XML-style tags must be symmetrically closed; (2) the output must contain a `<suggested_answer>` tag. Violations of either condition yield zero formatting reward.

**Final Reward.** The final composite reward is normalized to  $[0, 1]$  to provide a stable training signal.

During GRPO training, we observed several notable trends:

- a steady increase in average response length, suggesting that the model learned to generate more detailed reports;
- a gradual rise in web-search calls and a larger increase in crawl-page calls, indicating deeper evidence gathering;
- an initial sharp decline in policy entropy followed by a gradual rebound, reflecting early convergence to effective strategies before exploring more nuanced behaviors.

## 4 Experiment

### 4.1 Experimental Setup

**Benchmark.** We evaluate all the artifacts on two benchmarks. One is DeepResearch Bench [52]. This benchmark comprises 100 doctoral-level research tasks, designed to test advanced reasoning and information synthesis capabilities. The tasks are distributed across four diverse and knowledge-intensive domains: Science & Technology, Finance & Business, Software Engineering, and Others. The other is DeepResearchGym [31], an open-source framework designed to benchmark deep research systems by combining a reproducible search API with a rigorous, multi-dimensional evaluation protocol that assesses alignment with user information needs, factual faithfulness, and report quality.

**Baselines.** To establish a robust comparison, we include several state-of-the-art systems as baselines. These are categorized into two primary groups based on their architecture and accessibility:

- **Deep Research Agents:** This category encompasses advanced agentic systems designed for multi-step, deep information retrieval, typically employing complex planning and tool-use workflows.
  - *Proprietary Systems:* We compare against leading closed-source commercial agents, including OpenAI Deep Research [53], Gemini-2.5-Pro Deep Research [54], Perplexity Deep Research [55], and Grok Deeper Search by xAI [56].
  - *Open-source Frameworks:* To benchmark against accessible academic works, we include MiroFlow [57] and OAgents [39]. Additionally, we evaluate WebWeaver [16].
- **Deep Research Models:** This category includes Large Language Models (LLMs) that are either inherently integrated with search capabilities or specifically fine-tuned for deep research tasks.
  - *Proprietary Search-enhanced LLMs:* We evaluate representative commercial search-integrated models. From OpenAI, we consider O3, GPT-4.1, and GPT-5 [58]. We also include the Gemini-2.5 family (Pro and Flash) [59], Perplexity Sonar and Sonar-Pro [60], Kimi-K2 [61], and MiniMax M2 [62]. These models represent the state-of-the-art in standard retrieval-augmented generation.
  - *Open-source Models:* To assess the performance of O-Researcher, we compare our SFT and RL variants alongside other state-of-the-art open-weights deep research models, including Tongyi-Deep Research [63] and MiroThinker [64].

**Impact of Context Length.** To investigate the effect of context length on learning complex research workflows, we conducted experiments training on trajectory data truncated to 32k and 64k tokens. Our findings indicate a substantial performance improvement when scaling from 32k to 64k, suggesting that a longer context is crucial for capturing the complete research arc. However, we observed diminishing returns when scaling from 64k to 128k, implying a potential plateau in performance gains from context length alone.

## 4.2 Evaluation Metrics

To provide a multi-faceted assessment of model performance, Deep-Research-Bench [52] employs two sets of metrics: **RACE** for evaluating the qualitative aspects of the report and **FACT** for quantifying its factual correctness and citation quality. These are detailed as follows:

**RACE (Report Quality):** This metric assesses the overall quality and presentation of the report through four criteria:

- **Comprehensiveness:** The extent to which the report covers all key aspects of the query.
- **Insight/Depth:** The level of analysis, novelty, and depth beyond simple information aggregation.
- **Instruction-Following:** How well the report adheres to any implicit or explicit constraints in the user’s query.
- **Readability:** The clarity, structure, and coherence of the writing.

**FACT (Factual Correctness):** This metric focuses on the reliability and accuracy of the report’s content, comprising two main items:

- **Citation Accuracy:** Whether the provided citations accurately support the claims in the text.
- **Effective Citations:** The relevance and quality of the sources cited to support the report’s claims.

## 4.3 Performance Comparison

As shown in Table 1, the experimental results demonstrate that O-Researcher-RL establishes a new state-of-the-art among open-weights deep research models, significantly outperforming concurrent open-source works such as Tongyi-Deep Research [63]<sup>1</sup> and MiroThinker [64]. Furthermore, our model transcends the capabilities of leading search-enhanced commercial LLMs, surpassing both the GPT-5 baseline [58] and OpenAI O3 [58], while also outperforming specialized

<sup>1</sup>FACT cannot be calculated as the model fails to provide reference links.

