The Prompt Engineering Report Distilled: Quick Start Guide for Life Sciences

Valentin Romanov[1#], Steven A Niederer[1,2]

1. National Heart & Lung Institute, Faculty of Medicine, Imperial College London, United Kingdom
2. Turing Research and Innovation Cluster: Digital Twins, The Alan Turing Institute, London, United Kingdom

 # Corresponding author: v.romanov@imperial.ac.uk

Key words: Large Language Models, Prompt Engineering, life sciences, zero-shot, few-shot, hallucinations

Abstract:

Developing effective prompts demands significant cognitive investment to generate reliable, high-quality responses from Large Language Models (LLMs). By deploying case-specific prompt engineering techniques that streamline frequently performed life sciences workflows, researchers could achieve substantial efficiency gains that far exceed the initial time investment required to master these techniques. The Prompt Report published in 2025 outlined 58 different text-based prompt engineering techniques, highlighting the numerous ways prompts could be constructed. To provide actionable guidelines and reduce the friction of navigating these various approaches, we distil this report to focus on 6 core techniques: zero-shot, few-shot approaches, thought generation, ensembling, self-criticism, and decomposition. We breakdown the significance of each approach and ground it in use cases relevant to life sciences, from literature summarization and data extraction to editorial tasks. We provide detailed recommendations for how prompts should and shouldn't be structured, addressing common pitfalls including multi-turn conversation degradation, hallucinations, and distinctions between reasoning and non-reasoning models. We examine context window limitations, agentic tools like Claude Code, while analyzing the effectiveness of Deep Research tools across OpenAI, Google, Anthropic and Perplexity platforms, discussing current limitations. We demonstrate how prompt engineering can augment rather than replace existing established individual practices around data processing and document editing. Our aim is to provide actionable guidance on core prompt engineering principles, and to facilitate the transition from opportunistic prompting to an effective, low-friction systematic practice that contributes to higher quality research.

1 Introduction

If the latest flagship models like ChatGPT-5 Pro and Claude Opus 4.1 work "out of the box" then why do Large Language Model (LLM) companies provide highly detailed and specific instructions ('GPT-5 Prompting Guide' 2025; Anthropic 2025a) explaining how to prompt their models? The answer to this question is the motivation behind writing this review. The simple answer to the question is that for most basic tasks, these models perform exceptionally well. However, for anything academic or professional, whether finding specific references, editing, coding, data extraction, model fitting and other tasks, a prompt such as "*improve this paragraph to sound more professional*" is not only ill-defined, lacking specificity, its also vague; Improve how? What does professional mean? What does 'to sound more' mean in this context? Are there gold standard examples of what professional looks like?

The development of systematic questions and to improve LLMs answering is the essence of prompt engineering, which is the intentional curation and development of a prompt by imbuing it with various conditionals (Schulhoff et al. 2025). By definition, this must mean prompt engineering requires a certain level of cognitive load, reflection and understanding of the problem it is being engineered towards. Most non-AI experts approach prompting opportunistically rather than systematically (Ye et al. 2024). That being said, designing deliberate and effective prompts is hard even for LLM and domain experts (Zamfirescu-Pereira et al. 2023) and is an ongoing area of research in Natural Language Processing (Bommasani et al. 2022) with companies like Anthropic providing an automated generator(Anthropic 2024a) to help reduce friction.

And yet, because an answer to a hard question appears to only be a prompt away, LLM adoption within sciences has skyrocketed. Of the 2534 responses provided to a Danish survey on LLM usage in academia, almost 50% reported using LLMs for "editing a research article for improving readability" and "refining language for research proposals", with ~ 35% using them for writing and editing code, summarizing and analysing literature, or proposing a title, abstract or for generating keywords (Andersen et al. 2025). These usage proportions are higher than they were in 2023 (Van Noorden and Perkel 2023), and as model capabilities improve, it's hard not to envision that more and more work will be offloaded to LLMs in the near future. Within the life sciences, LLMs have been used within the field of cardiovascular disease (Mishra et al. 2024; Sharma et al. 2024), protein-protein interactions (M. Jin et al. 2024; Chang et al. 2025), extracting materials data from research papers (Polak and Morgan 2024), and assisting with chemical synthesis predictions (Zheng, Zhang, Borgs, et al. 2023; Ruan et al. 2024). LLM applications extend to enhancing mathematical reasoning through topic-aware prompt engineering and self-evaluation (Sukherman and Folajimi 2025; Imani et al. 2023), though recent work has shown that chain-of-thought approaches can sometimes reduce performance on certain mathematical tasks (Evstafev 2025). LLMs are being integrated into digital twins (Amad et al. 2025; Sun et al. 2024), for creating gene signalling networks (Tewari et al. 2025) and in drug discovery and development (Y. Zheng et al. 2024).

In this review, we distil the latest advances in prompt engineering techniques for life sciences applications. We provide a detailed examination of core prompting techniques

as described within the Prompt Report (Schulhoff et al. 2025), focusing on six categories: zero-shot, few-shot approaches, thought generation, ensembling, self-criticism, and decomposition. Where possible, we provide case studies to illustrate these concepts. We also review published prompts and suggest improvements based on current best practices. Further, we examine studies that have deployed LLMs for tasks ranging from literature summarization and data extraction to protein interaction prediction and synthesis planning. We focus on core prompt engineering principles, building complexity incrementally while introducing relevant nuance, and providing concrete recommendations where possible. By highlighting common pitfalls such as multi-turn conversation degradation, hallucination risks, and the differences between reasoning and non-reasoning models, we aim to guide researchers in selecting and implementing prompting strategies appropriate for their experimental and professional objectives.

We also address emerging frameworks including agentic tools and Deep Research, emphasizing their current limitations alongside potential applications. While predicting the trajectory of prompt engineering remains challenging given rapidly evolving model capabilities, we document current best practices that have remained consistently relevant since the introduction of ChatGPT-3.5 in 2022. We propose that systematically engineered prompts, when properly implemented, may reduce the burden of various research and non-research activities, from literature review and data analysis to providing new perspectives on established workflows in drug discovery, systems biology, and precision medicine. Importantly, prompt engineering augments rather than replaces existing pipelines, offering researchers additional tools for interacting with familiar data and processes. Our aim is to provide actionable guidance on core prompt engineering principles, contributing to the transition from opportunistic prompting to an effective, low friction systematic practice.

2 Prompting and Prompt Engineering

Our discussion on prompting will focus specifically on what it means to prompt and what it means to prompt engineer. Programmatically automating and developing strategies for iterating on and improving prompts can be complex and is not how most academics engage with LLMs and as such, are not covered here. Creating thoughtful, fully specified prompts can take significant time and can be frustrating. The goal of this review is to provide comprehensive demonstrations of how to engineer prompts, with as little friction as possible, to allow academics to move quickly and confidently.

Core prompting techniques can be broken down into; 1) Text based Techniques, 2) Multilingual Techniques and 3) Multimodal Techniques (audio, video etc) (Schulhoff et al. 2025). The core focus of this review will be on text-based techniques. We will mention considerations for getting reliable results from multilingual prompting and plan to release a separate report on multimodal techniques in the future. In the Prompt Report, the authors identified 58 different prompting techniques. These techniques can be broken into 6 major categories, covering the various unique ways to approach constructing prompts. In this section, we will focus on these 6 categories and provide a flavour of each prompting technique with research specific use cases.

2.1 Zero-shot

There are several ways to think about prompting engineering. As a thought experiment, we can directly map two different but common approaches to LLM usage patterns. Exploratory searches, such as reading an initial PubMed abstract to understand a topic, align with zero-shot prompting where one can enquire and rapidly obtain high-level understanding of the concept. On the other hand, using structured, well-specified prompt engineering approaches like few-shot, ensembling and decomposition require significantly more effort to engineer and are more akin to a systematic review.

Zero-shot refers to the case where the user simply provides a prompt to the LLM without defining what the output should look like. In this case, the assumption is that the model will have seen similar data in its dataset during pre-training. The best way to structure a zero-shot prompt is to provide the request or question together with rules (Wei, Bosma, et al. 2022) or specifics that it can and can't do. While there are many zero-shot strategies one can deploy, we refer the user back to the original Prompt Report. Here we will discuss the most frequently used techniques deployed within academic work where LLMs are utilized for summarizing articles, extracting information, or editing manuscripts.

Not all prompts are equal, especially within academia. We provide 2 examples of fairly simple prompts that can have outsized negative consequences if not mitigated for. In Figure 1 the user is asking an LLM a question, to learn something about the topic. Learning from LLMs while effective in some domains, is less effective within domains with little training data, for example, a PhD student trying to learn a nuanced concept published earlier on in the year. LLMs are unreliable within work environments that push the frontier of knowledge.

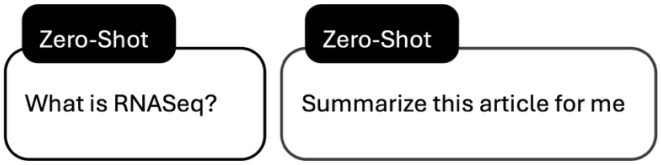

Figure 1. Examples of Zero-Shot prompting. This is how most academics use chatbots like ChatGPT. While effective, these 2 prompts touch on 2 key points of failure when using chatbots, 1) extracting knowledge from an LLM, and 2) summarizing dense, nuanced information from academic sources.

Generating article summaries is a popular demonstration of zero-shot prompting and significantly useful for academic work (Figure 1). Well represented summaries can help researchers narrow down the potential list of articles that need to be read to gain the necessary coverage of a topic and to reduce fatigue. In a recent study, Hanson et al. (2023) showed that total articles indexed in Scopus and Web of Science have grown exponentially in recent years; in 2022 the article total was approximately ~47% higher than in 2016, which has outpaced the limited growth in the number of practising scientists (Hanson et al. 2024). As such, LLM summaries that capture the coverage, context and nuance of research articles can be extremely useful.

Can we trust LLM summaries of academic research? In short, current research does not support this. While LLMs have been extensively used for summarizing research articles, current research suggests that these summaries lack nuance. Recently, Peters and Chin-Yee (2025) tested a number of frontier LLMs (Sonnet 3.7, ChatGPT -4o, DeepSeek, amongst other models) in their ability to summarize articles across multidisciplinary sciences and medical journals, finding that LLM generated summaries were 5 times more likely to contain overgeneralizations compared to human-authored summaries (Peters and Chin-Yee 2025). Similarly, LLM generated abstracts, while creating abstracts that were more readable than the original, lacked quality as measured by the CONSTOR-A checklist (Hwang et al. 2024).

