Reasoning AI models, their prompting and Deep Research tools

1. Introduction to reasoning models

Reasoning models are large language models (LLMs) designed to "think" in multiple steps before answering. Unlike classical LLMs (such as GPT-4 or ChatGPT), they do not try to predict the final answer immediately, but decompose a complex problem into partial *chain-of-thought* steps (A Visual Guide to Reasoning LLMs - by Maarten Grootendorst). This makes them more akin to human reasoning and able to solve more difficult problems, especially in mathematics, logic or programming. OpenAI trains this new family of models (referred to as *o1*) using reinforcement learning so that the model "rubs" its thinking processes, tries different strategies, and can recognize and correct its own errors (Introducing OpenAI o1| OpenAI). As a result, the reasoning model can solve a much more complex problem than the classical model -
For example, in the International Mathematical Olympiad test, it solved 83% of the examples correctly, while the original GPT-4o only solved 13% (Introducing OpenAI o1| OpenAI).

(A Visual Guide to Reasoning LLMs - by Maarten Grootendorst) *Diagram comparing a classic "regular" LLM (left) with a "reasoning" LLM (right). Reasoning models insert a sequence of internal thought steps (in red) before the final answer, essentially "reasoning" and checking before answering (A Visual Guide to Reasoning LLMs - by Maarten Grootendorst).*

Response generation process: a classic LLM like ChatGPT generates a response in mostly one step - it processes the input question in one step pass and immediately generates the final text. If the problem is complex (e.g., a logic puzzle), the classical model often needs to prompt the user to think step by step (e.g., *the prompt "Let's think step by step"*), otherwise it may "jump" to the wrong conclusion. In contrast, the Reasoning LLM has *built-in* multi-step reasoning directly into the model (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). This means that the model internally builds and traverses a chain of partial inferences (without necessarily displaying them to the user) before formulating a conclusion (A Visual Guide to Reasoning LLMs - by Maarten Grootendorst). This enables him to analyse the assignment consistently, choose solution strategy and control itself during generation. OpenAI o1 models have been trained to spend more time "thinking" before answering in a similar way to a human - during practice they learn to refine the solution process and recognize when they are going in the wrong direction (Introducing OpenAI o1| OpenAI). While this process is slower, it significantly reduces the occurrence of major errors on difficult problems (the o1 preview had fewer major errors than the original GPT-4) (OpenAI o1 explained: Everything you need to know).

Suitability of reasoning models for different tasks: thanks to the ability of internal reasoning, reasoning LLMs excel especially in complex tasks that require logical inference, multi-step procedure or detailed analysis. Examples include:

- Mathematical and logical problems: E.g., competitive examples in physics, chemistry or mathematics, complex word problems. (GPT-4o solved only ~13% of these, while the reasoning model solved 83% (Introducing OpenAI o1| OpenAI).)
- Programming and debugging: solving complex programming tasks and debugging code. The Reasoning model achieves high scores in coding competitions (e.g. 89th percentile on Codeforces) (Introducing OpenAI o1| OpenAI) and can step through code and find bugs.
- Scientific and technical analysis: annotation of genomic data, derivation of patterns in quantum optics, etc., where multiple facts need to be logically combined (Introducing OpenAI o1| OpenAI).
- Large-scale tasks with long context: Reasoning models support extremely long input (OpenAI o1 up to 128k tokens) (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub), so they can absorb and analyze, for example, an entire chapter of text or the results of multiple experiments at once.

Conversely, for common and straightforward tasks, the reasoning model may not always be the ideal choice. OpenAI reports that in many common cases (e.g., conversations, general knowledge queries, or creative writing), the standard GPT-4o is so far more powerful and practical (Introducing OpenAI o1| OpenAI). Classic models excel in:

- Broad knowledge of training data: the GPT-4 has a vast general knowledge of the world and can generate text fluently. It can respond well general queries, which a reasoning model (specialized for solving) may not - e.g., o1-preview could not answer a question about itself due to narrower knowledge outside the training domain (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub).
- Conversation and creative content:ChatGPT (GPT-4) is tuned for friendly conversation, explaining in different styles or creating stories. Reasoning models focus on analytical correctness; the tone of their responses tends to be more dry and technical. For example, for writing an essay or a fable, the classical model is more appropriate.
- Quick answers: For simple questions (e.g. factual), ChatGPT will give an answer in seconds. In contrast, the reasoning model can cause unnecessary "thinking" and delays. For OpenAI o1, progress indicators were introduced because some questions take it noticeably longer (OpenAI o1 explained: Everything you need to know). For jobs where speed is more important than perfect depth, the classic model is better.

Below is a summary of the suitability/deficiencies for different types of tasks:

Task type	Classical LLM (GPT-4)	Reasoning LLM (o1 etc.)
Complex mathematical problems	Limited power - GPT-4 solved only 13 % of examples from IMO qualifications ([Introducing OpenAI o1	OpenAI](https://openai.com/index/introducing-openai-o1-preview/#:~:text= In%20hour%20tests%2C%20the%20next,in%20hour%20technical%20research%20post)).
Competitive programming	Very good (GPT-4 reaches ~20% on Codeforces)	Excellent - o1 scores 80-90% on programming benchmarks (OpenAI o1 explained: Everything you need to know).
General knowledge questions	Excellent - Broad knowledge through training on the web data.	May lack context outside its domain (must provide info in the prompt) ([Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models
Conversation and creative Writing	Natural and creative Outcomes, tuned for Dialogue.	Formal and analytical style, focused on precision rather than creativity.
Speed Answers	Quick - usually in order seconds.	Slower - takes longer to "think", minutes for complex queries (has progress bar) (OpenAI o1 explained: Everything you need to know).
Current knowledge	Without tools does not know the data after 2021; with web browsing plugin can search for info.	It does not yet have direct access to the web (o1-preview could not browse the web ([Introducing OpenAI o1

History and overview of reasoning models: the first LLMs (GPT-3, GPT-4) showed surprising emergent abilities to solve complex problems, often hallucinated and "skipped" important steps. Therefore, researchers came up with techniques such as *Chain-of-Thought prompting* (2022) - where the model is instructed to "think it through step by step" - which improved the success rate of complex tasks (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). In 2023, other techniques appeared (self-consistency, tree-of-thought, etc.), but these were still *tricks in the prompts* for classic models.