**Table 1** Overall evaluation results of DeepResearch Bench.

Model	RACE					FACT	
	Overall	Comp.	Depth	Inst.	Read.	C. Acc.	E. Cit.
Deep Research Agents							
Grok Deeper Search	40.24	37.97	35.37	46.30	44.05	83.59	8.15
Perplexity Deep Research	42.25	40.69	39.39	46.40	44.28	<b>90.24</b>	31.26
Gemini-2.5-Pro Deep Research	48.88	48.53	48.50	49.18	<b>49.44</b>	81.44	111.21
O3 Deep Research	46.98	46.87	45.25	49.27	47.14	77.96	40.79
MiroFlow	44.87	44.57	49.30	45.45	46.11	86.98	23.93
WebWeaver	50.62	<b>51.29</b>	51.00	49.98	48.89	78.25	<b>166.73</b>
OAgents-DR	<b>50.76</b>	50.39	<b>51.20</b>	<b>50.32</b>	49.41	36.98	12.56
Deep Research Models							
Claude-3-7-Sonnet	40.67	38.99	37.66	45.77	41.46	93.68	32.48
Claude-3-5-Sonnet	28.48	24.82	22.82	35.12	35.08	<b>94.04</b>	9.78
Perplexity-Sonar-Reasoning-Pro	40.22	37.38	36.11	45.66	44.74	39.36	8.35
Perplexity-Sonar-Pro	38.93	36.38	34.26	44.70	43.35	78.66	14.74
Gemini-2.5-Pro	35.12	34.06	29.79	41.67	37.16	81.81	<b>32.88</b>
Gemini-2.5-Flash	32.39	31.63	26.73	38.82	34.48	81.92	31.08
OpenAI O3	43.71	42.02	38.80	50.29	45.90	19.00	1.10
GPT-5	46.77	45.41	44.54	50.29	47.47	32.25	12.21
GPT-4.1	33.46	29.42	25.38	42.33	40.77	54.1	5.0
Kimi-K2-Thinking	45.65	44.57	43.94	48.97	44.65	11.39	0.97
Kimi-K2	44.47	42.78	39.65	<b>50.82</b>	46.00	5.83	0.27
MiniMax M2	46.06	45.19	44.56	49.01	44.68	18.33	2.88
Tongyi-Deep Research	45.66	44.70	44.20	48.82	44.20	-	-
MiroThinker	41.79	40.22	35.41	47.79	46.64	26.32	0.22
Qwen-2.5-72B-Instruct	33.38	30.27	23.41	43.80	39.49	44.27	8.96
O-Researcher-SFT	46.24 <sub>+12.86↑</sub>	44.41 <sub>+14.14↑</sub>	46.84 <sub>+23.43↑</sub>	46.79 <sub>+2.99↑</sub>	46.76 <sub>+7.27↑</sub>	29.13 <sub>-15.14↓</sub>	22.63 <sub>+13.67↑</sub>
O-Researcher-RL	<b>48.48</b> <sub>+15.10↑</sub>	<b>47.32</b> <sub>+17.05↑</sub>	<b>49.54</b> <sub>+26.13↑</sub>	48.64 <sub>+4.84↑</sub>	<b>47.58</b> <sub>+8.09↑</sub>	31.99 <sub>-12.28↓</sub>	26.01 <sub>+17.05↑</sub>

proprietary agents like Perplexity Deep Research [55]. This indicates that our framework successfully bridges the performance gap between open-source models and top-tier closed-source systems.

The most notable performance transformation is observed when comparing O-Researcher to its backbone, Qwen-2.5-72B-Instruct [65]. While the base model maintains a relatively high citation accuracy (44.27%), it yields a very low volume of effective citations (8.96), reflecting the inherent limitation of general-purpose LLMs in conducting exhaustive evidence-based reports. Through training on deep research trajectories, O-Researcher-SFT learns to execute complex research protocols and generate significantly more comprehensive reports, which is evidenced by a substantial surge in effective citations (+13.67). However, the increased length and complexity of these generated trajectories also introduce a degree of redundancy and noise, leading to a noticeable decline in citation accuracy to 29.13%.

To address this trade-off, our reinforcement learning stage incorporates the reward function (Eq. 1) specifically designed to optimize report effectiveness and factual grounding. As a result, O-Researcher-RL not only achieves the highest overall RACE score of 48.48 but also successfully mitigates the precision loss observed in the SFT stage. Specifically, the RL variant improves Citation Accuracy from 29.13% to 31.99% while further boosting Effective Citations to 26.01. This evolutionary path suggests that our RL strategy effectively filters out redundant information and enforces more rigorous evidence-linking, striking a superior balance between analytical depth and factual rigor. In contrast, other reasoning-centric models such as OpenAI O3 and Kimi-K2-Thinking [61] exhibit critically low citation metrics, highlighting O-Researcher’s unique advantage in providing transparent and verifiable research insights through its optimized training pipeline.

#### 4.4 Evaluations on DeepResearchGym

In our experimental setup, we adopt the "Commercial" configuration, where each system utilizes its proprietary search infrastructure to retrieve information. We report performance on the official 100-task-subset. Table 2 summarizes the performance of various artifacts, categorized into search-enhanced LLMs [61; 62], deep research agents [34; 39; 57; 60], and open-sourced deep research models [36]. Notably, O-Researcher-72B establishes a new state-of-the-art among

**Table 2** Evaluations on DeepResearchGym-Commercial-100.

Artifacts	Relevance		Faithfulness		Quality	
	KPR	KPC	Precision	Recall	Clarity	Insight
Deep Research Agents						
Perplexity Deep Research	66.14	1.82	30.56	<b>98.98</b>	96.3	86.4
O3 Deep Research	62.4	1.56	33.46	83.14	92.8	76.8
MiroFlow	<b>80.55</b>	1.83	25.93	84.48	96.1	95.4
OAgents-DR	80.17	<b>1.16</b>	<b>35.65</b>	97.11	<b>99.5</b>	<b>99.7</b>
Deep Research Models						
Kimi-K2-Thinking	64.32	1.28	10.56	87.78	97.4	82.8
MiniMax M2	37.56	<b>0.91</b>	5.81	50.57	52.6	43.1
Tongyi-DeepResearch	67.33	1.8	-	-	95.90	79.1
O-Researcher-32B	68.35	0.93	32.01	94.44	95.9	91.1
O-Researcher-72B	<b>77.28</b>	1.74	<b>51.45</b>	<b>94.61</b>	<b>100.00</b>	<b>99.3</b>

the evaluated open-source models and remains highly competitive against closed-source commercial agents. In terms of report quality, O-Researcher-72B achieves perfect or near-perfect scores, recording a 100.00 in Clarity and 99.3 in Insight, outperforming commercial baselines such as Sonar-Deep-Research and O3-Deep-Research. Crucially, regarding retrieval faithfulness, O-Researcher-72B exhibits exceptional grounding. It attains a Citation Precision of 51.45, which is the highest across all categories in the table. This indicates that while maintaining high recall, O-Researcher avoids the trade-off of hallucinated citations often observed in other systems. Furthermore, in terms of relevance (KPR), O-Researcher-72B substantially outperforms standard search-enhanced LLMs and effectively rivals specialized agents, demonstrating a robust ability to synthesize comprehensive reports that satisfy complex user information needs.