A goal of this review is to bring together the latest ideas on prompt engineering and provide recommendations where possible to enhance past work. In Figure 2, we have taken the original prompts given by the authors and included suggestions that may improve the adherence, specificity and output of the LLMs. Our analysis reveals several areas for improvement across all prompt types. First, the “simple prompt” lacks field specificity and examples. Providing the LLM with multiple examples of high-quality human-generated summaries in the relevant field, along with examples of poor summaries, helps establish guide LLM output. Second, while the systematic prompt introduces step-by-step reasoning, it would benefit from concrete examples of effective thought processes for information retrieval, allowing the model to better understand what is expected. Third, the accuracy prompt's use of negative instructions ("do not introduce any inaccuracies") can be an issue as LLMs cannot reliably self-assess factual accuracy. Instead, specific guidelines about what to include or how to verify information would be more effective. Finally, the full text prompt highlights critical technical constraints that require careful navigation. Processing entire research articles demands careful monitoring of token consumption; one effective strategy is to provide the LLM with one well-specified prompt and a single article per conversation to maintain focus

and accuracy. A constant theme throughout this review is that creating well-specified prompts is hard, requiring not just clear instructions but also concrete examples, field-specific context, and awareness of technical limitations.

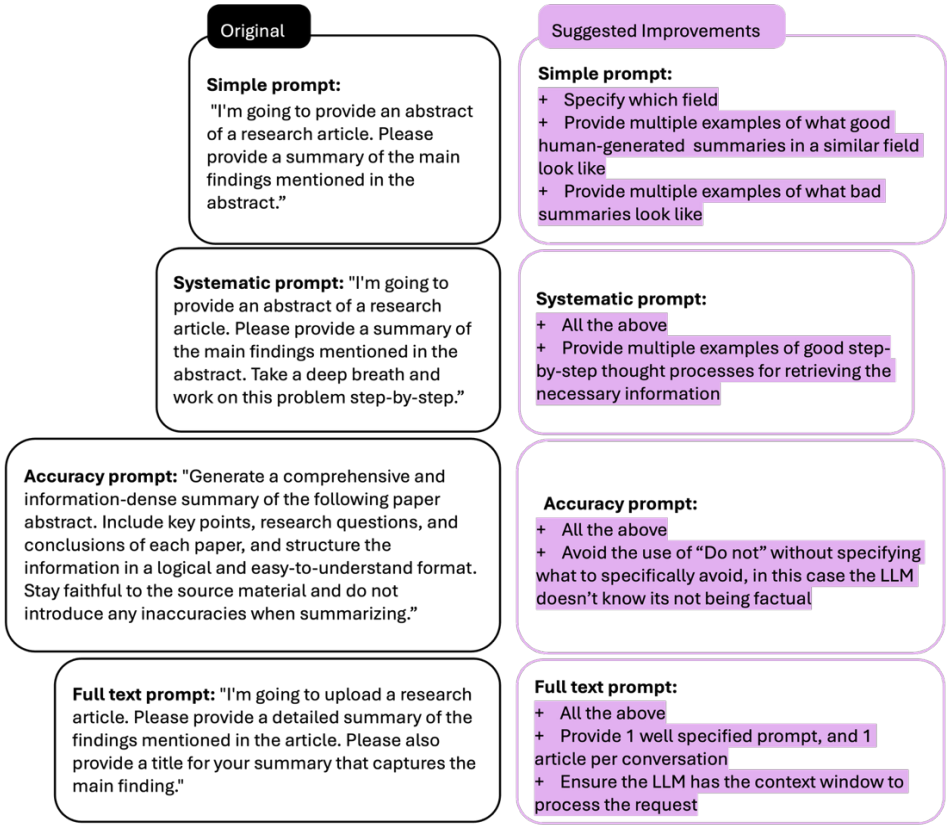

Figure 2. A zero-shot prompt case study. Original prompts from Peters and Chin-Yee (2025) (Peters and Chin-Yee 2025) are shown on the left, with suggested improvements on the right. Each improvement (+) represents a specific enhancement that may improve outcomes: adding domain specificity, incorporating quality exemplars, clarifying instructions to avoid ambiguity, and addressing technical constraints.

2.2 On Context Window and Token Consumption

The context window of an LLM refers to the number of tokens an LLM can attend to (remember) at once. When passing a text phrase into an LLM, the text is encoded into a sequence of token ids, mapping to substrings (Rajaraman et al. 2025). One token roughly represents 0.5 words (based on Gemini tokenization ratio via AI Studio). Exceeding the length of the context window typically results in hallucinations, and sever performance degradation(An et al. 2024). Further, LLMs are typically not able to utilize the entire context window, remembering information in the beginning and at the end of the context window, but overlooking information in the middle (H. Jin et al. 2024). Based on publically available information, the free version of ChatGPT has a context window of 8k tokens, Gemini has a context window of 32k tokens while Claude has 200k tokens. A typical research article is roughly 4k tokens. In Figure 3 we illustrate the relative difference in context size between the free offerings for 3 different LLM providers, where the sphere volumes are proportional to their respective context windows. This visualization

demonstrates that while ChatGPT's free tier can accommodate approximately 2 research articles, Gemini can process 8 articles, and Claude's substantially larger context window can handle approximately 50 articles simultaneously. This dramatic difference in capacity has deep implications for conducting reliable, cohesive, and comprehensive literature reviews, while working with multiple documents and document types.

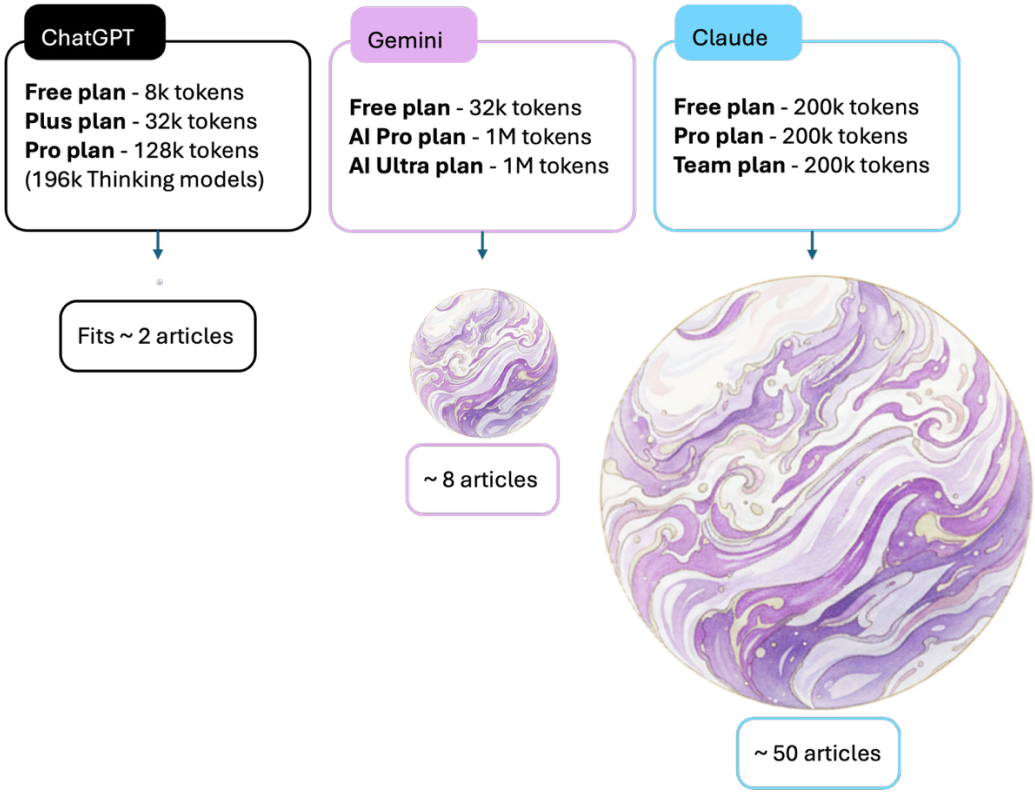

Figure 3. Comparative visualization of context window capacities across free-tier LLM offerings. Token limits for ChatGPT (8k tokens), Gemini (32k tokens), and Claude (200k tokens) are shown with their respective subscription tiers, alongside a volumetric representation where sphere sizes are proportional to context window capacity. The visualization translates these token limits into practical research contexts, demonstrating that the free versions can accommodate approximately 2 articles (ChatGPT), 8 articles (Gemini), and 50 articles (Claude) based on an average research article length of 4k tokens. Claude's free tier offers 25x the capacity of ChatGPT and 6.25× that of Gemini.

2.3 Personas

Assigning a persona to an LLM is one of the most popular use cases of this technology, for example, “Take of the role of a clinician” or “Take on the role of a literature review specialist”. Personality assignment is used frequently within and outside the realm of academia, Character.ai, a website hosting thousands of AI personalities has around 20+ million monthly visitors. The most prominent use of personality assignment can be found in system prompts provide to LLMs by all major provides. System prompts describe how the chatbot will behave and interact with the user. For example, the system prompt for Claude Opus 4.1 from Anthropic outlines how the model should help the user with

improving prompts, how to handle difficult people, even how to provide emotional support(Anthropic 2024b).

Personas represent abstractions that emerge from statistical patterns in the training data. These abstractions often amplify latent biases and stereotypes present in the source material. Figure 4 illustrates this phenomenon through Claude Opus 4.1's interpretation of a "clinician" persona. When presented with an identical prompt across three independent conversations, the model generated responses that, while thematically consistent (empathetic, knowledgeable, objective, calm, clear), exhibited notable variation in the specific attributes emphasized, ranging from "trustworthy" and "analytical" in Response 1 to "evidence-based" and "inquisitive" in Response 2. The prompt lacks specificity such as clinical specialty, practice setting, geographic location, or level of seniority. The model's responses thus reflect not a singular, well-defined professional identity, but rather a composite representation drawn from diverse clinical archetypes within its training data.