The turning point came in 2024, when models explicitly trained to reason emerged:

- OpenAI o1 (2024) - the first publicly available series of reasoning models. *The o1-preview* was released in September 2024 (Introducing OpenAI o1| OpenAI), followed by *o1* in December 2024 as part of ChatGPT Pro. These models have a built-in chain-of-thought and achieve state-of-the-art results in STEM domains (Introducing OpenAI o1| OpenAI) (OpenAI o1 explained: Everything you need to know). O1 can also parse images and has a context of 128k tokens. There is also o1-mini, a smaller and cheaper variant aimed at fast programming (80% cheaper to run than o1) (Introducing OpenAI o1| OpenAI). The drawbacks of o1 so far are its limited availability (only for paying users/API Tier 5) and higher price; moreover, in an early version it did not have integration of tools such as web browsing or file upload (Introducing OpenAI o1| OpenAI).
- Google Gemini (2024) - the next generation of Google DeepMind models. Gemini 1.5 Pro, launched in December 2024, includes *Deep Research agent mode* (see Chapter 3) and context for up to 1 million tokens (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental).

Emphasizes *multi-step planning* - for complex queries, it first proposes a solution plan that the user can modify, and then iteratively searches the web and compiles findings (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental). The first version had lower factual accuracy (about 6% in Humanity's Last Exam benchmark (Perplexity Deep Research Takes on OpenAI & Gemini)), but has great potential due to its strong integration with Google search and agent architecture. It is available as part of Gemini Advanced (~$20/month) in English (Perplexity Deep Research Takes on OpenAI & Gemini).

- Open-source projects (2024) - community efforts have led to the emergence of open-source reasoning models. A notable effort is DeepSeek- R1 (2024) - an open model trained with RL that is close in performance to proprietary models (A Visual Guide to Reasoning LLMs - by Maarten Grootendorst). DeepSeek uses the Mixture-of-Experts (MoE) architecture and has achieved considerable success in e.g. programming tasks (51.6% on Codeforces vs. only 23.6% on GPT-4 (DeepSeek)) and on the AIME math test (vs. 9%) (DeepSeek). The developers have released it for free (both the model weights and the API), so it can be deployed for free (A Visual Guide to Reasoning LLMs - by on GPT-4 Maarten Grootendorst). Limitations may be more complex to use (custom HW for 70+ billion parameters is required) and slightly lower reliability or world knowledge compared to giant closed models.
- Other models include Anthropic Claude 2 (2023) - although not specifically trained by RL for reasoning, it can also solve complex problems thanks to 100k context and improved training. Elon Musk has introduced xAI Grok (2023) with internet access, but so far it lags far behind tests (Grok-2 only scored ~3.8% in the HLE benchmark (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology)). One can expect other players (IBM, Meta, etc.) to include reasoning elements in their in future LLMs.

Summary: *Reasoning AI models* represent a new approach where models simulate analytical thinking in steps during inference. As a result, they can solve complex problems in mathematics, programming or science with high accuracy, where classical LLMs have encountered their limits. At the cost of higher computational demands and slower response times, they deliver dramatic improvements - for example, they outperform GPT-4 by tens of percent in tests (Introducing OpenAI o1| OpenAI). But for common everyday queries or creative work, the classic models remain more practical. The development of reasoning models is accelerating (OpenAI o1, Google Gemini, DeepSeek, etc.) and their availability is gradually increasing. This opens up new opportunities for developers to let AI *think* complex problems *through* in depth and get more reliable results where LLMs have failed so far.

---

2. Prompting reasoning models

To get the most out of reasoning models, we need to adapt the style of query generation, or prompt engineering. Let us first recall the basic rules of prompting in classical LLMs and then highlight the differences in reasoning models.

Basic rules of prompt engineering (for classic models): several best practices apply to common LLMs like ChatGPT (AI Prompt Best Practices: Learn Prompt Engineering) (AI Prompt Best Practices: Learn Prompt Engineering):

- Clarity and specificity: make the question unambiguous. Avoid vague wording. For example, instead of *"Tell me about AI,"* prefer *"Explain the main differences between artificial intelligence and machine learning in~ 300 words"*. The more specifically you describe the required content (topic, context, perspective) and form (e.g., "in bullet points", "200-word summary"), better the model will understand the assignment (AI Prompt Best Practices: Learn Prompt Engineering).
- Provide context: the model does not have a permanent memory of the conversation or knowledge of your intent, so provide it with all the context it needs. If the question is tied to a previous discussion, remind the model of key facts. E.g., *"Let's build on the previous text: [summary]. Based on it, answer..."* This will give the model a clear starting point (AI Prompt Best Practices: Learn Prompt Engineering).
- Structuring and format: Tell the model what format to answer in. You can explicitly state *"respond as a numbered list"* or *"return JSON with the following fields"*. You can also structure the prompt itself - e.g. bullet point the requirements. A good trick is to set the role: *"Imagine you are an experienced doctor..."*, which will give the model a hint about the tone and style of the response (AI Prompt Best Practices: Learn Prompt Engineering) (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub).
- Iteration and refinement: it's rare to get the prompt right on the first try. Think of it as an iterative process: *input→ output → modifying the→ entry for better output*. If the model's answer is not to your liking, refine the query, add constraints, or break the problem into smaller parts and try again (AI Prompt Best Practices: Learn Prompt Engineering). The model is sensitive to wording, sometimes just a slight change in wording or question will help.
- Remove ambiguity: identify potentially unclear terms and clarify them. For example, the question *"Tell me about bank deposits"* is ambiguous (bank vs. riverbank in English *banks*). A better question is *"Explain how bank deposit accounts work in personal finance"*.
  If you need to specify a perspective (legal, historical) or what to leave out, please specify. It is also worth mentioning what the model *should not* mention, e.g., *"Do not include technical jargon"* (AI Prompt Best Practices: Learn Prompt Engineering).
- Examples (few-shot): if the task is complex or the format is unusual, you can show the model one or two examples of the input and the desired output (*few-shot learning*). For example, *"Example: Input X -> Response Y."* This will give the model a pattern to follow. For classical models, this will often help increase the accuracy of the interpretation of the input. (For reasoning models, by contrast, see below.)