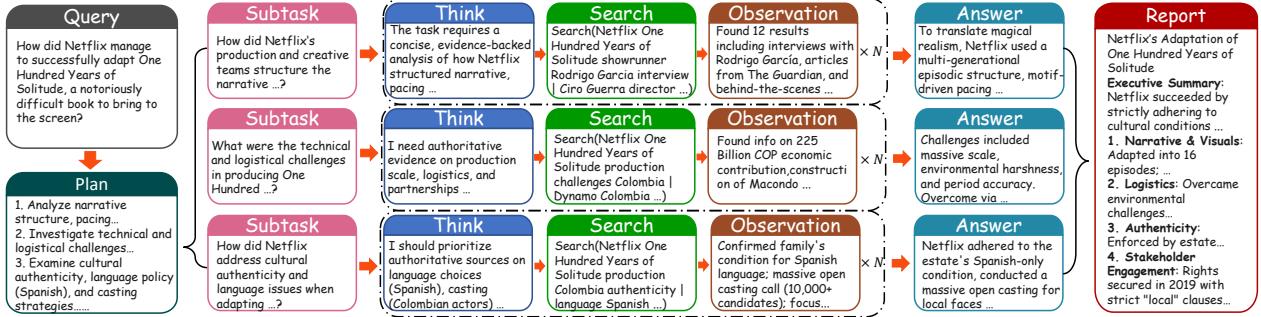
## 5 Discussion

**Parallel execution enables effective report generation.** To quantitatively assess the impact of our proposed methodology, we compare the performance of GPT-5 with and without the parallel execution workflow. As illustrated in **Table 3**, employing parallel execution yields a substantial improvement across all evaluated metrics. The overall score increases significantly from 42.92 to 49.60. More specifically, we observe notable gains in Comprehensiveness (40.59 → 49.61) and Insight (38.58 → 48.69), which are critical dimensions for deep research tasks. This demonstrates that decomposing a complex problem into manageable sub-tasks enables the model to conduct a more thorough investigation and generate deeper analysis. Even metrics with already high baseline performance, such as Instruction Following and Readability, see further improvements. These results strongly validate the necessity of a structured workflow for handling complex research queries, as it provides a systematic framework that guides the model to outperform its prompting counterpart consistently.

**Table 3** The comparison between GPT-5 with and without parallel workflows.

Workflows	Overall	Comp.	Ins.	Inst.	Read.
Sequential Execution	42.92	40.59	38.58	48.05	46.88
<b>Parallel Execution</b>	<b>49.60</b>	<b>49.61</b>	<b>48.69</b>	<b>50.58</b>	<b>50.32</b>

**Qualitative Analysis of Parallel Execution Workflow.** To visualize the efficacy of our parallel execution strategy, Figure 4 presents a case study on the complex adaptation of *One Hundred Years of Solitude*. The workflow demonstrates how O-Researcher decomposes the query into orthogonal dimensions—Narrative Structure, Logistics, and Cultural Authenticity—thereby allowing dedicated agents to execute concurrent *Think-Search-Observation* loops.



**Figure 4** Qualitative case study of Parallel Execution. The complex adaptation task is decomposed into three distinct subtasks (Narrative, Logistics, Authenticity). Each subtask independently undergoes a *Think-Search-Observation* loop, retrieving high-granularity evidence (e.g., specific budget figures, casting numbers) before being synthesized into a comprehensive report. This contrasts with sequential methods that often fail to maintain such depth across multiple dimensions simultaneously.

Crucially, this parallelization mitigates the context competition inherent in sequential generation. By isolating the search contexts, the agents successfully retrieve and retain high-granularity evidence, such as the specific "225 Billion COP economic contribution" and "10,000+ casting candidates," which might otherwise be diluted in a single long-context stream. The final aggregation step synthesizes these independent, evidence-rich threads into a cohesive executive summary. This qualitative evidence reinforces our quantitative findings in Table 3, confirming that structured decomposition is the key driver behind the significant improvements in Comprehensiveness (49.61 vs 40.59) and Insight(48.69 vs 38.58) scores.

**Reasoning Steps for Report Generation.** What is the optimal step number for our report generation workflow? The step number is important since it determines the length of the training data, which directly controls the training cost and the training performance. In this work, we try three different workflow steps: 5, 10, and 20. As illustrated in Table 4, the results indicate that the 10-step workflow consistently outperforms the 5-step approach across all evaluation metrics, achieving a higher overall score (49.61 vs 48.80). Notably, the 10-step workflow shows the most significant improvement in Comprehensiveness. This suggests that a more fine-grained decomposition of tasks enables a more thorough exploration of the research problem. Therefore, we identify the 10-step workflow as the optimal choice, offering the best balance between performance gains and computational expense. When we increase the step number from 10 to 20, there is no obvious performance gain, but a much longer retrieval context. Therefore, we choose 10 as our step number.

**Table 4** Impact of Reasoning Steps. Performance on the OAgents framework

Step Number	Overall	Comp.	Ins.	Inst.	Read.
5 Steps	48.80	47.89	48.31	49.81	49.91
10 Steps	49.61	49.60	48.71	<b>50.62</b>	<b>50.31</b>
20 Steps	<b>50.76</b>	<b>50.39</b>	<b>51.20</b>	50.32	49.41

Our ablation studies reveal two critical factors for enhancing performance in deep research tasks. As shown in Table Table 3, adopting a parallel workflow significantly boosts the overall score of GPT-5 from 42.92 to 49.60. This highlights the importance of structured decomposition for complex queries. Furthermore, Table 4 demonstrates a clear correlation between the number of reasoning steps and performance within the OAgents framework. Increasing the steps from 5 to 20 elevates the overall score from 48.80 to a peak of 50.76. This suggests that allowing the model more "thinking time" through a greater number of steps is essential for achieving state-of-the-art results. Both findings underscore that advanced agentic workflows, rather than raw model capability alone, are key to unlocking top-tier performance on these challenging benchmarks.

## 6 Conclusion

In this work, we introduced a novel multi-agent workflow for the automated synthesis of end-to-end deep research data. By structuring collaboration among multiple LLM agents to decompose, debate, and verify complex tasks, this system provides a scalable pipeline for generating high-quality training corpora. Leveraging this synthesized data, we designed a two-stage training strategy that combines supervised fine-tuning with a novel reinforcement learning method to further refine model performance. Extensive experiments on a major deep research benchmark demonstrate that our framework empowers open-source models of various sizes to achieve new state-of-the-art performance, significantly closing the gap with leading closed-source models and establishing new benchmarks in the open-source domain.