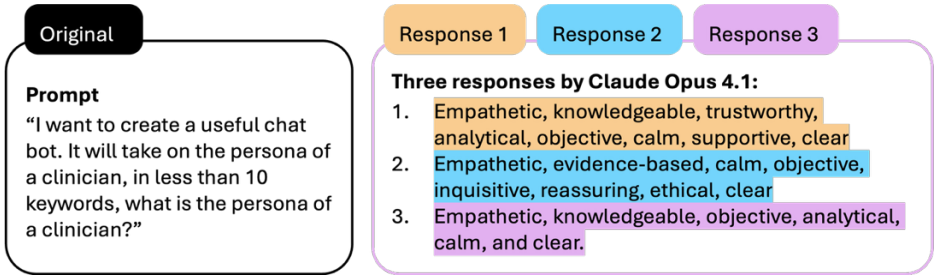

Figure 4. Variability in LLM persona interpretation across independent conversations. The same prompt requesting Claude Opus 4.1 to describe "the persona of a clinician" in under 10 keywords was submitted in three separate conversation threads. Despite the identical input, the model produced three distinct sets of attributes, demonstrating both semantic overlap (e.g., "empathetic," "calm," "objective" appearing across responses) and notable divergence in emphasized characteristics (e.g., "trustworthy" vs. "evidence-based" vs. "analytical"). Underspecified persona prompts can yield inconsistent and unreliable outputs.

Figure 5 showcases two role-playing prompts for chemistry applications. The first assigns the LLM the persona of Bohr, a literature review specialist within a seven-member AI team optimizing aluminium-based metal-organic frameworks crystallinity (Zheng, Zhang, Nguyen, et al. 2023). Each team member, from Project Manager Atlas to the Robot Technician, has distinct responsibilities, enabling specialized task distribution. The second prompt (Figure 4b) takes a simpler approach where the LLM serves as a chemistry assistant extracting synthesis parameters into JSON format (Silva et al. 2024).

The two prompting strategies along with personality assignment are different. Multi-agent systems like Bohr's team excel when problems require diverse expertise and iterative refinement, such as optimizing synthesis conditions through Bayesian optimization. In contrast, the standalone assistant with explicit output examples proves more effective for standardized data extraction from literature. Both approaches include examples of expected outputs. Examples help to reduce ambiguity and improve output consistency.

Throughout this review, we employ specific terminology with precise operational definitions. Large Language Models (LLMs) refer to the transformer-based neural architectures trained on extensive text corpora (GPT-4, Claude, Gemini). Agents or agentic systems denote LLMs augmented with tool-use capabilities, memory systems, and autonomous task execution frameworks such as Deep Research (OpenAI), Claude Code (Anthropic), and similar implementations.

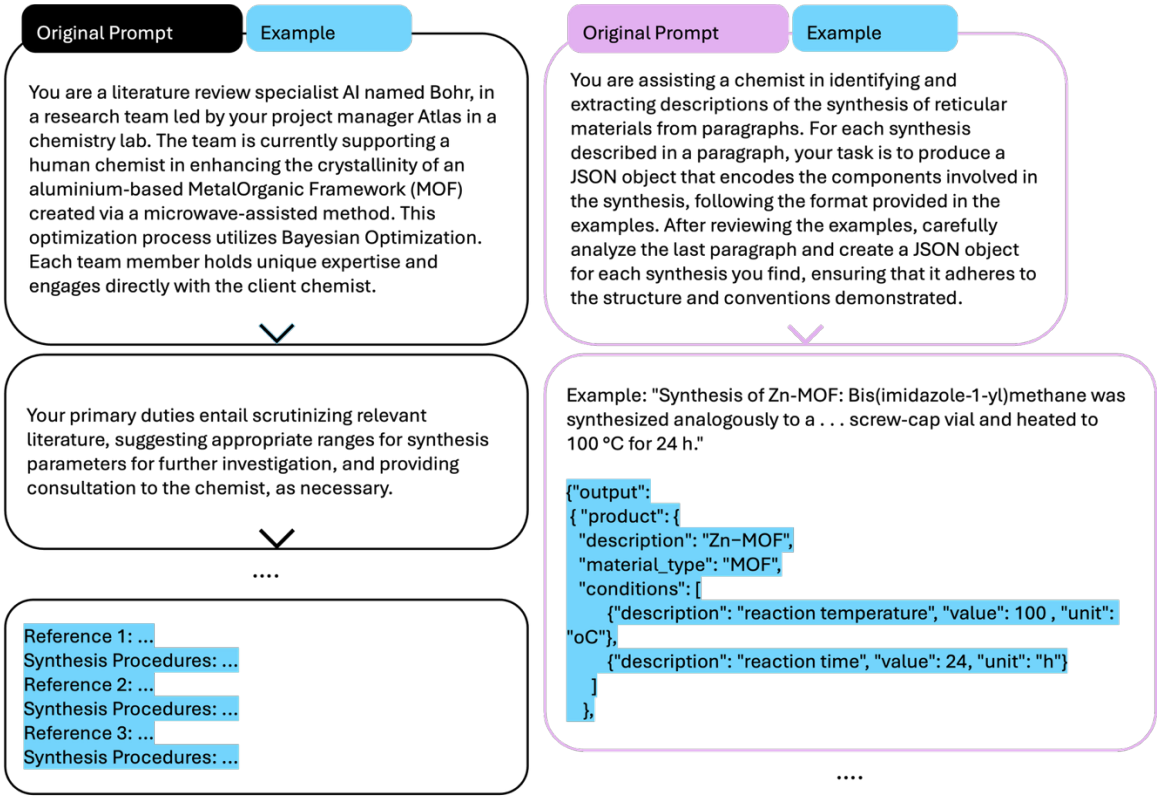

Figure 5. Examples of a personality and an assistant prompt from literature. Left: An example of a literature review specialist named Bohr, as part of a larger team of AI agents, each with their own persona, including Project Manager, Analytical Assistant, Chemical Synthesis Consultant, Modeling and Coding Specialist, Robot Technician, Lab Equipment Designer, and the Data Analysis Assistant Specialist (Zheng, Zhang, Nguyen, et al. 2023). Right: A chemistry assistant prompt for extracting information regarding the synthesis of reticular materials from information provided by the user. The assistant prompt has two parts, a description of the task and an example of the output (Silva et al. 2024). Left: Adapted with permission under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

Is there a difference between asking the LLM to take on a persona vs acting as an assistant? The scientific literature is yet to settle on this point. Evaluations are made harder because some articles refer to personas as assistants. The current state of the field suggests there may be some advantages to using expert personas that work within their domain, however the effect is small and performance is typically unreliable (Araujo and Roth 2025). While its suggested that there is room for sculpting distinct AI personalities, as it stands, assigning personas to an LLM on objectives tasks such as mathematics and physics, typically results in unpredictable behaviour (M. Zheng et al. 2024). Li et al. (2025) suggested the need for “rigorous science” around persona generation finding that personas representing the general population for predicting voting patterns were shown to be significantly biased (A. Li et al. 2025).

One interesting emerging application of personas is in population and disease modelling. Although LLMs like Gemini 2.5 Pro, Claude Opus and ChatGPT-4o can generate informative prior distributions for explaining cardiovascular health outcomes, the reliability of this approach is inconclusive. Although these models can generally identify the correct directional associations (e.g., whether a factor increases or decreases disease risk), they struggle with calibrating the magnitude and confidence intervals of these associations (Riegler et al. 2025). This contrasts with Capstick et al. (2025), who successfully leveraged LLMs to generate prior distributions that significantly accelerated predictive model development, reducing the required manual labelling time by more than six months (Capstick et al. 2025).

2.4 Stylistic Guidance

The LLM can be guided via prompts to edit sentences, change tone, and stylistic direction of the text (Lu et al. 2023). LLM adoption within the scientific community for helping write, and translate academic text has been astronomical (Van Veen et al. 2024). Using LLMs "to help write research articles" was the 4th most popular use case, accounting for ~28% of use by researchers, as reported in a Nature survey (Van Noorden and Perkel 2023). An analysis of biomedical research literature discovered that around13.5% of abstracts (~200,000 papers) published in PubMed in 2024 used an LLM. Usage varied greatly (lower bound between 5% to more than 40%) between research fields, affiliated countries and journals (Kobak et al. 2025). In 2025, the majority of researchers seem to employ LLMs primarily as editing tools rather than for complete article generation, which may enhance the clarity and accessibility of their research (DeHaan et al. 2025).

Whether using LLMs for these tasks is good for science in general is unclear. From biases (Bender et al. 2021), to fabricating references (Walters and Wilder 2023), there are significant limitations that the user should be aware of. However, as model training scales and new algorithmic methods are added to the pre-training and post-training processes, LLMs are becoming substantially more capable and reliable. While LLMs can still hallucinate references using state-of-the-art models, like Gemini 2.5 Pro and Claude Opus 4.1, it is becoming rarer, and the quality of references that these models suggest, has significantly improved.

While adoption is increasing, there is little guidance or best practices on how to utilize LLMs for editing scientific writing. We provide a structured framework below for leveraging LLMs as editing tools, demonstrating how specific prompt engineering techniques can guide models to perform targeted improvements in grammar, style, and word choice while preserving technical terminology and authorial voice (Figure 6). By requiring the model to quote original text, we check for hallucinations. This is an effective approach when working with either text or data.