These principles apply to all LLMs in general. However, Reasoning models have some peculiarities that are good to consider when prompting (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub) (Prompt Engineering for OpenAI's O1 and

O3-mini Reasoning Models| Microsoft Community Hub):

- Built-in chaining of thoughts: In a reasoning model, there is no need to encourage it to reason in steps - it does it automatically. For example, there is no need to add phrases like *"Think carefully"* or *"step by step"*. For OpenAI's o1/o3 models, this would be a waste of tokens and may even be counterproductive (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). (These guidelines make sense for GPT-4o, which otherwise does not run multi-step mode (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub).) A brief direct query is often sufficient; the model will work out the internal logic on its own.
- Need to add knowledge: Reasoning models (e.g., o1) have a narrower scope of knowledge beyond their training focus (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). They are not connected to the Internet (unless you use a special tool) and may lack some facts that a regular GPT-4 would know. Therefore, if the query is about something obscure or very new, provide the information in the prompt. For example, if you want a legal analysis and the model does not have the law in mind, provide the text of the law as part of the prompt. You would not need to do this for the classic GPT-4 - it has a broader encyclopedic awareness. For a reasoning model, however, you'd better not leave anything important out (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub) (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub).
- Use the big context, but don't abuse it: These models support extremely long inputs (hundreds of pages of text). This is great for complex assignments - you can insert all relevant documents, data, etc. But be careful about relevance: too much ballast can confuse or unnecessarily burden the model (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). Give only the information you really need and leave out irrelevant passages.
- Minimum of examples: the O1 and co. generally don't need a *few-shot* examples - they are trained to understand the task from the description. The recommendation is to start with a *zero-shot*, i.e. just clearly describe the task without examples (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). If the model does not fully understand the format, just give one simple example. However, definitely do not give long strings of examples, as this can *degrade performance* (the model will start to mimic the example rather than solving independently) (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). For example, for legal analysis, there is no need to demonstrate a complete sample case analysis in the Prompt - rather, just type *"Use IRAC (Issue, Rule, Analysis, Conclusion) format"* and the model will already organize the answer accordingly (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub).
- Setting the role and tone: even with the reasoning model, the initial system message helps to set the right style. For example, *"You are an experienced legal analyst who explains the application of the law in a clear and logical manner..."* helps the model choose the appropriate formal tone (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). The model could handle the reasoning itself without this, but the role will ensure consistency of output (e.g., that it cites facts, uses industry terminology, etc.).
- Format and scope specifications: These models follow the formatting instructions very well, so take advantage of that. For example, *"sort output into a table"*, *"reply with JSON with the structure..."*, or *"provide a five bullet answer"*. This will prevent ambiguity in the output (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). Also, make the required detail : *"respond briefly in one paragraph"* vs. *"provide a detailed analysis in several paragraphs"* (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). The model itself does not know how much detail you want, but it follows the instruction.clear
- Check and verify: even the reasoning model can be wrong. For critical problems, it is a good idea to use a follow-up query for verification. For example, ask *"Are you sure of your conclusion? Explain why."* Or have the model recapitulate how it reached its conclusion. O1 tends to self-check - it can alert itself if something doesn't fit (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst - - Campus Technology). This can be used to *"Check that you have used all the facts and that the conclusion really follows from the law."* If something he has left out, he will acknowledge and add it in a subsequent response (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). To be absolutely sure, you can run the model multiple times and compare the results (consistency suggests reliability). course, Offinal verification of sensitive results should be done by a human.

Let's summarize the best practices of Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models into an overview | Microsoft Community Hub) (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub):

- Formulate the task clearly and directly, don't wrap it around unnecessarily. The Reasoning model doesn't need as rich a conversation context as ChatGPT - it just needs a specific question/problem.
- Provide all the key information the model needs to solve the problem (especially if it is specialized or up-to-date). Conversely, omit irrelevant text that would distract from it.
- Do not use long prompt-paths and examples unless necessary. O1 does not rely on copying examples, it can handle *zero-shot*. One example at most to clarify the format.
- Specify the required format and detail, the model will follow it. Set the role or style if the response is to be a formal message, list of steps, function code, etc.
- Don't force a chain-of-thought in the prompt - you don't need to with a reasoning model and you may lose valuable tokens (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). On the other hand, with classic GPT-4 you would do this (to reason at all).

- Iterate and test: even though o1 can usually handle complex tasks the first time, don't be afraid to tweak the prompt if the result is not as expected (e.g. add length constraints). A small change in wording can significantly improve the result.
- Verify important outputs: when deployed in practice, ensure that critical responses are verified. Use the model for self-checking (follow-up questioning), and of course the final expert check.

Sample prompts and their explanations:

Example 1 - Legal analysis:
Consider that we need the model to assess whether a particular situation fulfils the characteristics of a breach of contract. Bad prompt (for GPT-4o): *"Is A liable for breach of contract?"* - is too concise and the classical model might hallucinate. Better prompt (for reasoning model o1):

> System: *you are a legal analyst who arbitrates disputes under US contract law.*
> User: *The parties have entered into a contract for the supply of goods. A agrees to deliver 100 units by May 1, for which B will pay him $10,000. A delivered only 80 units and did not deliver the rest at all. B paid only $8,000.*
> *Question: Has Party A breached the contract and can Party B seek damages? Explain your conclusions logically and use the IRAC (Issue, Rule, Analysis, Conclusion) format.*

This prompt clearly states the facts (who did what) in a separate paragraph, providing the model with initial context. An explicit legal question follows. With the role of "legal analyst" set, the model knows what tonality and style to use (formal, factual). The IRAC format tells it how to structure the answer. No need to give an example of a ready-made solution - the model follows the format. All the key information (delivery date, quantity, amount) is included, so the model doesn't have to guess. The O1 then internally goes through the logical steps in the response - identifies the breach (breach of the obligation to deliver), applies the rule (contractual obligation), assesses that A has not complied so B has a claim etc, and writes it up neatly according to IRAC. If we asked GPT-4 to do the same thing, we would have to exhort it to e.g. "do a detailed step-by-step legal analysis...", otherwise it might not list all the reasoning. (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub) (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub)

Example 2 - Mathematical problem:
Classical GPT-4 often stalls on complex math problems unless you explicitly tell it to "count in steps". The Reasoning model does this automatically. Prompt:

> *"Solve the equation `x^3 - 2x^2 - 5x+ 6= 0`. Find all the real roots and explain the procedure."*

For GPT-4 it might help to add "step by step", but for the reasoning model it is not necessary - it starts trying the divisibility of polynomials, finds the root of x=1, does the polynomial division etc., all internally, and then answers: the roots are 1, -2, 3 and describes how it came up with it. The user doesn't have to run the model - he knows "how to solve" thanks to RL training, not just "what to solve" (A Visual Guide to Reasoning LLMs - by Maarten Grootendorst).

Recap: Prompting reasoning models build on the general foundations (clear instructions, context, format), except that the model doesn't need as much hand-holding about logic. Let it work - just tell it exactly *what to solve* and give it the data to do it if necessary. You don't (and shouldn't) need to overwhelm him with examples or encourage him to think out loud. The important thing is to give him all the pieces of the puzzle (facts, definitions) needed to solve it and define what the result should look like. Then you can expect a well-structured, logically reasoned answer. Of course, the modeller's "trust but verify" still applies - for critical outputs, you'd better perform verification (either by further questioning or by your own checking).