## 7 Contributions

### Core Contributors

- Yi Yao
- He Zhu

### Contributors\*

- Piaohong Wang
- Xinlong Yang
- Xiaowan Li
- Jiaxian Li
- Sinuo Wang
- Jiaqi Wu
- Jincheng Ren
- Qianben Chen
- Dingfeng Shi
- Qiexiang Wang
- Xinpeng Liu
- Minghao Liu

### Corresponding Authors

- Wangchunshu Zhou

## References

- [1] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Keshwam, Ahmad Al-Dahle, Arun Raman, Arthur Feng, Amir Yazdanbakhsh, et al. The llama 3 herd of models. [arXiv preprint arXiv:2407.21783](#), 2024.
- [2] OpenAI. Learning to reason with llms. [OpenAI System Card](#), 2024.
- [3] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. [arXiv preprint arXiv:2408.03314](#), 2024.
- [4] DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Cheng Cheng, Chong Zhang, Yuxuan Xie, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. [arXiv preprint arXiv:2501.12948](#), 2025.
- [5] Jerry Orcutt, Toan Naguyen, Fabio Farina, A. Bashir, Min Cheng, Roman Duda, Eric Fosler-Lussier, and Luke Zettlemoyer. Agentinstruct: Generative agents for high-quality instruction tuning. [arXiv preprint arXiv:2407.03502](#), 2024.
- [6] Lichang Xu, Wei Chen, Yanyan Ge, Xinyu Wang, Hanning Zhu, Rui Feng Zhang, and Julian McAuley. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. [arXiv preprint arXiv:2406.08464](#), 2024.
- [7] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. [International Conference on Learning Representations \(ICLR\)](#), 2024.
- [8] Yijia Shao, Yucheng Jiang, Theodore A Kan, Peter Xu, Omar Khattab, and Monica S Lam. Assisting in writing wikipedia-like articles from scratch with large language models. [arXiv preprint arXiv:2402.14207](#), 2024.
- [9] Junyou Li, Qin Zhang, Yang Yu, Qiang Fu, and Dejing Ye. More agents is all you need. [arXiv preprint arXiv:2402.05120](#), 2024.
- [10] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. [arXiv preprint arXiv:2401.10020](#), 2024.
- [11] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report, 2023.
- [12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.
- [13] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengan Huang, Chenxu Lv, et al. Qwen3 technical report, 2025.
- [14] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. [arXiv preprint arXiv:2507.02592](#), 2025.
- [15] Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Yida Zhao, Liwen Zhang, Litu Ou, Dingchu Zhang, Xixi Wu, Jialong Wu, et al. Websailor-v2: Bridging the chasm to proprietary agents via synthetic data and scalable reinforcement learning. [arXiv preprint arXiv:2509.13305](#), 2025.
- [16] Zijian Li, Xin Guan, Bo Zhang, Shen Huang, Houquan Zhou, Shaopeng Lai, Ming Yan, Yong Jiang, Pengjun Xie, Fei Huang, Jun Zhang, and Jingren Zhou. Webweaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research. [CoRR](#), abs/2509.13312, 2025.
- [17] Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webdancer: Towards autonomous information seeking agency. [CoRR](#), abs/2505.22648, 2025.
- [18] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. [arXiv preprint arXiv:2504.21776](#), 2025.
- [19] Zhengwei Tao, Haiyang Shen, Baixuan Li, Wenbiao Yin, Jialong Wu, Kuan Li, Zhongwang Zhang, Huifeng Yin, Rui Ye, Liwen Zhang, Xinyu Wang, Pengjun Xie, Jingren Zhou, and Yong Jiang. Webleaper: Empowering efficiency and efficacy in webagent via enabling info-rich seeking. [CoRR](#), abs/2510.24697, 2025.
- [20] Wenlin Zhang, Xiaopeng Li, Yingyi Zhang, Pengyue Jia, Yichao Wang, Hufeng Guo, Yong Liu, and Xiangyu Zhao. Deep research: A survey of autonomous research agents, 2025.
- [21] Hui Yang, Sifu Yue, and Yunzhong He. Auto-gpt for online decision making: Benchmarks and additional opinions, 2023.