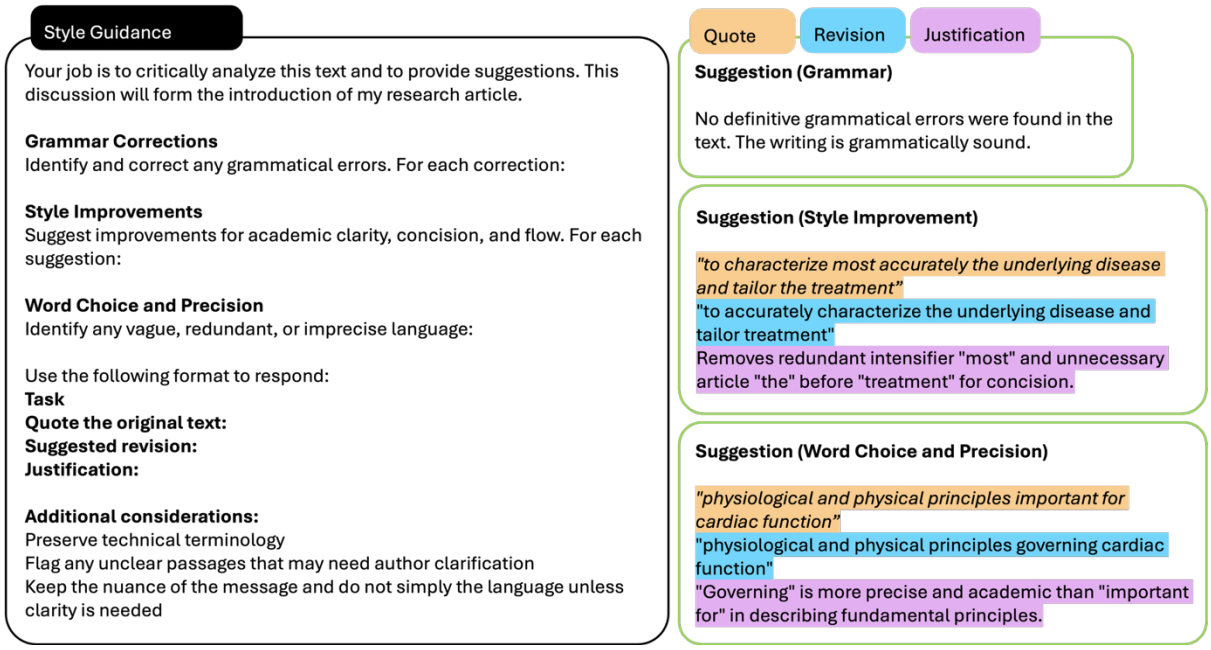

Figure 6. Recommended approach for leverage LLMs for editing text. The framework illustrates prompt engineering strategies for different editing tasks (grammar corrections, style improvements, and word choice refinement), with examples showing original text (orange), suggested revisions (blue), and justifications (purple) to maintain transparency during the editing process.

Recommendations for engineering effective zero-shot prompts:

- Be explicit. Give the LLM context as to what it is working with, its significance and where it will be used. Here, we pulled part of the introduction from one of our research articles (Niederer et al. 2019). By letting the LLM know its part of an introduction, there is less chance of non-compliance as it doesn't need to figure out where this may belong within the broader conversation
- Ask the LLM to look over your text and to make suggestions. Its justification for why it thinks this passage requires modification is an important window into understanding how it arrived at its conclusion
- Always ask the LLM to provide "ground truth". Here, we ask the LLM to quote directly from the article. This reduces the chance of hallucinations and is immediately verifiable by the user
- Ideally, provide examples of what good Grammar, Style and Word choice looks like. Provide examples of past introductions that had the qualities you would like to imbue your text with
- Provide additional considerations. An important consideration for scientific editing is to ensure the LLM does not simplify the language, that it preserves field-specific terminology. While stating this in the prompt helps, providing examples of field specific terminology is significantly more useful
- Beyond editing, LLMs can help with overcoming writing barriers (Aydin et al. 2023), and refine and proof-read communications from non-English speaking researchers (Smith et al. 2024).

3 Few-shot

In few-shot prompting, we provide the LLM with a small number of examples (typically 2-10) that demonstrate the desired input-output structure for a given task. This approach leverages the model's in-context learning capabilities, allowing it to infer task patterns without parameter updates. The number of demonstrations significantly impacts performance where even 2-3 examples can substantially improve task understanding compared to zero-shot approaches. Recent work has shown that scaling to many-shot regimes (hundreds to thousands of examples) can further improve output quality.

Specifically, providing extensive demonstrations has been shown to enhance generalization across diverse tasks, enable solutions to more complex problems requiring multi-step reasoning, and potentially mitigate certain biases acquired during pre-training (Agarwal et al. 2024). Moreover, this many-shot approach can reduce reliance on computationally expensive fine-tuning procedures, though at the cost of increased context length and potential exposure to adversarial exploits as demonstrated in recent jailbreaking studies. Prompt engineering follows a structured format where the task specification is accompanied by representative input-output pairs that implicitly state task requirements (Figure 7).

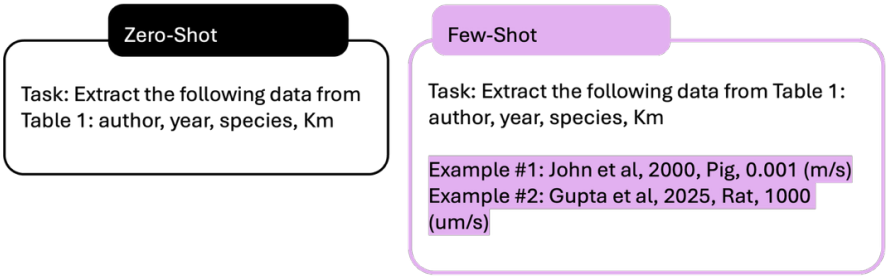

Figure 7. Comparison of zero-shot and few-shot prompting strategies for structured data extraction. In zero-shot prompting (left), only the task instruction is provided without examples. In few-shot prompting (right), the same task is augmented with demonstrative examples showing the expected output format and data transformation. Note the examples illustrate both formatting consistency (author citation style, year, species) and unit handling variability (m/s vs µm/s) that the model must learn to generalize from the provided context.

The quality, and repeatability of the output depends on several factors including:

1) The number of examples provided to the LLM. Typically, the more examples, the better (Brown et al. 2020), while ensuring to monitor overall token count. Agarwal et al. (2024) found that using many examples (>1000 (~85k tokens) in some cases) led to dramatic improvements in summarization, mathematical problem solving, algorithmic reasoning among many other tasks (Agarwal et al. 2024).

2) LLMs are sensitive to the order of information in prompts. Recent evidence demonstrates that the sequential arrangement of in-context examples can substantially impact model performance, with accuracy variations of 5.5-10.5 percentage points depending on example ordering alone (Bhope et al. 2025). This positional bias appears particularly pronounced in tasks requiring structured information extraction from

scientific literature. Figure 8 illustrates a simple mitigation by repositioning the task instruction from the terminal position to the initial position within the prompt architecture.

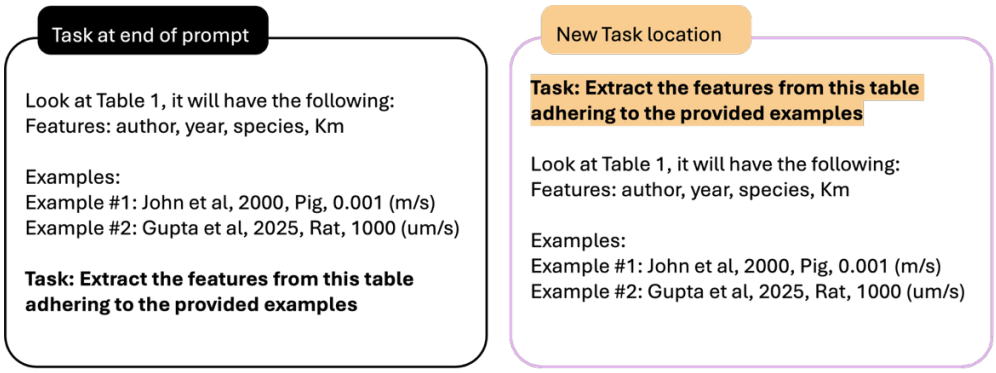

Figure 8. Prompt reordering strategy for improved task adherence in structured data extraction. The left panel shows the conventional prompt structure with task instructions positioned after contextual information and examples. The right panel demonstrates the optimized structure with task instructions relocated to the beginning of the prompt. Both configurations target extraction of bibliometric features (author, year, species, Km values) from tabular data, with identical examples provided (John et al., 2000; Gupta et al., 2025). This positional modification exploits the attention bias of large language models toward early prompt content, resulting in greater adherence to extraction templates.

3) Effective LLM-based extraction requires diverse in-context examples that reflect real-world data heterogeneity. Homogeneous example sets with perfectly formatted, complete entries result in brittle prompts that fail when encountering messy, poorly structured texts. Figure 9 contrasts these approaches: uniform examples versus strategically varied ones that include missing fields, alternative units (48h vs. 48 hours), concentration ranges, and embedded information within complex sentences. This diversification may significantly improve output quality and prompt adherence, as models trained on varied examples develop more robust pattern recognition for incomplete or differently formatted data.

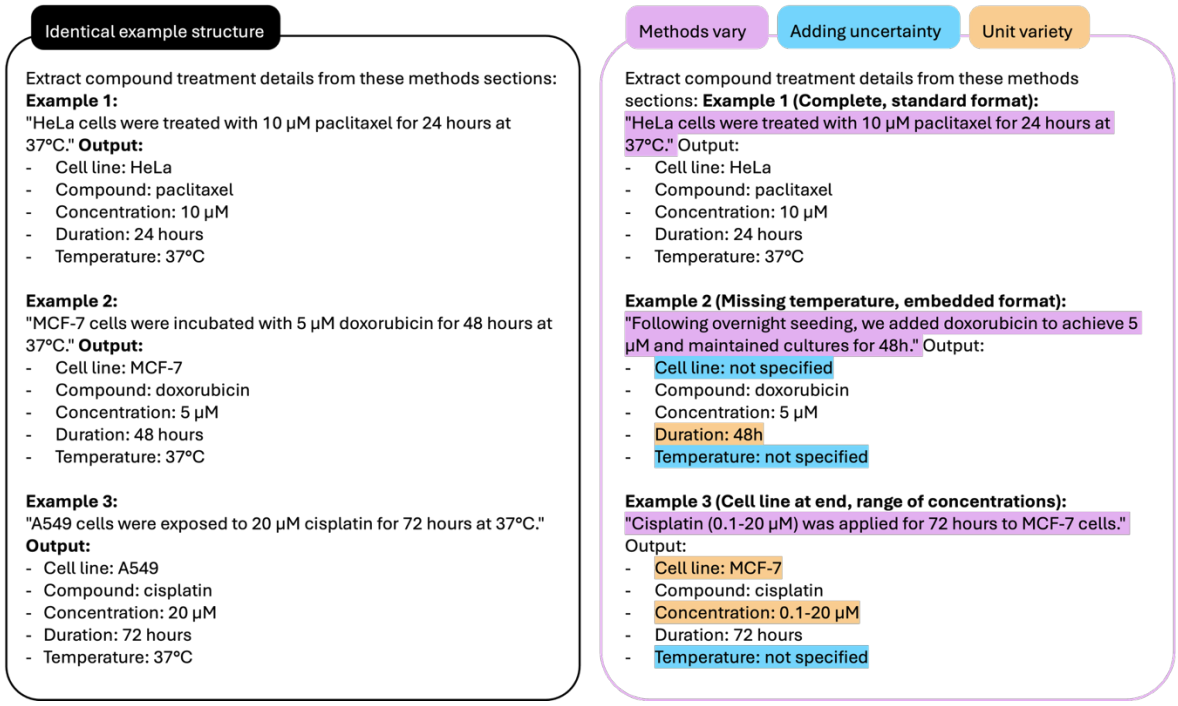

Figure 9. Example diversity improves extraction robustness. Left: Homogeneous prompt with three identically structured examples containing complete experimental parameters. Right: Diversified approach incorporating realistic variations with missing data fields (cell line, temperature), embedded information in narrative text, concentration ranges (0.1-20 µM), and a variety of units. Colour coding indicates variation types: purple (structural differences), cyan (missing/uncertain data), orange (unit variety).