---

3. Deep Research Tools

With classical LLMs (including reasoning models) we run into a limit to their knowledge: the model knows nothing about events after its training date and can *make up* facts (*hallucinations*). Deep Research tools have emerged as a solution to this problem. They are AI agents that combine LLM with active online information retrieval and analysis. It's like combining a model with a web search engine and a researcher in one - the model *itself* crawls the internet, collects relevant data from multiple sources, and composes a detailed answer with citations (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained) (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology). These tools can do in minutes what would take a human analyst hours or days: search hundreds of pages, filter out relevant information, collate it, and write a structured report with links to original sources (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained).

Characteristics and Principle: Deep Research (abbreviated as *DR*) operates as an autonomous agent on top of LLM. The user enters a complex query or topic, and the agent then iteratively searches the web for relevant content (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained) (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained). This is not one simple web query - the agent performs many queries: it always reads something, finds new clues, and searches further based on those clues, much like a human would do by clicking through links sequentially (Perplexity Deep Research Takes on OpenAI & Gemini) (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental). During this process, the LLM not only generates the final answer, but also uses it to analyze the found texts (it can "read" articles, PDF documents, even interpret images or graphs) (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained). Importantly, the DR agent cross-checks information - comparing different sources, looking for matches and inconsistencies (Deep Research in AI: Advanced

Methodologies & Breakthrough Applications Explained) (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained). This greatly reduces the risk of hallucinations: it does not rely on the LLM's "memory" alone, but verifies facts in real sources (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained). The output is then a comprehensive report - not just a short answer, but the entire summaries of findings, often divided into clear sections, accompanied by a list of references (citations) to original articles/studies (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology). OpenAI likens the DR agent's response to rather an elaboration by a skilled analyst than the usual short chatbot response (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained).

Specific implementations vary (see tools below), but in general DR agents include the following key components:

- Multi-source analysis: parallel and sequential drawing from and comparing multiple sources (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained). For example, an agent finds 10 articles on a topic, extracts relevant data from them, and tracks whether
  whether they repeat key information or contradict each other.
- Summarization: after collecting the data, the model will summarize it into an understandable form (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained).
  It explains complex topics simply, extracting the essentials from large amounts of data. Often divides output into chapters with headings, uses bullets, tables, etc. to improve readability.
- Insight extraction: it's not just about retelling sources - the agent actively draws conclusions and connections (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained). For example, it identifies trends across studies, highlights key statistics, pulls information together, and may add a synthesis or recommendation at the end. Essentially, it mimics an expert "what's the bottom line" report.
- Citations and transparency: every statement in the report is supported by a link to the source (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology). This allows the user to click and verify that the information is indeed from a credible article. DR agents also have a log of what they are doing when they search -- some (like OpenAI) show the progress of the analysis so you can see what queries and pages they are going through (Perplexity Unveils Deep Research: AI-Powered Tool for Advanced Analysis - InfoQ) (Perplexity Unveils Deep Research: AI-Powered Tool for Advanced Analysis - InfoQ).

The whole process runs autonomously for several minutes. For example, the OpenAI Deep Research agent usually works for 5-30 minutes on a single complex question (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology) (depending on the settings; the user waits in the meantime as if a researcher is working for him). Perplexity AI quotes a length of under 3 minutes for most queries due to optimizations (Perplexity Deep Research Takes on OpenAI & Gemini) (Perplexity Deep Research Takes on OpenAI & Gemini). Google Gemini agent does this in about 15 minutes
(Perplexity Deep Research Takes on OpenAI & Gemini). In general, the more thorough and accurate the output, the longer the agent searches (e.g., the OpenAI agent goes more in depth than the fast Perplexity agent, see comparison below).

Examples of practical use: deep research tools are used where you need to quickly get a detailed overview of a topic or problem based on up-to-date information. Some illustrations of real-life applications:

- Financial analysis: A financial markets analyst may task an agent to produce a comprehensive report on a particular sector - e.g. *"Find out the current trends in X commodity prices, including factors influencing growth or decline, and cite analysis from the last 3 months"*. The agent will review financial news, economist blogs, and expert analysis where appropriate, and deliver a summary with charts and links. OpenAI is specifically targeting finance or policy , for whom DR will save hours of work searching for background information (New OpenAI 'Deep Research' Agent Turns ChatGPT into professionalsa Research Analyst -- Campus Technology) (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology).

- Technology research: imagine a postgraduate student who needs to quickly get to grips with the latest trends
  sensors for autonomous vehicles for the presentation (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental). Instead of reading dozens of articles, the DR agent is tasked with, *"Find out what types of sensors are used autonomous cars, compare their advantages/disadvantages, and describe new technologies on the horizon."* Within minutes, the agent searches for relevant academic papers, industry blogs, product reports, and delivers a summary that includes sections on LiDAR, radar and camera systems, a table comparing range and accuracy, and a mention experimental sensors with links to articles. What would otherwise take a week of research, a student has in an afternoon.

- Marketing and market research: A manager preparing a strategy can use DR to analyze competitors. E.g. *"Prepare a competitive analysis for the smartwatch market - what major models have competitors launched in the last year, their key features and pricing strategy, plus user reviews."* The agent will go through tech sites, user reviews and forums, and deliver a report with a competitor overview, a table
  Google lists just such a use-case: a small entrepreneur can use to quickly map the competition and make suggestions on where to take his business (GeminiTry Deep Research and Gemini 2.0 Flash Experimental)DR : .

- Personal decision making and consumer research: even individuals can use DR to make informed decisions. For example, someone wants to buy a new car - instead of reading reviews, he or she types in different things around the web: *"Compare my car models A, B, C in terms of reliability, fuel economy and safety. Find out from tests and reviews how they perform and recommend which to choose."* The agent gathers data from auto magazines, crash test results, user experiences on forums, and submits a summary with a table plus a link to sources. OpenAI mentions that DR can also assist consumer decisions, such as comparing furniture or electronics (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology). The difference from regular search is that you get a *synthesis*, not just a list of links.

In short, Deep Research is useful wherever you would normally spend a lot of time looking for and reading up on current information. The DR agent does this for you and serves up the finished report. Of course, it makes sense for more complex queries - asking it for a famous person's date of birth is pointless (Google can do that in a flash). It is designed for complex research queries, "complex questions requiring expert level analysis" (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology).