- [22] Wangchunshu Zhou, Yuchen Eleanor Jiang, Long Li, Jialong Wu, Tiannan Wang, Shi Qiu, Jintian Zhang, Jing Chen, Ruipu Wu, Shuai Wang, Shiding Zhu, Jiyu Chen, Wentao Zhang, Xiangru Tang, Ningyu Zhang, Huajun Chen, Peng Cui, and Mrinmaya Sachan. Agents: An open-source framework for autonomous language agents. 2023.
- [23] Wangchunshu Zhou, Yixin Ou, Shengwei Ding, Long Li, Jialong Wu, Tiannan Wang, Jiamin Chen, Shuai Wang, Xiaohua Xu, Ningyu Zhang, Huajun Chen, and Yuchen Eleanor Jiang. Symbolic learning enables self-evolving agents. 2024.
- [24] Keivalya Pandya and Mehfuzা Holia. Automating customer service using langchain: Building custom open-source gpt chatbot for organizations, 2023.
- [25] Junde Wu, Jiayuan Zhu, and Yuyuan Liu. Agentic reasoning: Reasoning llms with tools for the deep research, 2025.
- [26] Google Team. Introducing gemini deep research. <https://gemini.google/overview/deep-research/>, 2025. Accessed: 2025-04-06.
- [27] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments, 2025.
- [28] Yuxiang Zheng, Shichao Sun, Lin Qiu, Dongyu Ru, Cheng Jiayang, Xuefeng Li, Jifan Lin, Binjie Wang, Yun Luo, Renjie Pan, et al. Openresearcher: Unleashing ai for accelerated scientific research, 2024.
- [29] Zhaorui Yang, Bo Pan, Han Wang, Yiyao Wang, Xingyu Liu, Minfeng Zhu, Bo Zhang, and Wei Chen. Multimodal deepresearcher: Generating text-chart interleaved reports from scratch with agentic framework, 2025.
- [30] Yuan Liang, Jiaxian Li, Yuqing Wang, Piaohong Wang, Motong Tian, Pai Liu, Shuofei Qiao, Runnan Fang, He Zhu, Ge Zhang, Minghao Liu, Yuchen Eleanor Jiang, Ningyu Zhang, and Wangchunshu Zhou. Towards personalized deep research: Benchmarks and evaluations. [CoRR, abs/2509.25106](#), 2025.
- [31] João Coelho et al. Deepresearchgym: A free, transparent, and reproducible evaluation sandbox for deep research. [arXiv preprint arXiv:2505.19253](#), 2025.
- [32] Renjun Xu and Jingwen Peng. A comprehensive survey of deep research: Systems, methodologies, and applications, 2025.
- [33] Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, et al. Deep research agents: A systematic examination and roadmap, 2025.
- [34] OpenAI. Introducing deep research. <https://openai.com/index/introducing-deep-research/>, 2025. Accessed: 2025-04-06.
- [35] Tongyi DeepResearch Team. Tongyi-deepresearch. <https://github.com/Alibaba-NLP/DeepResearch>, 2025.
- [36] Tongyi Lab. Tongyi deepresearch technical report. [arXiv preprint arXiv:2510.24701](#), 2025.
- [37] xAI Team. Introducing grok deepsearch. <https://x.ai/news/grok-3>, 2025. Accessed: 2025-04-06.
- [38] Zhipu AI. Autoglm-research. <https://autoglm-research.zhipuai.cn/>, 2025.
- [39] He Zhu, Tianrui Qin, King Zhu, Heyuan Huang, Yeyi Guan, Jinxiang Xia, Hanhao Li, Yi Yao, Ningning Wang, Pai Liu, Tianhao Peng, Xin Gui, Li Xiaowan, Yuhui Liu, Xiangru Tang, Jian Yang, Ge Zhang, Xitong Gao, Yuchen Eleanor Jiang, Changwang Zhang, Jun Wang, Jiaheng Liu, and Wangchunshu Zhou. Oagents: An empirical study of building effective agents. In [Findings of the Association for Computational Linguistics: EMNLP 2025](#), pages 13354–13369, Suzhou, China, November 2025. Association for Computational Linguistics.
- [40] Jina AI. node-deepresearch. <https://github.com/jina-ai/node-DeepResearch>, 2025.
- [41] Tianrui Qin, Qianben Chen, Sinuo Wang, He Xing, King Zhu, He Zhu, Dingfeng Shi, Xinxin Liu, Ge Zhang, Jiaheng Liu, et al. Flash-searcher: Fast and effective web agents via dag-based parallel execution. [arXiv preprint arXiv:2509.25301](#), 2025.
- [42] Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research assistants, 2025.
- [43] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale, 2025.
- [44] Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization, 2025.

- [45] Weizhen Li, Jianbo Lin, Zhusong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, et al. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl. [arXiv preprint arXiv:2508.13167](#), 2025.
- [46] Qianben Chen, Jingyi Cao, Jiayu Zhang, Tianrui Qin, Xiaowan Li, King Zhu, Dingfeng Shi, He Zhu, Minghao Liu, Xiaobo Liang, et al. A<sup>textsuperscript {2}</sup> fm: An adaptive agent foundation model for tool-aware hybrid reasoning. [arXiv preprint arXiv:2510.12838](#), 2025.
- [47] Dingfeng Shi, Jingyi Cao, Qianben Chen, Weichen Sun, Weizhen Li, Hongxuan Lu, Fangchen Dong, Tianrui Qin, King Zhu, Minghao Liu, et al. Taskcraft: Automated generation of agentic tasks, 2025.
- [48] wangrui6. Zhihu-kol dataset. Hugging Face, sep 2024. Accessed: 2025-11-11.
- [49] ByteDance-Seed. Widesearch dataset. Hugging Face, 2024. Accessed: 2025-11-11.
- [50] sentence-transformers. Eli5 dataset. Hugging Face, 2022. Accessed: 2025-11-11.
- [51] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. [arXiv preprint arXiv:1707.06347](#), 2017.
- [52] Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench: A comprehensive benchmark for deep research agents. [arXiv preprint arXiv:2506.11763](#), 2025.
- [53] OpenAI. Openai deep research: Autonomous agents for comprehensive investigation. <https://openai.com/index/deep-research>, 2025.
- [54] Dave Citron. Try deep research and our new experimental model in gemini, your ai assistant. <https://blog.google/products/gemini/google-gemini-deep-research/>, 2024.
- [55] Perplexity AI. Introducing perplexity deep research. <https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research>, 2025.
- [56] x.AI. Grok. <https://x.ai/api/>, 2025.
- [57] MiroMind AI Team. Miroflow: A high-performance open-source research agent framework. <https://github.com/MiroMindAI/MiroFlow>, 2025.
- [58] OpenAI. Build leading ai products on openai's platform. <https://openai.com/api/>, 2025.
- [59] Google DeepMind. Gemini 2.5: The next generation of multimodal intelligence. <https://deepmind.google/technologies/gemini/>, 2025.
- [60] Perplexity AI. Perplexity sonar and sonar-pro: Enhancing search with llms. <https://docs.perplexity.ai/>, 2025.
- [61] Moonshot AI. Kimi-k2: Moonshot ai large language model. <https://platform.moonshot.cn/>, 2025.
- [62] MiniMax AI. MiniMax M2 & Agent: Ingenious in Simplicity. <https://www.minimax.io/news/minimax-m2>, 2025.
- [63] Team Tongyi DeepResearch, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, et al. Tongyi deepresearch technical report. [arXiv preprint arXiv:2510.24701](#), 2025.
- [64] Team MiroMind, Song Bai, Lidong Bing, Carson Chen, Guanzheng Chen, Yuntao Chen, Zhe Chen, Ziyi Chen, Jifeng Dai, Xuan Dong, et al. Mirothinker: Pushing the performance boundaries of open-source research agents via model, context, and interactive scaling. [arXiv preprint arXiv:2511.11793](#), 2025.
- [65] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. [arXiv preprint arXiv:2412.15115](#), 2024.

## A Case Study

### A.1 Question

What are the current mainstream AI alignment technologies? Please briefly compare their performance in terms of safety when handling large-scale language models.