Recommendations:

- Avoid using the same patterns, in the example on the left we used: [Cell line] + were + [verb] + with + [concentration] + [compound] + for + [duration] + at + [temperature]. With many examples that follow this order, we bias the model to look for this order and if it cant find it, it may have a harder time extracting information
- Avoid using the same constants, for example using 37 C for each example may cause the model to hallucinate to extract 37 C even if its not in the article or may force the model to return “not specified” because it couldn’t find it
- Provide a variety of representations of how cell lines may be shown within the article
- Include a variety of concentrations and units (uM, mM, mmol/L, mg/L etc). Avoid asking the LLM to convert to specific units unless multiple examples show case how to do this well.

4) Domain-specific training examples are surprisingly unnecessary for effective data extraction. State-of-the-art LLMs can successfully transfer extraction patterns across disparate scientific fields, relying on structural templates rather than domain-matched examples. Figure 10 demonstrates this cross-domain transferability where a prompt designed for cell biology experiments, with examples featuring neurons, siRNA transfection, and calcium imaging, successfully extracts analogous parameters from material science (thin film annealing) and analytical chemistry (chromatography). The model maps conceptual categories (sample→material, treatment→conditions,

measurement→analytical method, timepoints→duration) across domains without requiring field-specific training.

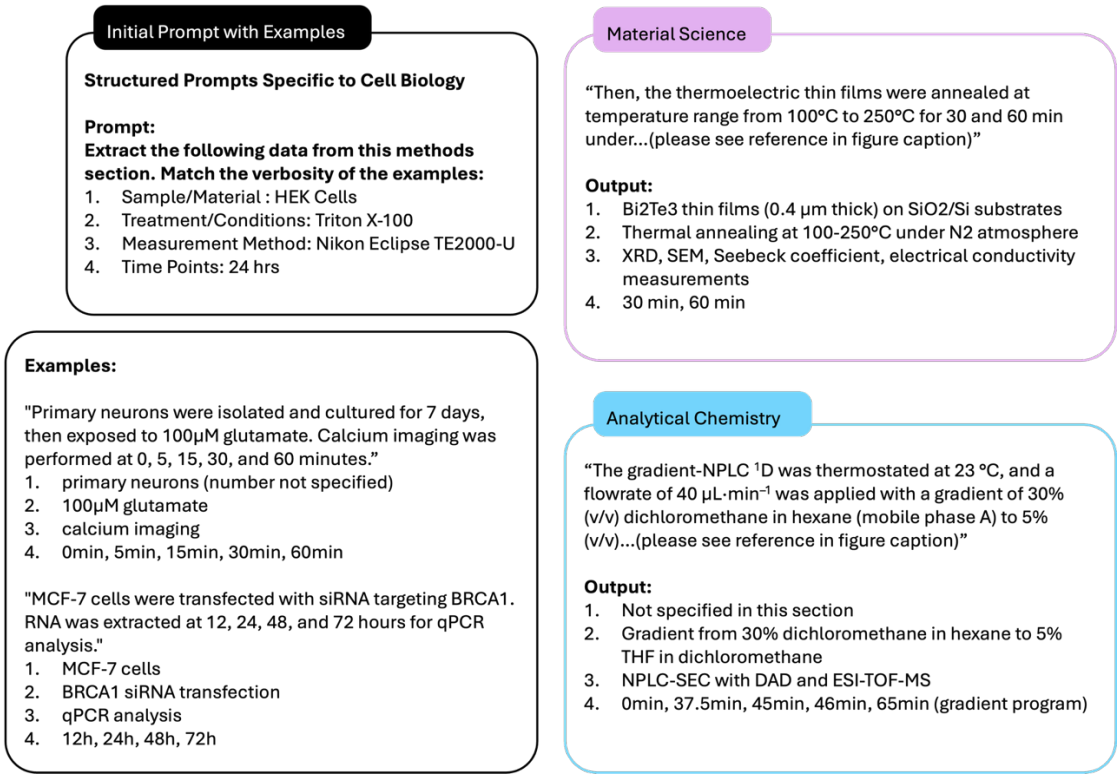

Figure 10. Cross-domain transferability of structured extraction prompts. Left: Cell biology-specific prompt with examples from neuron culture and transfection experiments. Right: Successful application to unrelated domains such as material science (Albert et al. 2025) (Bi_2Te_3 thin film processing) and analytical chemistry (Molenaar et al. 2022) (gradient NPLC-MS). Despite zero domain-relevant examples, the model correctly maps structural categories across fields, extracting temperature ranges, time points, and measurement methods.

It is generally good practice to ensure that the examples are similar to what the model may expect to encounter in the data it is given (Liu et al. 2021) but contain enough diversity in examples to account for many different ways that information can be presented, summarized and articulated (Wang et al. 2024). In fact, with just 18 examples, Su et al. (2022) showed 12.9% relative gain in performance when selecting examples that were both representative of the task and diverse in coverage within context for the task at hand (Su et al. 2022).

This contrasts markedly with scenarios involving poorly defined information retrieval. When prompt examples contain factual errors or methodological inconsistencies, model accuracy deteriorates significantly with studies demonstrating performance degradation of up to 80% when examples contain incorrect information (Yoo et al. 2022). As an example, a researcher has collected several studies where a specific experiment was analysed using an appropriately justified statistical test. If the statistical tests applied to the experiments are inappropriate, and these examples are provided to the LLM, the model is likely to learn these mistakes and in turn provide incorrect information or hallucinate incorrect answers.

Recommendations:

- When extracting information from research articles or other structured texts, focus on developing prompts with clear, consistent formatting. Provide multiple representative examples that demonstrate the exact output structure you require.
- The impact of incorrect examples varies dramatically with model scale. Smaller models (such as PaLM-8B or GPT-3 ada/babbage) rely heavily on their pre-trained knowledge and will largely ignore examples that contradict their semantic priors. This means incorrect examples may have minimal impact on their outputs. Conversely, larger state-of-the-art models (PaLM-540B, GPT-4) actively learn from the patterns in your examples and will override their pre-training to follow the examples you provide, even when those examples are incorrect (Wei et al. 2023).
- When working with large models, exercise particular care in example selection and validation. These models will faithfully reproduce any patterns present in your examples, including errors. This capability offers flexibility for novel tasks but requires significant quality control. Conversely, smaller models are significantly harder to influence with examples meaning it is much harder for them to override their pre-trained knowledge based on examples provided by the user

4 Thought Generation

Currently, the most capable models available to researchers utilize some form of "reasoning" which requires a fundamental level of language understanding to be able to articulate, by drawing from a discrete set of evidence and predictions for the task at hand (Zhang et al. 2025). When the LLM is tasked with a complex scientific question within mathematics, it must deduce the best way to approach the problem. Traditionally, the LLM attempts to solve the problem, "all at once". To date, the best performance has come from models that deploy a form of sequential thinking, by going through a series of intermediate reasoning steps, known as Chain-of-thought (CoT) (Wei, Wang, et al. 2022). This extra reasoning time has been shown to improve LLM accuracy within mathematics from 28% (no CoT) up to 90% when examples are included alongside reasoning (Liang 2025). Chain-of-thought prompting has seen tremendous uptake by the research community, deployed for coding (J. Li et al. 2025), speech translation (Hu et al. 2025), and education (Chan et al. 2025), just to name a few. At it's most basic, CoT can be invoked by asking the LLM to "think step by step". Providing examples alongside the reasoning prompt can further dramatically improve response accuracy (Liang 2025; Zhang et al. 2022). Figure 11 illustrates this effect on a microfluidics problem, without explicit reasoning instructions, the model conflates physical parameters (incorrectly using w=100 μm for channel width as it misunderstood the geometry considerations) and produces order-of-magnitude errors in droplet volume calculations.