Available DR tools: currently (beginning 2025) there are several major tools and services that offer Deep Research functionality. The following table summarizes their manufacturers, features, limitations and pricing:

Tool (manufacturer)	Functions and features	Limitations	Price and availability
OpenAI ChatGPT - Deep Research (OpenAI)	Advanced DR agent integrated in ChatGPT. It runs on a specialized version of the GPT-4 model (called *o3*) with an emphasis on deep verification and analysis (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology). Spends up to ~30 min. on independent research, process text, images and PDFs (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained) (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained). The result is a highly detailed report (often dozens of pages) with structured sections, tables, charts, and~ dozens of citations (Perplexity Unveils Deep Research: AI-Powered Tool for Advanced Analysis - InfoQ) (Perplexity Unveils Deep Research: AI-Powered Tool for Advanced Analysis - InfoQ). accuracy It has the highest on tests, scoring 26.on Humanity's Last Exam (vs. only 3.3% for GPT-4o) (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology), beating even the competition (Gemini ~6%) thanks to iterative approach and minimal hallucinations.	Very high demands on time and resources - it takes 5-30 min. and is "expensive". Currently limited to 100 queries per month and only for users in the USA (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology). It is not yet globally accessible (status February 2025). It lacks *real-time* multimodal functionality during (graphs are generated at the end). It is a premium feature - not included in the regular Plus account.	$200/month - available in ChatGPT Pro (separate highest paid layer) (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology). Gradually, it is to be made available to ChatGPT Plus/Team/Enterprise users (with lower limits) (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology).
Perplexity AI - Deep Research (Perplexity)	A standalone web tool and API focused on fast DR. It scans dozens of sources and generates a report in ~3 minutes (Perplexity Deep Research Takes on OpenAI & Gemini) (Perplexity Deep Research Takes on OpenAI & Gemini). It uses an open-source reasoning model (DeepSeek-R1) and an optimized workflow, making it very fast (in the test it was ~9x faster than ChatGPT DR) (Perplexity Unveils Deep Research: AI-Powered Tool for Advanced Analysis - InfoQ). Accuracy surprisingly high - ~21% in HLE benchmark, close to OpenAI (due to use of DeepSeek) (Perplexity Unveils Deep Research: AI-Powered Tool for Advanced Analysis - InfoQ). The output includes well-structured text, tables and citations of ~50 sources (Perplexity Unveils Deep Research: AI-Powered Tool for Advanced Analysis - InfoQ), it can be exported to PDF. Big plus: completely free in the basic version.	Slightly less depth of analysis - at the speed of ❤️ min it sometimes fails to track everything (Perplexity Deep Research Takes on OpenAI & Gemini) (Perplexity Deep Research Takes on OpenAI & Gemini). V Comparison with OpenAI DR sometimes misses some data or more detailed analysis when it is not easy to find (Perplexity Deep Research Takes on OpenAI & Gemini). Also the visual output is simpler - just static text and tables, no interactive graphs (Perplexity Deep Research Takes on OpenAI & Gemini). Restrictions: anonymous user has a limit of 5 queries per day (Perplexity Deep Research Takes on OpenAI & Gemini); for higher limits it is necessary to subscribe to Pro.	Free (5 queries/day for non-registered). $20/month Pro version offers up to 500 queries per day and faster responses (Perplexity Deep Research Takes on OpenAI & Gemini). Available globally via web browser, mobile app (iOS, Android) and macOS app (Perplexity Deep Research Takes on OpenAI & Gemini).

Tool (manufacturer)	Functions and features	Limitations	Price and availability
Google Gemini - Deep Research (Google)	DR agent integrated into the Gemini AI assistant (successor to BARD). When queried in *"Deep Research"* mode, it first generates a research plan to give to the user for approval/editing (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental). It then searches for a few minutes Google and gradually refines the analysis (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental). It uses Gemini's own LLM 1.5 (or 2.0 Flash) and Google Search (it has direct access to the index). A window context of up to 1 million tokens allows to accommodate really large documents (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental). The resulting report is exportable to Google Docs and contains links. The advantage is the strong integration with the Google ecosystem - it can leverage search knowledge and may soon build on Workspace (GDrive data, etc.).	Only in English and as part of the limited Gemini Advanced service (available in a few countries so far). Results in accuracy tests have lagged so far - *Gemini Thinking* (older version) gave only ~6.2% in HLE (Perplexity Deep Research Takes on OpenAI & Gemini). Research planning is fine, but it was noted that it can get stuck *in a static plan* (not looking beyond pre-approved points) (Perplexity Deep Research Takes on OpenAI & Gemini). The interactivity is less than in OpenAI - the user enters only at the beginning (plan approval), then the agent runs itself. The price is lower, but the performance and depth of analysis is also lower.	$20/month within *Gemini Advanced* plan (or part of the top-level Google One subscription). Launched in December 2024 (Perplexity Deep Research Takes on OpenAI & Gemini). Availability: web-based Gemini application (global for now, language EN), mobile Gemini app (coming soon), integration into Google Workspace is in the pipeline (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental).
Other (other)	Bing Chat (Microsoft): Microsoft's chatbot also offers a "*Research*" mode - it uses the GPT-4 model with direct Bing search. It can answer with citations, but the depth is less (more of a one-question answer with citations, not a long report). YouChat, Phind, etc.: similarly combine LLM with web; Phind specializes in developer queries with code. xAI Grok: Musk's model with web access (via X/Twitter), but not yet a robust general search tool.	Bing Chat: limited to sessions after a few queries, sometimes rejects overly long surveys. Other: varying quality of sources, sometimes cites only Wikipedia, etc. Grok: limited mainly to X data, not full site.	Bing: free for Edge/Bing users (limited number of queries per day). Phind/YouChat: free online. Grok: beta access for selected X Premium users (status 2024).

(Note: Accuracy data (HLE benchmark) and times are based on public tests and an announcement in February 2025 (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology) (Perplexity Deep Research Takes on OpenAI & Gemini). Accuracy rate means the ability to answer expert questions across 100 topics - HLE= Humanity's Last Exam, see above.)

From the above, it is clear that OpenAI Deep Research offers the deepest analysis (most resources, interactive process) but is exclusive and Expensive. Perplexity is a great compromise - it is surprisingly capable and available to everyone for free, albeit sometimes in less detail. Google Gemini DR is somewhere in between: systematic, but not yet as accurate or available to everyone. Competition is evolving rapidly, and we can expect more platforms to be added soon and to improve existing ones.