#### A.1.1 Result From GPT-5:

##### Main Technical Routes and Evidence

- **Supervised Fine-tuning (SFT, including instruction fine-tuning and self-instruction)**
  - **Idea:**
  - **Security Performance:** It can significantly improve "usability/compliance with instructions," but in terms of security, it usually only covers risk types clearly present in the training corpus. Its generalization and adversarial robustness are limited, and it is easily broken by jailbreak prompts. While common as a starting point, it often requires subsequent preference optimization or security filtering. Self-instruction work shows that instruction fine-tuning can improve task compliance and generality, but does not solve the problem of adversarial jailbreaks [5]. Systematic safety assessments (like TruthfulQA and HELM) also suggest that SFT alone has significant room for improvement in "factuality/realism" and "avoidance of inappropriate content" [8; 9].
- **Reinforcement Learning from Human Feedback (RLHF)**
  - **Idea:** Collect human preference comparisons (labeling the outputs of two models as superior or inferior), train a reward model, and then use an RL algorithm like PPO to optimize the policy so that the model's output better aligns with human preferences and safety criteria.
  - **Security Performance:** InstructGPT showed that RLHF significantly improves usefulness and alignment, with a measurable reduction in toxicity and inappropriate content [1]. Anthropic's "Helpful and Harmless Assistant" also proved that RLHF can improve both targets simultaneously, with a lower incidence of harmful responses compared to an SFT-only baseline [2]. Limitations include high cost (human annotation) and potential for over-rejection or being bypassed by jailbreak attacks if reward modeling is inappropriate [7].
- **Reinforcement Learning from AI Feedback (RLAIF) and "Constitutional AI (CAI)"**
  - **Idea:** Use a set of public principles ("constitution") to constrain the model, then use the model itself or an auxiliary model to evaluate preferences, reducing reliance on human annotation. It is essentially AI feedback replacing part of human feedback.
  - **Security Performance:** Anthropic's CAI shows that principle-driven AI feedback can achieve similar or even better performance than human feedback in "harmlessness," and the annotation process is more scalable. The model's rejections are more consistent and traceable to the principles [3]. This approach is better than pure RLHF in cost and scalability, but also faces challenges from jailbreaks [7].
- **Direct Preference Optimization (DPO)**
  - **Idea:** Instead of RL, it uses pairwise preference samples for direct parameter optimization (increasing the log-likelihood of "better" outputs and decreasing that of "worse" ones). It achieves an alignment effect similar to RLHF but with simpler, more stable, and offline training.
  - **Security Performance:** The DPO paper shows it is comparable to or even better than RLHF in multiple preference alignment tasks with lower complexity. For security, its capability depends on whether the preference data covers the "harmlessness" dimension. If so, DPO can inherit these security attributes effectively [2]. However, it is still subject to data distribution limitations and adversarial attacks.
- **Security Classifiers/Guardrails**
  - **Idea:** Add a security classifier and policy filtering (e.g., intercepting self-harm, illegal guidance, hate speech) to the model's input or output. This can also be implemented via security system prompts or

policy templates.

- **Security Performance:** Meta's Llama Guard encodes security policies into a multi-label classifier for auditing inputs/outputs, significantly improving coverage of inappropriate content and easing integration with enterprise policies [6]. These guardrails are effective for known risk categories but face risks of false positives/negatives and need continuous updates.
- **Red Teaming and Adversarial Training**
  - **Idea:** Systematically design jailbreak/adversarial prompts to stress-test the model, using the findings for data augmentation and iterative training to repair vulnerabilities.
  - **Security Performance:** Recent work shows that even LLMs aligned by RLHF/CAI can be jailbroken with a high success rate by automated adversarial methods, demonstrating that existing alignments are not indestructible. Continuous investment in red teaming and defense mechanisms is necessary [7].

## Brief Comparison (for LLM Security)

- **Security Effectiveness (Preventing Inappropriate/Dangerous Outputs)**
  - RLHF and CAI are generally superior to SFT alone, significantly reducing toxicity and providing more consistent rejections [1; 2; 3].
  - DPO, if using data with security preferences, can achieve a similar level of "harmlessness" as RLHF with simpler training [2].
  - Security guardrails (like Llama Guard) have strong interception capabilities for known risk types and are a necessary layer for deployment, but do not replace the alignment of the model itself [6].
- **Cost and Scalability**
  - **RLHF:** High cost of human annotation and slow iteration [1; 2].
  - **CAI/RLAIF:** Relying on AI feedback, the annotation cost is significantly reduced and scalability is better [3].
  - **DPO:** Simpler, offline training pipeline leads to lower cost; the key is obtaining high-quality preference data [2].
  - **Guardrails:** Deployment and maintenance costs are manageable, but policies need continuous updates [6].
- **Robustness and Limitations**
  - Existing alignment methods (SFT/RLHF/CAI/DPO) can be bypassed by carefully designed prompts; research has demonstrated cross-model transferable jailbreak attacks [7]. They must be used with red teaming and multi-layered protection.
- **Balance between Over-rejection and Practicality**
  - If RLHF/CAI overemphasizes "harmlessness," it may result in "over-conservatism." A balance must be achieved through careful reward design and principal trade-offs [2; 3].
- **Authenticity and Factuality (as a security-related dimension)**
  - RLHF (like InstructGPT) outperforms unaligned baselines on truthfulness benchmarks like TruthfulQA. However, "hallucination" is not completely eradicated, requiring further enhancements like search augmentation and fact-checking [1; 8; 9].

## Comprehensive Recommendations

- A common combination in production is: **SFT as a foundation + preference optimization (RLHF/CAI or DPO)** to improve "helpfulness and harmlessness" + **security guardrails** for input/output interception + **continuous red teaming** and adversarial data augmentation.
- If cost and scalability are primary considerations, **CAI/RLAIF** or **DPO** should be prioritized. If the highest quality is the goal, **RLHF** remains a strong choice, provided human annotation resources are available. Regardless of the method, continuous security assessments (e.g., TruthfulQA, HELM) and red teaming are essential to maintain security against real-world threats.