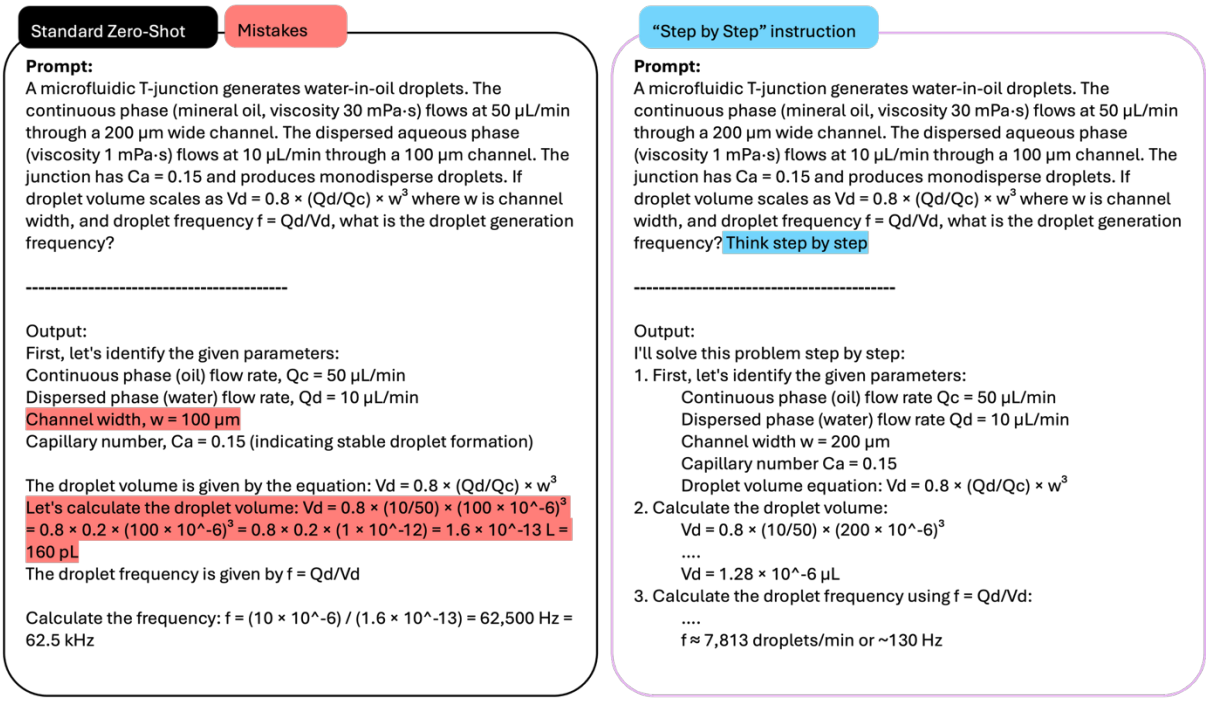

Figure 11. Chain-of-thought prompting can be beneficial under certain conditions. Left: A typical zero-shot prompt will typically struggle with logic or mathematics-based problems, in this case misidentifying channel width (100 µm instead of 200 µm), used wrong unit conversions, and a large error in final droplet output counts 62.5 kHz vs. actual ~130 Hz. Right: Adding "Think step by step" instruction produces correct sequential reasoning. Tested on Anthropic Haiku 3.5, a non-reasoning model. Red highlighting indicates mistakes in the non-reasoning approach

It was difficult to demonstrate the difference between "step by step" vs non-reasoning approaches simply because most models in late 2025 are reasoning models and utilize a form of CoT in responding to user queries.

4.1 On Reasoning

Reasoning models such as ChatGPT-5, Opus 4.1, Gemini 2.5 Pro, DeepSeek R1 etc are capable of multi-turn 'thinking' natively. Reasoning models assign a certain number of their available context to think over the problem. With platforms like AI Studio or via API, it is possible to limit or extend the number of tokens available for thinking. The prompt, the reasoning steps and the final output, all count towards the final token count. The longer the model can think over the task, as long as the prompt is well defined and specified, the greater the chance it will succeed at accomplishing it. On the other hand, since thinking consumes more tokens, the probability of hallucinations increase as more and more of the context window is consumed.

Some models, such as Opus 4 and ChatGPT-5 decide how "hard" to think. In practice, ChatGPT-5 will decide how long to think about a problem before giving the final answer. We found greater consistency in prompting the model to "think hard" rather than letting it decide. For example, we gave an unnumbered list of references to ChatGPT-5. We asked ChatGPT to count the number of references. In the first conversation, it thought for less than 5 seconds and answered, 77 references. After starting a new conversation,

ChatGPT-5 decided to think for longer, ~ 400 seconds, producing the correct count. Sometimes both Claude Opus 4.1 and Gemini 2.5 Pro will decide to think quickly, and typically get it wrong. We recommend strictly prompting models to think hard and to allocate a larger thinking budget, as long as the conversation is well defined.

Can we combine chain-of-thought prompting together with a reasoning model to boost performance? Latest research suggests, no, CoT appears to degrade performance. Combing CoT together with o1-preview (OpenAI), a reasoning model, significantly impacted accuracy (reducing by up to 36.3% compared to ChatGPT-4o) (Liu et al. 2025) when compared to zero-shot prompting. For a variety of tasks in cognitive psychology, CoT combined with reasoning models was shown to significantly decrease performance across the board. Non-reasoning models fared better but still saw a drop in performance. The key take away is that CoT performance is context dependent and will outperform in certain use cases (mathematical reasoning) while under performing in others (cognitive tasks that may cause the LLM to "overthink").

4.2 On Multiturn Conversations

Multi-turn conversations severely compromise extraction accuracy in scientific data mining tasks. Each iterative query compounds token consumption and forces the LLM to consider all information provided by the user and that it has generated during the course of the conversation (thinking tokens included). Figure 12 illustrates this degradation across four turns, the model progressively loses critical associations between drugs, proteins, and binding values, ultimately producing incomplete and incorrectly mapped data. However, this is not an issue when using a single prompt.

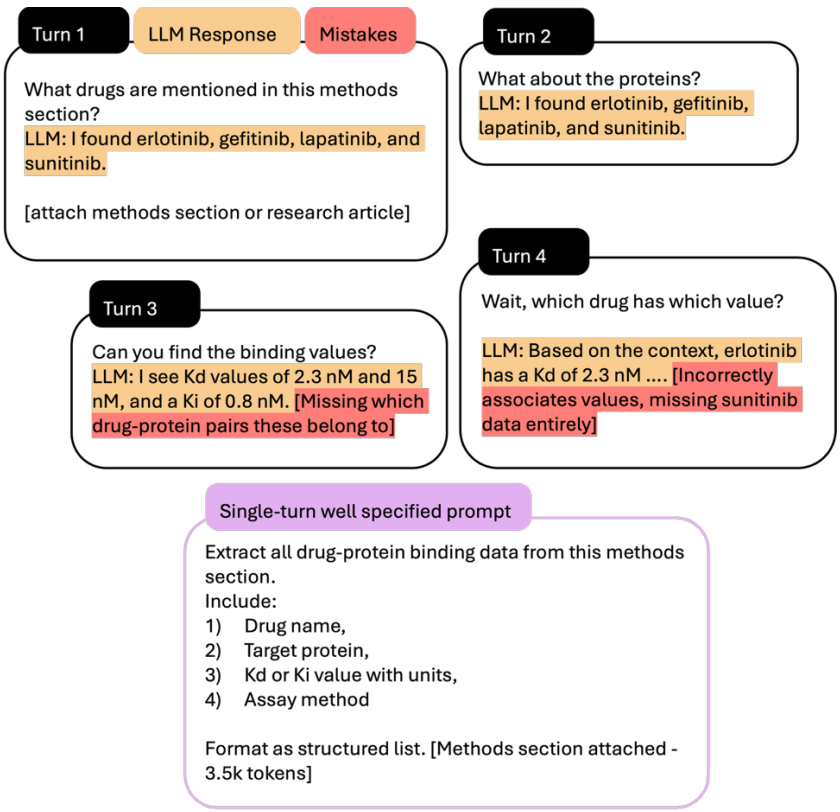

Figure 12. Multi-turn degradation compared to a well specified single-prompt for scientific data mining. Top: Four-turn conversation showing progressive loss of data integrity, where eventually the LLM encounters missing drug-protein associations (Turn 3), incorrect value mapping, and complete data omission (Turn 4). Bottom: Single-turn prompt with explicit instructions. Red highlights errors, missing associations and omitted data.

Work by Laban et al. (2025) (Laban et al. 2025) demonstrated sever performance degradation in all state-of-the-art models and closed weight LLMs during multi-turn conversations. Specifically, they showed, 1) a well-defined and specified initial first prompt had performance of 90% (a combination of aptitude and reliability), which dropped to 65% when, 2) the prompt was broken into multiple smaller underspecified prompts. Models with higher aptitude, that is higher intelligence, tend to be more reliable in single-turn conversations, however, will severely degrade in reliability under multi-turn conversations, regardless of how intelligent they are. One key takeaway from this study is that underspecified prompts will lead to LLMs reasoning in "the wrong direction", from which no amount of extra prompting, will help them recover. Poorly specified prompts snowball into worse subsequent assumptions, compounded by the fact that the model typically skips asking the user for clarification when the users intent is ambiguous (Shaikh et al. 2025).

Recommendations:

- If your initial prompt did not work, try the same prompt again in a new conversation (Laban et al. 2025). Since LLMs are probabilistic, it's likely that the LLM may take the wrong initial reasoning step. This is reflected in typical online discourse of one person saying "this LLM is not capable, it wasn't able to do X", with another person responding, "your prompt worked for me".
- If the model starts hallucinating or underperforming during a multiturn conversation, stop, and ask the model to "consolidate all the prompts I have given to you, into a single prompt, as they are without inferring intent". This is a great way to build a strong, highly specific first prompt.
- Stick to the topic. If the task is data extraction, the prompt should cover data extraction with examples. Do not attempt to layer multiple reasoning steps into the prompt (Becker et al. 2025), for example, avoid "Extract the following: Temp (C), Species, Kd from Table 1. Then calculate K (umM) and display each reference next to the extracted values. Include the title and year of publication." Break prompt to focus on the task at hand, without deviating.
- Provide the LLM with your initial best prompt and ask it for clarifying questions based on your prompt. This initial check can catch ambiguities further improving model performance.

5 Ensembling

The following set of techniques, while highly effective, are underutilized among academics using LLM chatbot interfaces. In our experience, there is an assumption, a prompt that successfully extracts data or provides correct information will perform identically in subsequent conversations. However, due to the non-deterministic nature of LLMs, even a well-structured and carefully specified prompt may produce

inconsistent outputs across separate conversations, occasionally yielding incorrect responses despite previous performance. One way to address this specific variability is to execute the same prompt across multiple independent conversations (*n* trials) for identical tasks, to identify the most frequent response pattern (Figure 13). For data extraction tasks, this approach increases the likelihood of converging on the ground truth present in source materials. For classification and analysis tasks, this method reduces output variance and improves reliability.