DR tool prompting best practices: for an AI agent to do a good job, we need to formulate the assignment correctly - a little differently than a regular LLM. A DR prompt is more of a *research question* or project assignment than a simple query. Here are some recommendations:

- Clearly define the aim of the research: Specify exactly what you want to find out. Instead of a vague *"Tell me about electric cars"*, write *"Find out the latest statistics on electric car sales in Europe over the last 5 years and identify the main factors that have influenced the increase or decrease in each year."* The more specific the questions you ask, the better the agent knows what to look for.
- Divide a complex question into sub-questions: you can list several points you want to cover in the prompt. E.g., *"... The report should include: (1) iOS vs Android adoption statistics for the last 5 years in the top 10 developed and 10 developing countries; (2) the percentage of the population that is learning a second language in these countries; (3) trends in mobile penetration; (4) recommendations on which markets to target for the new translation app."* This essentially dictates the outline to the agent. In tests, it has been shown that such a structured query leads to a very good answer - the agent covers all the points. (See below for a sample prompt of this type.) (Perplexity Deep Research Takes on OpenAI & Gemini)
- Specify the form of the output: as with the LLM, you can say what the output should look like. While DR agents have a default format (header, overview, chapters, resources), you can influence it a bit. E.g. *"... present the results in a summary table and include recommendations at the end."* Both Perplexity and OpenAI in the examples respond to the request to include tables or summaries (Perplexity Deep Research Takes on OpenAI & Gemini). If you want something like that, say so right away.

- Additional context (optional): unlike the reasoning model, you don't have to supply the DR agent with knowledge - it finds it on its own. But perhaps if you know of a particular resource that you don't want to leave out, you can mention it. E.g. *"... focus mainly on data from WHO and World Bank."* The agent will then prioritize these sources in the result, if they exist. Similarly, you can specify the *type of* sources: *"... use mainly academic studies."* (The tool will then search Google Scholar rather than popular sites.) Some DRs also have options in the UI for source preferences.

- Check and adjust the plan (if the tool allows it): before starting the search, Google Gemini will generate a plan - e.g. what sub-questions it will answer (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental). Take advantage of this and complete or correct it before you start the actual research. If you see that the agent has forgotten something, add it. You don't have this option for other agents, but you can make a similar outline the "plan" yourself in the prompts (see above).

- Patience during the run: Some agents (OpenAI) show the progress of the analysis - e.g. how many percent have been searched. Don't interfere, let them work. For OpenAI, they run for 10+ minutes, which is normal (Perplexity Deep Research Takes on OpenAI & Gemini). You should only interrupt if the research question has lost its meaning.

- Review the quotes and conclusions: once the output is generated, we recommend reviewing the sources the agent has listed, especially if you have important decisions to make on that. DR greatly reduces the risk of hallucinations, but you may still find that a detail is not accurate or the interpretation is questionable (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology).
  A quick check of key sources will give you confidence. (An expert at the University of Surrey warned that even if AI does do the analysis, humans need to take the time to verify - otherwise they may miss if AI has misjudged something (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology).

- Follow-up questions: as with the chat, you can ask follow-up questions after receiving the report. For example, *"Good, now find out the same for 2023 separately for the Czech Republic."* The agent can then follow up and complete without having to redo everything from scratch. Both OpenAI and Google support such follow-up queries - the model has the context of what it found stored internally, and responds faster to query extensions.

Sample prompt for Deep Research: (inspired by a real benchmark test)

> *"Help me find out the adoption rates of iOS and Android, the percentage of the population that wants to learn another language, and the change in mobile penetration over the last 5 years for the top 10 developed and 10 developing countries (by GDP). Sort the results into a clear table and add recommendations on which markets should be targeted for a new language learning app."*

This query has clear sub-items (OS adoption, interest in languages, mobile penetration) and defined groups of countries. An agent is therefore likely to:

. Find a list of the top 10 developed and developing countries by GDP.
. Searches for OS statistics (iOS vs Android market share) in these countries for the last 5 years.
Look for surveys or data on what percentage of people are learning a second language.
. Finds mobile penetration data (mobile users in the population).
. tabulate everything and conclude which countries are promising for a language learning app (e.g. high mobile penetration but low percentage of people learning the language=> growth potential).

In a real test on a similar query: Perplexity did it in ~3 min but some metrics were incomplete (e.g., missing complete data for all countries), Google Gemini solved it in ~6 min with solid tables and analysis, and the OpenAI DR agent spent ~11 min and delivered extremely detailed analysis including trends and graphs (Perplexity Deep Research Takes on OpenAI & Gemini). The open demo mentions that ChatGPT Deep Research turned this prompt into a 41-page report with 75 citations (Google Deep Research vs. OpenAI Deep Research: The Future of ...) - truly an analytical work. Such an illustration shows the difference in depth: Perplexity gave quick answers, OpenAI went much more in-depth (including visualizations and strategic recommendations) (Perplexity Deep Research Takes on OpenAI & Gemini) (Perplexity Deep Research Takes on OpenAI & Gemini). So the choice of tool depends on whether
you prefer speed or maximum detail.

Summary: *Deep Research* tools are moving AI capabilities from "answering knowledge questions" to conducting full-fledged research. They use LLM to analyze information from the web, eliminating the limitations of training data (they can work with the latest knowledge) and greatly reducing hallucinations (they verify all facts in the sources). We have seen in real examples DR can generate complex reports for both professional and practical purposes - from scientific research to business analysis to personal decision making. When using it, the key is to get it right
formulate the query as a research brief, or structure the requirements. The current main tools are OpenAI (the most thorough but expensive), Perplexity (accessible and fast) and Google Gemini (integration with the Google ecosystem). Across the board: AI will greatly speed up information gathering and summarization, but humans should always review and critically evaluate the final report, for major decisions.

---

4. Prompting Deep Research Tools

Now let's focus on how to properly *prompt* (query) Deep Research agents and how this differs from prompting reasoning models or regular LLMs. In the previous section we have already outlined best practices for DR - here we will summarize and compare.

Prompting reasoning vs. DR models - differences: With a reasoning model (without a web connection), you often need to supply the model with *all the information* to solve in the prompt because it can't see anything on its own (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models | Microsoft Community Hub). In contrast, the DR agent actively procures the information - you just tell it *what to find out*, you don't have to write out the data in the prompt. this However, meansthat the query for DR can be more general (e.g., "Find out the latest developments in XYZ field"), whereas a reasoning model would only answer such a query from memory (and might hallucinate or not know the new data). So for DR, don't be afraid to ask very broad questions that require searching many sources - that's what the agent is there for.

The other difference is that the reasoning model works directly in the conversation - you can guide it step by step (e.g. give it data to analyze). In contrast, a DR agent works more in *a batch fashion* - you give it a complex task and it will "pause" for a while and then deliver the result. So a DR tool prompt is often longer and more complex than a typical chat prompt. You can include multiple sub-questions, output requests, etc. because you are not expecting an immediate one-sentence response, but a structured message.