## References

- [1] Ouyang, L., et al. (2022). *Training language models to follow instructions with human feedback*. arXiv:2203.02155.
- [2] Bai, Y., et al. (2022). *Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback*. arXiv:2204.05862.
- [3] Bai, Y., et al. (2022). *Constitutional AI: Harmlessness from AI Feedback*. arXiv:2212.08073.
- [4] Rafailov, R., et al. (2023). *Direct Preference Optimization: Your Language Model is Secretly a Reward Model*. arXiv:2305.18290.
- [5] Wang, Y., et al. (2022). *Self-Instruct: Aligning Language Model with Self Generated Instructions*. arXiv:2212.10560.
- [6] Inan, H., et al. (2023). *Llama Guard: Building Safety Classifiers for Language Models*. arXiv:2312.06638.
- [7] Wei, A., et al. (2023). *Universal and Transferable Adversarial Attacks on Aligned Language Models*. arXiv:2307.15043.
- [8] Lin, S., et al. (2021). *TruthfulQA: Measuring How Models Mimic Human Falsehoods*. arXiv:2109.07958.
- [9] Liang, P., et al. (2022). *Holistic Evaluation of Language Models (HELM)*. arXiv:2211.09110.

### A.1.2 Result From O-Researcher:

#### Abstract

This paper synthesizes reports from multiple subtasks, systematically compares and summarizes the advantages, limitations, and key practical points of mainstream alignment techniques in improving the security of large-scale language models (LLMs), and provides the core principles and representative methods of each technique. Key conclusions include: No single method can comprehensively excel in the dimensions of "usefulness, security, robustness, and cost"; engineering practices often adopt a multi-layer alignment stack (data governance → SFT/instruction fine-tuning → preference alignment → red team/runtime protection) to achieve a compromise [1; 2; 3; 4]. RLHF is effective but costly and lacks robustness [1; 3; 4]. RLAIF significantly reduces costs but is sensitive to the quality of the AI "teacher" and remains fragile against strong adversarial attacks [6; 8; 22]. Policy regularization methods (DPO, etc.) have advantages in sample efficiency but limited jailbreak resilience [9; 19; 20; 21]. This paper provides a structured comparison across multiple dimensions and concludes with practical recommendations for building secure and aligned LLMs.

#### Report Structure

1. Comparative Framework and Evaluation Dimensions
2. Method Comparison (Item-by-Item)
3. List of Mainstream Alignment Techniques
4. Practical Recommendations and Multi-Layer Alignment Stack
5. Conclusions and Research Gaps
6. References

#### 1. Comparative Framework and Evaluation Dimensions

For ease of comparison, the following unified evaluation dimensions are adopted:

- **Direct Safety Effects** (Toxicity, Harmlessness, Bias, Factuality)
- **Usefulness/Helpfulness** (instruction-following, helpfulness metrics)
- **Training efficiency** (data requirements, annotation cost, computation/wall time)
- **Deployment cost** (latency, throughput, model size, inference overhead)
- **Robustness** (red team, jailbreak, hint injection, adversarial examples)
- **Cross-domain/multi-turn dialogue security** (round-cumulative risk, toxicity drift)

- **Governability and auditability** (interpretability, reproducibility)

*(Note: Different studies use different benchmarks. This article cites specific reports to indicate comparability conditions [3; 5; 8; 9].)*

## 2. Method Comparison (Item by Item)

### 2.1 Comparison of RLHF with Other Alignment Techniques

- **Advantages (Security-Oriented)**

- Directly optimizes to human preferences, typically reducing toxicity and improving instruction compliance [1; 3].
- Mature Engineering: Data pipelines, PPO training, and RM training processes have been systematized in the industry [1; 3].

- **Limitations (Security-Oriented)**

- Label consistency and cultural bias issues; RM training may amplify biases or be affected by reward gaming [1; 3].
- Alignment Tax: Security improvements often come at the expense of downstream task performance [1].
- Red team evaluations show that training-period alignment alone cannot completely prevent systematic abuse [4; 9].
- High Cost: High-quality human preference labeling is expensive and difficult to scale [3].

- **Key Comparisons**

- **vs. DPO:** Simpler and more efficient training with results close to RLHF, but with a risk of bias in out-of-distribution safety [2; 3].
- **vs. Constitutional AI (CAI):** Achieves large-scale harmless training via AI self-annotation, but its effectiveness is limited by the accuracy of the constitution [2; 4].
- **vs. SFT:** Lower cost and easier to deploy, but with limited safety improvement. Often used as a prerequisite for RLHF/DPO/CAI [1; 3].

### 2.2 Comparison of RLAIF with Other Alignment Techniques

- **Advantages (Security-Oriented)**

- Using an AI to generate preference labels or scores significantly reduces cost and makes alignment scalable, approaching the effect of RLHF [1; 6].
- d-RLAIF (Online LLM Scoring) eliminates the RM training step, simplifying the process [1].
- Frameworks like SRPO can improve robustness under unstable preferences [23].

- **Limitations (Security-Oriented)**

- Highly sensitive to the quality of the "evaluator/teacher" model [2; 6].
- Robustness to red team/jailbreak evaluation is not significantly better than other training-period alignment methods [8; 9].
- AI-generated labels have poor interpretability and auditability [4].

- **Key Comparisons**

- **vs. RLHF:** Alignment quality is comparable but at a lower cost; however, it requires a strong teacher model [2; 6].
- **vs. DPO:** d-RLAIF avoids training an RM but online invocation incurs deployment costs; C-DPO provides a compromise [1; 3].
- **vs. SFT:** RLAIF can impose more explicit preference targets but relies on AI commenting quality [2].

### 2.3 Four-dimensional comparison of RLAIF / RLHF / Policy Regularization

- **Training efficiency:** RLAIF > DPO ≈ RLHF [1; 2; 6].

- **Deployment cost:** DPO ≈ SFT is lowest; RLAIF increases cost if online scoring is used [1; 2; 10].

- **Robustness (Toxicity/Jailbreak):** All methods are still vulnerable to systematic jailbreaks. DPO needs specific improvements for OOD security [1; 2; 4; 8; 23].
- **Multi-turn Conversation Security:** Multi-turn toxicity drift is an open problem for all methods, requiring specific multi-turn alignment data [3; 18].

## 2.4 Comparison of Policy Regularization (CLIP/CoT) with Other Alignment Methods

- **Advantages**
  - High sample and data efficiency; can significantly reduce online sampling computation [1; 19].
  - CoT alignment can improve inference robustness and cross-task capability [11; 13].
- **Limitations**
  - External rewards (like CLIP) are not entirely consistent with text security goals, causing side effects like bias amplification [13; 20].
  - Limited performance in automated red team/jailbreak evaluations [4].