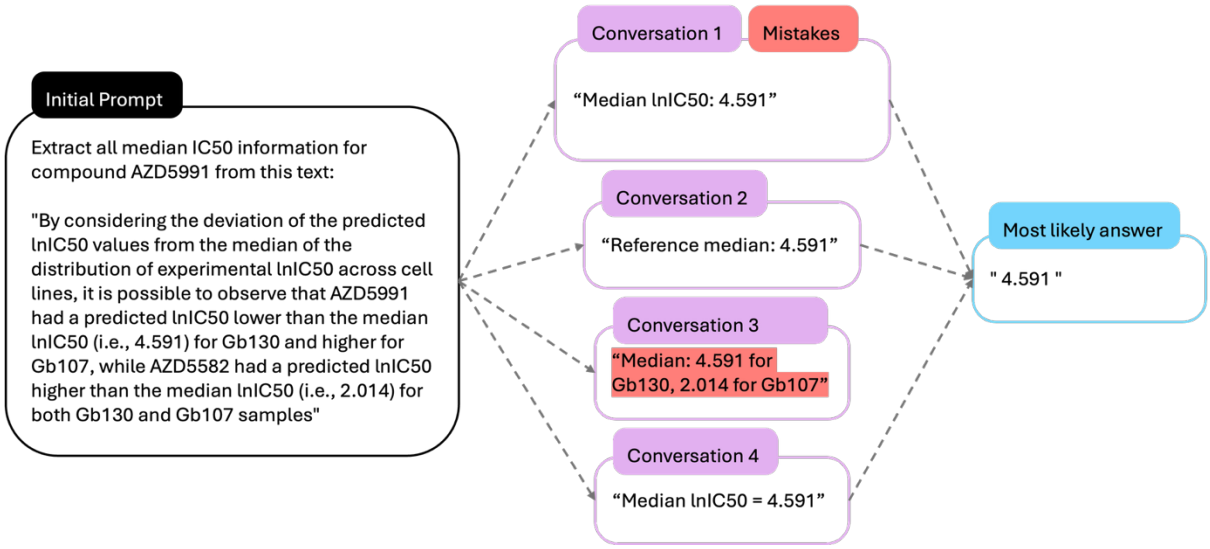

Figure 13. Example of Ensembling. Left: Initial prompt requesting median lnIC50 extraction for compound AZD5991 from a snippet of text (Carli et al. 2025). Right: Four independent conversations, using the same prompt and snippet of text resulted in three correctly identify median values, while one trial (Conversation 3) erroneously attributes both 4.591 and 2.014 to AZD5991, despite 2.014 being the median for AZD5582. The ensemble approach converges on the correct answer (4.591) by consensus.

There are several variations of Ensembling that extend beyond simple repeated queries with majority voting (i.e. re-running the same prompt in a new conversation). The self-consistency approach developed by Wang et al. (2023) demonstrated that sampling multiple reasoning paths and selecting the most frequent answer significantly improves accuracy (+17.9% on GSM8K) (X. Wang et al. 2023). Building on this work, adaptive-consistency methods are those that dynamically adjust the number of samples based on agreement levels, reducing computational costs by up to 7.9x while maintaining accuracy (less than 0.1% accuracy drop) (Aggarwal et al. 2023). Further, confidence-weighted approaches achieve 40% reduction in required samples by prioritizing high-confidence responses over simple vote counting (Taubenfeld et al. 2025). It's important to note that LLMs exhibit significant non-determinism even at temperature=0 (the parameter controlling output randomness), with accuracy varying dramatically across tasks and models.

Depending on the task, the input provided to the LLM and the desired output, multiple sampling could be essential. For practical implementation, research suggests using at least 5 independent samples for reliability assessment (Megahed et al. 2025). Depending on the task, utilizing semantic similarity rather than exact string matching for aggregation

(Oh and Lee 2025) can also be effective. These techniques have proven effective in production environments, with ensemble methods showing improvements in key business metrics (Fang et al. 2024), suggesting that the computational overhead of multiple sampling is justified by substantial gains in accuracy and reliability for critical data extraction tasks.

5.1 On Deep Research Mode

Deep Research was initially released within the ChatGPT suit of tools by OpenAI in February, 2025. Described as an agentic tool, Deep Research promised to “find, analyse, and synthesize hundreds of online sources to create a comprehensive report at the level of a research analyst”(‘Introducing Deep Research’ 2025). Similarly in late 2025, Anthropic described the development of their Deep Research framework, a suite of subagents orchestrated by a lead agent that manages memory and citations, “When a user submits a query, the lead agent analyses it, develops a strategy, and spawns subagents to explore different aspects simultaneously” (Anthropic 2025c)

In our use of Deep Research tools provided by either Google, Anthropic, or Perplexity, the exact same prompt used with the same agent elicits markedly different reports each time (Figure 14). When asked to find research articles on "decomposition" with applications to science, word counts varied substantially between duplicate runs, with Gemini showing the largest variance (4,319 vs 3,299 words). Most notably, reference overlap between duplicate reports remained low across all models, ranging from 5% for Perplexity to 50% for ChatGPT, with Gemini and Claude showing 25% and 12% overlap respectively. This suggests that, at least in our limited sample, no definitive 'ground truth' references exist. This means that with each report the “key takeaway” from each report could vary. Gemini had the largest proportion of non-academic citations (~37%, 1st report) followed by Perplexity. On average, each Deep Research agent produced roughly 30 citations per report. Inconsistencies in output length, source selection, and citation patterns highlight the challenge of reproducibility in Deep Research tools (Wong et al. 2025). For scientific work, such variability adds to the academic burden of reviewing and parsing information rather than reducing it. Evaluating the quality of each report is complicated and time consuming, prompting Du et al. (2025) to develop an LLM-judge driven framework for analysing Deep Research articles by benchmarking them on 100 PhD level problems across 22 distinct fields (Du et al. 2025).

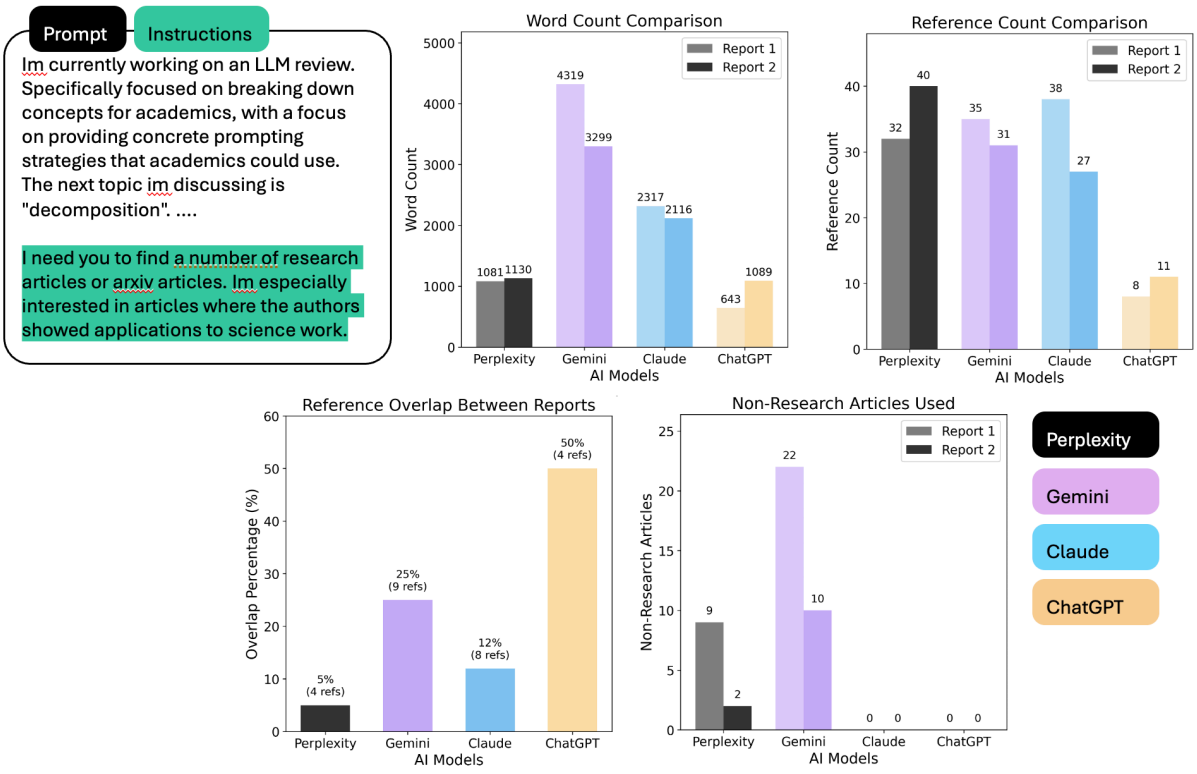

Figure 14. Reproducibility analysis of Deep Research tools across duplicate runs using the same prompt. Left: The prompt provided to each model. The prompt was developed in an 'opportunistic' prompting style, that is, we did not engineer this prompt to reflect how most people use these tools. Right panels: Comparison metrics between two independent runs (Report 1: light bars, Report 2: darker bars) showing (i) word count variation with differences up to 1,020 words within the same model, (ii) reference count inconsistencies ranging from 8-13 citations difference, (iii) reference overlap demonstrating only 5-50% shared citations between duplicate reports, and (iv) non-research article inclusion varying from 0-22 sources. We used the free versions of Gemini, ChatGPT and Perplexity.

There may be value in applying Ensembling principles to generation of Deep Research articles. In this instance, the Deep Research agent would initialize several concurrent searches, generating several reports which are then filtered and distilled based on the initial prompt requirements.

5.2 On Hallucinations and Self-criticism

Much has been written on LLM hallucinations (Rudolph et al. 2023; Emsley 2023; Alkaissi and McFarlane 2023) and the fact remains, hallucinations are one of the majors reasons for low adoption of LLMs within academic and business pipelines. Hallucinations are non-factual responses, such as an answer to a question, or generated academic references that do not exist, typically persuasively presented to the user as fact. Latest research from OpenAI has concluded that hallucinations arise from errors in binary classifications(Kalai et al. 2025). That is, the model is rewarded for providing/guessing an answer and penalized if it doesn't know. As such, the model is incentivized to "always know", even when it doesn't.

While the frequency of non-factual responses has significantly reduced since the days of ChatGPT-3, hallucinations are now harder to spot and are potentially more insidious

because of it. The advent of ChatGPT-5 Pro and similar SOTA models has significantly increased the scope of where these models can be used, having shown remarkable ability to tackle hard maths problems. As LLM capability and accuracy improves, more trust and autonomy will be given to the models. Most of the time the calculations will be correct. Most of the times, the references will be factual. For example, from the OpenAI o3 System Card, "*o3 tends to make more claims overall, leading to more accurate claims as well as more inaccurate/hallucinated claims*" (OpenAI 2025b). The question becomes, when should we take the time and effort to validate LLM output and when can we just trust it? As it stands, all LLM output should be validated.