For the reasoning model, we cautioned against using *step-by-step* instructions, because the model does this internally. Similarly, for DR agents, you don't (and shouldn't) explicitly describe *how* to search - e.g., don't say *"search this first, then that"*. These agents have their own search heuristics, and your intervention could rather limit their effectiveness. Instead, articulate *what to find out*. The exception is Google Gemini, where you can modify the suggested plan - but that's part of the UX, not the prompts per se. So in the DR prompt, you're describing the goal and content of the output, not the process (leave that to the agent).

Key principles of prompting DR tools:

Research question instead of a direct question: formulate the input more like a term paper or report assignment. E.g., *"Analyze the impact of technology X on industry Y over the past decade, including major milestones and the current state of research."* That way the agent knows to look for historical developments, key studies, current reviews, etc. If you gave the same thing to a traditional LLM, they either wouldn't know or would be hallucinating - a DR agent would will find out.

. Do not search manually in advance: Some users have a tendency to stuff links and quotes they found into the prompts. This is not necessary and may be rather detrimental (the agent would then favour those links and possibly ignore others). It is better to indicate the *types of sources* ("... use official WHO statistics..."). The exception is if you have a specific document that *you are sure* should be included (for example, an internal PDF report) - you can upload this (with OpenAI DR) or give a URL (some agents can read from URLs). But otherwise: trust the agent to do the searching, they are more efficient at it.

. Check and iterate through follow-up instead of one prompt: If the result is not quite as expected, it is often better to ask a clarifying *follow-up* question than to re-formulate the entire prompt immediately. The agent already has a lot of data from the first run. For example, if Perplexity misses something, just ask: *"You forgot to mention XYZ, please add it."* and it will quickly find the missing part. In the prompts, next time you would mention it from the beginning. So try to make the initial prompt as good as possible, but don't be afraid to interactively polish the result with another query - it's faster than rewording everything.

. Specifics of different tools - background models: It's useful to know *what LLM a given DR tool drives*, as this will affect style and capability:

- OpenAI Deep Research uses a variant of GPT-4 (o3) with an emphasis on reasoning (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology). This means that the answers are very detailed, vivid, and almost always correctly reasoned. The style is rather formal and technical. It also means that it does not need any prompting to 'think' - the prompt can be more concise on the solution instructions, as the GPT o3 chooses its own course of action. It can handle more general queries with bravado, but expect long waits.
- Perplexity Deep Research runs on an open-source model (DeepSeek-R1) (Perplexity Unveils Deep Research: AI-Powered Tool for Advanced Analysis - InfoQ). This has high performance in logic tasks, but is not as fine-tuned for output stylistics as GPT-4. Answers can be more concise and sometimes less "polished". Therefore, you can put more emphasis on form in the prompt (e.g. *"make it understandable even for laymen"* or *"avoid too much technical detail"*) if necessary. DeepSeek is very fast, so that Perplexity can afford short iterations of the search - but write the prompt complete anyway, don't expect the agent *to ask*. (Perplexity, unlike GPT-4 DR, doesn't ask the user anything at runtime.)
- Google Gemini DR uses the Gemini 1.5/2.0 model. It does not yet have the logical depth of o1 or DeepSeek, so it is useful to formulate the prompt unambiguously and provide more context. What is a given with OpenAI (that it finds relevant data) may require a nudge with Gemini. For example, when querying medical research, the prompt could state *"... focus on peer-reviewed studies and meta-analyses from the last 5 years."* This will direct Gemini to better sources (otherwise he might take even popularization sites, where there is a risk of lower quality). We can use the knowledge that Gemini has 1M context tokens: feel free to put more text into the prompts that you already have, for analysis - it can handle it.

Tone and language: most DR agents currently work best in English. If you ask a query in another language, it may be that the agent will still search primarily for English resources and then translate the answer (as Bing does). The quality of the translation and the resources found may not be ideal. In general, for best results, it is recommended to write the prompt in English if possible, as this will allow you to use the full potential of the tool. (Of course, you can then have the final message translated, for example by ChatGPT itself.) This is the difference with reasoning models like o1 - they are trained multilingual and can handle English decently. But for DR, the primary domain is English, since most online content and models (Bing, Google) are optimized for EN.

Overview of the models behind the DR tools: as mentioned, OpenAI vs. Perplexity vs. Gemini use different LLMs, which is reflected:

- OpenAI DR (GPT-4 o3) - the most advanced model that spends more time thinking and searching than it answers (Introducing OpenAI o1| OpenAI) (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology). This is evident in the quality - it scored 26.6% the HLE, which is *an order of magnitude better* than the classic GPT-4o (3.3%) (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology). In practice, this means that where the regular ChatGPT flounders, the DR agent succeeds. This strength is by on givenRL's training on chain-of-thought and perhaps a larger "computational budget" for inference (OpenAI has suggested that o1/o3 have the ability to spend more computational steps searching for an answer than GPT-4o) (o1: A Technical Primer - LessWrong) (OpenAI's o1 is a misunderstood model - by Charlie Guo). For prompting, this means: you can afford to be ambitious - specify a very complex task - and the model it will probably handle in detail. Just remember the limit of 100 queries per month, so don't waste it on trivia.

- Perplexity DR (DeepSeek) - on the open-source DeepSeek-R1 (Perplexity Unveils Deep Research: AI-Powered Tool for Advanced Analysis - InfoQ). The latter is the result of distillation and RL training by the community and achieves performance comparable to GPT-4 in many benchmarks (DeepSeek)(DeepSeek). However, it does not have as broad a knowledge and feel for the language finesse. In practice, Perplexity handles this by running the model rather "briefly" to get through everything in 3 min and focusing on the facts. Prompt should be very specific with Perplexity about what we want - because once you explicitly mention something, the agent will include it. Conversely, what you don't mention, he may leave out, unless he considers it crucial himself. So it is useful to *list what the report should include* (as in the sample prompt above). This will minimize the chance that he will leave something essential out because of the time limit.

- Gemini DR (Gemini LLM) - Gemini is different in that it makes heavy use of Google search technology and an agent-based scheduling system (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental) (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental). Thus, the model relies on Google to find relevant things well for it. The prompt should therefore contain *keywords* (😃) much like you would typed into Google. For example, the query *"How has CRISPR-Cas9 evolved in gene therapy research between 2018 and 2024, and what are the major breakthroughs?"* - The bolded words are what I'm sure to include. If I asked the same question loosely *"Summarize gene therapy advances recently"*, he might be looking for all sorts of things. So for Gemini, pick out the terms he should be looking for. Otherwise, the Gemini model continues to improve - the version 2.0 (experimental) has significantly higher capacity and speed (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental) (Gemini: Try Deep Research and Gemini 2.0 Flash Experimental), so it may soon compete with OpenAI. For now, however, it is advisable to formulate the prompt in more detail, leaving nothing to the intelligence of the model (which is slightly weaker in reasoning than OpenAI).