## 3. List of Mainstream Alignment Techniques

- **A. Instruction Fine-tuning and Preference Alignment:** Core methods to align LLMs to user intent.
  - *Methods:* InstructGPT (SFT→RM→PPO) [3], DPO [4], Constitutional AI [5].
- **B. RLHF and its Safe Variations:** Introduce safety constraints into the RLHF process.
  - *Methods:* Safe RLHF [7], Cost-constrained RLHF [8], PMLR [9].
- **C. Data and Post-Training Layers:** Reduce risks via data construction or post-training filtering.
  - *Methods:* Toxicity fine-tuning, bias classifiers, debiased datasets [1; 6].
- **D. Security During Inference (Guardrails):** Apply rules and filtering to the decoding chain.
  - *Methods:* Guardrail frameworks, restricted decoding, online toxicity classifiers [1].
- **E. Automated/Proxy Alignment Paradigm:** Replace human feedback with model self-criticism.
  - *Methods:* H1–H5, RLAIF, IDA [2].
- **F. Multimodal and Robust Security:** Attack/defense systems for adversarial prompts.
  - *Reviews:* PromptBench and JailbreakBench [6].

## 4. Practical Recommendations and Multi-Layer Alignment Stack

- **Multi-layer alignment stack:**
  1. **Data Governance:** Data depollution, debiasing, and adversarial pollution detection [1].
  2. **Training Phase:** SFT → Preference Alignment (RLHF / DPO / RLAIF / CAI), adding safety constraints as needed [3; 4; 6; 7; 9].
  3. **Model Evaluation:** Multi-benchmark evaluation (helpfulness, toxicity, jailbreak) plus red team testing [8; 10].
  4. **Runtime Protection:** Guardrails, decoding filtering, content review, and continuous online monitoring [1; 4; 8].
- **Method Combinations and Trade-offs:** Prioritize DPO/CAI/RLAIF for computational constraints; prioritize RLHF for human preference consistency. Employ multi-layered protection in safety-critical scenarios [1; 2; 3; 4; 8].
- **Risk Management and Governance:** Establish model cards, red team reports, versioning, and external compliance reviews [1; 2].

## 5. Conclusions and Research Gaps

- **Key Conclusions**
  - No single method excels across "usefulness—safety—robustness—cost." Engineering often employs multi-layered, hybrid approaches [1; 2; 3; 4].
  - RLAIF and DPO offer scalable pathways, but all methods still have shortcomings against strong adversarial attacks and multi-round toxic drift [2; 3; 4; 6; 14].

- Automated alignment (H1–H5) is an important future direction but faces challenges like reward hacking and error accumulation [2].
- **Research Gaps and Suggested Directions**
  - Data and benchmarks for multi-round toxic drift are still insufficient; continuous investment in red teams and benchmarks is needed [8; 10].
  - Provable trade-offs between security, capability, and computation remain unclear. Robust alignment methods across languages and cultures are lacking [1; 2].
  - Reward model robustness, reward hacking detection, and theoretical guarantees for RLHF/RLAIF are important research directions [2; 7; 9].

## References

- [1] Ouyang, L., et al. (2022). *Training language models to follow instructions with human feedback*. arXiv:2203.02155.
- [2] Rafailov, R., et al. (2023). *Direct Preference Optimization*. arXiv:2305.18290.
- [3] Bai, Y., et al. (2022). *Constitutional AI: Harmlessness from AI Feedback*. arXiv:2212.08073.
- [4] Anthropic. (2022). *Red Teaming Language Models to Reduce Harms*. Anthropic Research.
- [5] Bai, Y., et al. (2022). *Constitutional AI: Harmlessness from AI Feedback*. arXiv:2212.08073.
- [6] Lee, K., et al. (2023). *RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback*. arXiv:2309.00267.
- [7] Dai, T., et al. (2023). *Safe RLHF: Aligning LLMs to Human Values and Safety*. arXiv:2301.03251.
- [8] Ray, A., et al. (2022). *Cost-constrained Reinforcement Learning from Human Feedback*. arXiv:2211.11391.
- [9] Go, D., et al. (2022). *Preference Model as a Regularizer (PMLR)*. arXiv:2209.02542.
- [10] Zou, A., et al. (2023). *Gemma-J: Jailbreak Generalization Benchmark for Large Language Models*. arXiv:2306.08862.
- [11] Gehman, S., et al. (2020). *RealToxicityPrompts*. arXiv:1909.11423.
- [12] Gehman, S., et al. (2020). *RealToxicityPrompts (PDF)*. ACL Anthology.
- [13] Liu, Y., et al. (2023). *Making Large Language Models Better Reasoners with Alignment*. arXiv:2309.02144.
- [14] Li, Z., et al. (2024). *Is DPO Superior to PPO for LLM Alignment? A Comprehensive Study*. arXiv:2404.10719v1.
- [15] Liu, Z., et al. (2024). *Incubating Text Classifiers Following User Instruction with Nothing but LLM*. arXiv:2404.10877.
- [16] Liu, J., et al. (2024). *Towards Scalable Automated Alignment of LLMs: A Survey*. arXiv:2406.01252v3.
- [17] Zhang, Z., et al. (2024). *Large Language Model Safety: A Holistic Survey*. arXiv:2412.17686.
- [18] Wang, J., et al. (2024). *Multi-Turn LLMs: Safety Not Guaranteed*. arXiv:2404.10876.
- [19] Zhang, Y., et al. (2025). *Efficient Safety Alignment of Large Language Models*. arXiv:2503.10093.
- [20] Wu, T., et al. (2024). *Image Alignment Leads to Misalignment*. arXiv:2406.07885v1.
- [21] Zheng, R., et al. (2024). *Enhancing LLM Safety via Constrained Direct Preference Optimization (C-DPO)*. arXiv:2403.02475.
- [22] Casper, S., et al. (2024). *A Critical Evaluation of AI Feedback for Aligning Large Language Models*. arXiv:2402.12366.
- [23] Chen, Y., et al. (2024). *Self-Improving Robust Preference Optimization (SRPO)*. arXiv:2406.01660.