In summary, prompting DR agents is about *carefully defining the goal and content of the report,* while leaving the actual search process to the AI. The differences between the tools are nuanced: OpenAI can handle a lot even from a sketchy assignment (but you'd better take the opportunity to describe the requirements thoroughly to use its full potential), Perplexity needs clear instructions on what to cover (so that nothing important is left out), and Google welcomes being prompted "serve" relevant terms for search. In all cases, a well-structured query leads to a better-structured answer. Conversely, a vague query (e.g. *"Do my research on medicine"*) can lead to a superficial result from even the best AI.

Recap: Prompting Deep Research tools involves asking a complex question that clearly defines *what the output should contain.* Unlike a traditional chatbot prompt, we can (and should) include more details and requirements, perhaps in the form of bullet points. The DR agent doesn't need us to tell it *how* to proceed - that's its job; but it does need to know *what we want to find out from it and in what form.* Thus, a quality DR prompt looks more like a task assignment for a human analyst. Different tools (OpenAI, Perplexity, Gemini) have different models behind them, but the bottom line is that the user should give good input instructions to all of them and then let the AI do the work. The final output should be verified and possibly refined or supplemented with further queries. By following these principles, DR agents can produce impressively high-quality searches in record time.

---

5. Summary

Key facts from the material:

- Reasoning AI models represent a new generation of LLMs that have the ability to internally reason in multiple steps. They differ(ChatGPT-4, etc.) in that they do not form an answer in one jump, but break the problem down into sub-reasonings and solve them sequentially (A Visual Guide to Reasoning LLMs - by Maarten Grootendorst). This allows them to perform significantly better complex problems (e.g., complex from classical models mathematical and logical problems) (Introducing OpenAI o1| OpenAI). OpenAI has introduced the o1 (2024) series trained by RL, outperforms GPT-4 by an order of magnitude in tests. Google responds with the Gemini model with agent-based scheduling. There are also open-source variants like DeepSeek.

- Prompting reasoning models requires a slightly different approach than for classical LLMs. The basic rules (clarity, context, desired format) apply, there is no need to impose a chain-of-thought on the model - reasoning models generate the chain-of-thought themselves (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). Conversely, it is important to supply any information that the model may not know (actual or specific data) because it does not have access to external knowledge (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). Best practice is to formulate the query directly and let the model workout the solution procedure. When prompting o1/o3, avoid unnecessary examples or step-by-step instructions; rather, use the option to set the role and desired output. The Reasoning model will typically handle a complex problem on the first try - if not, iterate through the prompt or query to verify.

- Deep Research (DR) tools integrate LLM with web search and allow AI to autonomously perform online searches. They create a series of web queries from your query, read dozens of pages, extract relevant information, and write a comprehensive report with citations (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained) (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained). This overcomes the limitations of conventional LLMs - they can work with current information and verify facts across sources, drastically reducing hallucinations. They are ideal for complex issues one would otherwise have to search for a long time (trend analysis, product comparisons, expert reports). The main proponents are ChatGPT Deep Research (OpenAI), Perplexity AI, and Google Gemini Deep Research. They differ in depth, speed and accessibility: OpenAI is the most thorough (and the most expensive), Perplexity very fast and available for free, Google connected to the search engine but less accurate so far.

- Prompting DR tools is similar to setting a task for an analyst. You need to clearly define *what the research is to find out*, and ideally Structure the query into several specific requirements (sub-questions). Unlike the chatbot, we do not conduct a step-by-step dialogue here (the agent controls the procedure itself), so we put all instructions into one prompt. We don't have to supply facts (the agent will find them), but we can specify preferred sources or form of output. The important thing is to *be specific* - the agent will then cover all these points. After the report is generated, follow-up questions can be asked to complete it. Overall: the better you define the specification, the better and more useful the output of the DR agent will be.

The most important information to remember:

- *Reasoning LLMs vs standard LLMs:* Reasoning models (o1, Gemini, etc.) "think" before answering and thus handle much harder tasks with less error (Introducing OpenAI o1| OpenAI). However, they are slower and less suitable for simple queries - there the classic LLM is sufficient.

- *Prompting reasoning models:* don't make them think in steps - they do it themselves (Prompt Engineering for OpenAI's O1 and O3-mini Reasoning Models| Microsoft Community Hub). Focus on accurately stating the problem and providing relevant information. Indicate what type of answer you want (style, format). Example: instead of *"Calculate a difficult example"*, give a specific example and a request to *"explain the solution*" - the reasoning model will then safely arrive at the correct result and describe the procedure.

- *Deep Research agents:* they are your AI research assistants - they can go through hundreds of web resources for you and write an informed report with references (Deep Research in AI: Advanced Methodologies & Breakthrough Applications Explained). Compared to a regular search, they save a huge amount of time and can integrate insights from different places. The outputs are in the form of reports (often several pages of text). Credibility is higher than traditional LLMs because they cite and check everything (New OpenAI 'Deep Research' Agent Turns ChatGPT into a Research Analyst -- Campus Technology), but it is still advisable to verify the results.

- *Prompting DR:* Write a detailed specification of what you want to find out - preferably in bullet points. Don't be afraid to ask for multiple things at once, the agent will cover them in sequence. Specify the time period, geographic scope, type of data you're after (e.g. *"statistics", "comparisons", "causes and effects"*).
  You can mention that you want tables or charts. The more specific and structured the query, the better the structure of the answer.

- *Validation and responsible use:* while AI can make information retrieval tremendously faster, it is not infallible. It is always up to the user to critically evaluate the results. With reasoning models and DR reports, important decisions should not be based blindly on AI alone. exit without checking. Use quotes - DR gives them to you as clues, so use them and see if the AI interpretation matches the source. When used correctly, however, these tools are extremely powerful and can free developers, researchers and other professionals from routine work and allow them to focus on higher problems.

For further study: For those interested in a visual demonstration, we recommend the video "Introduction to Deep Research" by OpenAI (Mark Chen et al., Tokyo, 2023), which shows how the DR agent proceeds and what results it generates (Introduction to Deep Research| IBL News .es). Also, a comparative article *Perplexity vs OpenAI vs Google Deep Research* (Perplexity Deep Research Takes on OpenAI & Gemini) with practical examples of prompts and outputs. These resources provide deeper insights and concrete examples to help you learn how to work with reasoning models and DR tools in practice.