A strong high-performing strategy has been developed to address these challenges: forcing the LLM to self-critique or self-reflect on its output (Yan et al. 2025). At its simplest, this tactic asks the LLM to critically evaluate the output it has generated and to either confirm that the output is correct or to suggest improvements (Figure 15). Iterative self-refinement with a simple few-shot strategy has been shown to dramatically improve model performance (GPT-3.5 and GPT-4) over direct generation by between 5 to 40% (Madaan et al. 2023). A similar strategy can be used alongside chain-of-thought prompting (Huang et al. 2022), and extended to language agents that maintain their own reflections to enhance discission-making in subsequent trials (Shinn et al. 2023). It is important to note that prompt engineering tactics between LLMs and Small Language Models (SLMs) are not always transferable. Yan at al. (2025) developed a self-reflection methodology for SMLs that increased the reasoning accuracy to 36.2% while being 10 times more efficient that the method it was built upon (Yan et al. 2025).

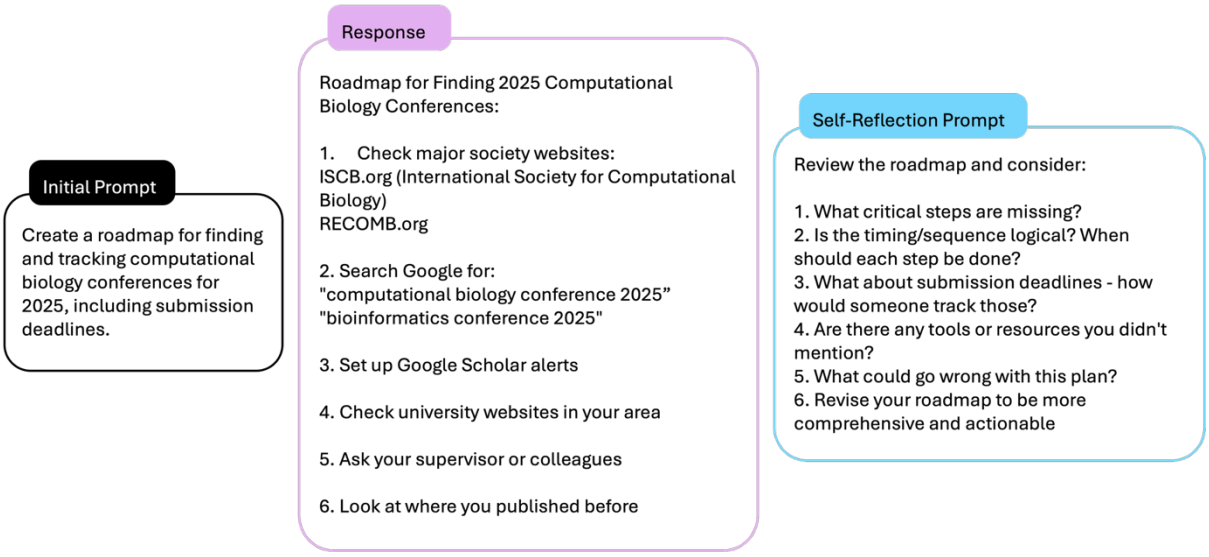

Figure 15. Self-reflection prompting to identify and correct LLM output. Left: Initial prompt is missing a lot of detail and information. Middle: Output based on the query. Right: Self-reflection prompt forces critical evaluation through both broader or more targeted questions. The user can ask the LLM to generate the self-reflection prompt based on deficiencies it observed in the response as a function of the input.

A slight modification to reflecting on the output is to reflect on the user's input. The LLM can prompt the user to provide more targeted information regarding their request, helping to build a much higher quality, well specified prompt by removing instances where the model must infer the user's intent. This is similar to ChatGPT's Deep Research

mode which asks the user for further information to refine the initial input before starting search. In our experience, this is an effective strategy for refining the prompts.

6 Decomposition

Prompt Decomposition is similar to CoT prompting. Prompting an LLM to perform CoT tasks the LLM with breaking the problem down into manageable sub-steps and to subsequently solve these steps within the same prompt output. Decomposition on the other hand, tasks the LLM with, 1) breaking the larger problem into sub-problems, 2) solving the first sub-problem, 3) appending it's solution back to the main prompt with all sub-problems. The main prompt now contains all sub-problems plus the first solution. Using the sub-problems and solution as context, the LLM moves on to the next sub-problem (Zhou et al. 2023). Prompt decomposition is one of several popular prompt engineering techniques for improving multi-agent communication which can involve orchestration of multiple agents and assignment of sub-problems for specialist LLMs to solve before bringing the solutions back to the main architect agent (Li et al. 2023).

Prompt decomposition can be invoked by simply asking the LLM to *"Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step"* (L. Wang et al. 2023). Prompt performance can be further improved by explicitly suggesting to the LLM what it should focus on during the planning stage, *"Let's first understand the problem, extract relevant variables and their corresponding numerals, and make a plan. Then, let's carry out the plan, calculate intermediate variables (pay attention to correct numerical calculation and commonsense), solve the problem step by step, and show the answer"*.

The issue with asking the LLM to, 1) understand the problem, 2) break down the main components, 3) devise a plan for the sub-problems, and to then 4) solve the sub-problem in a single turn conversation, is token consumption. LLMs have a finite number of tokens they can process and attend to, per conversation. Further, the number of tokens or words the LLM can produce as a response to a query is also limited. For example, in AI Studio from Google it is possible to set the maximum token output for Gemini 2.5 Pro to 65,536 tokens('Google AI Studio' 2024). In practice, an LLM must contain its response within the maximum output limits set by the platform or user which in turn limits the depth of the response (Rasal and Hauer 2024).

To clearly demonstrate this concept, we will use the work of Toufiq et al. (2023) as a case study (Toufiq et al. 2023). Here, the authors used 4 different LLMs together with a set of decomposed prompts to prioritize and select candidate genes from blood transcriptional modules for biomarker selection. Rather than attempting to evaluate all 30-40 genes simultaneously across six criteria while also integrating expression data and verifying references, they decomposed the problem into manageable chunks (Figure 16). Each prompt focused on a single, well-defined task: identifying functional convergences (Step 1), scoring individual genes one at a time against specific criteria (Step 2), summarizing justifications for only the top 5 candidates (Step 3), fact-checking individual statements (Step 4), selecting a top candidate based on curated summaries (Step 5), and progressively incorporating transcriptional data (Step 6-7).

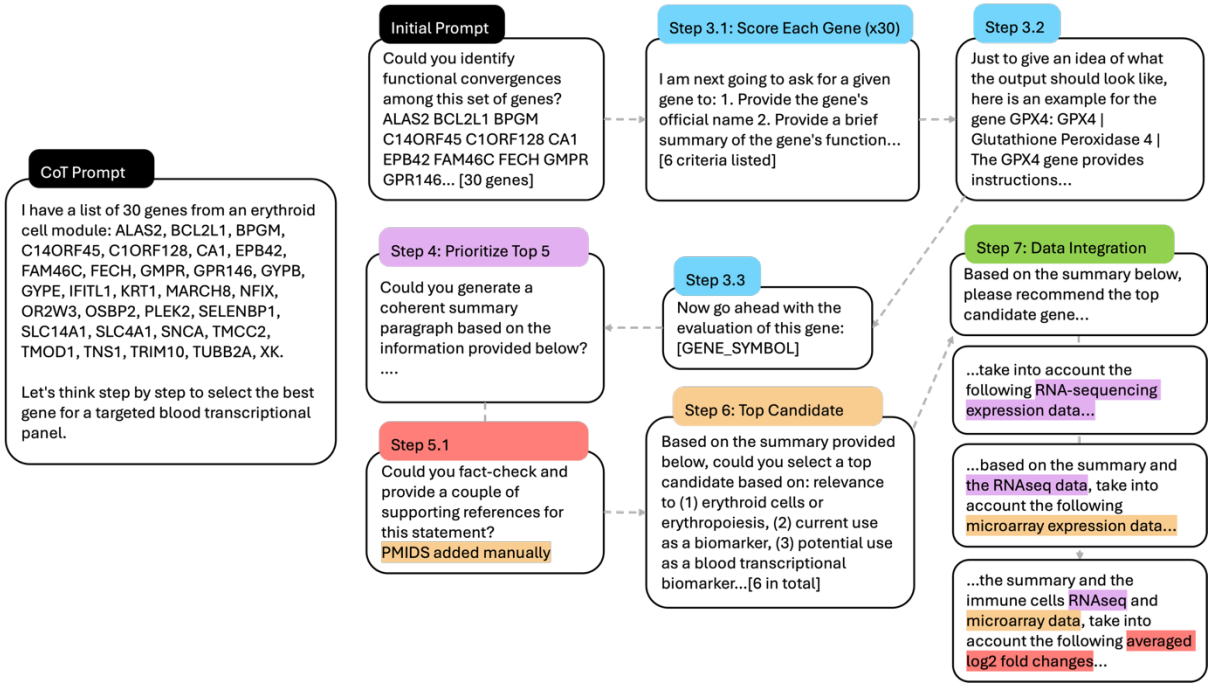

Figure 16. Task decomposition strategy for complex biomarker selection. Left: Chain-of-thought (CoT) prompt outlining the complete 30-gene evaluation task. Right: Decomposed workflow breaking the problem into seven discrete prompts, each with focused scope. Steps progress from functional convergence identification (Step 1 and 2), through individual gene scoring (Step 3.1-3.3, processing one gene at a time), to data integration (Steps 6-7, incorporating RNA-seq and microarray data). Coloured text indicates addition of external knowledge to the workflow. Adapted with permission (Toufiq et al. 2023) under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

One approach to addressing context window limitations involves initiating new conversation instances for each sub-problem requiring a solution. Decomposition strategies and established best practices have facilitated the development of next-generation academic tools with significant adoption observed within the fields of computer science and software engineering (Jin et al. 2025). Agentic frameworks including Claude Code (Anthropic 2025b), Codex (OpenAI 2025a), and Gemini CLI (Google 2025) leverage their respective large language models to both interact with users and execute functions on local systems. Operating through terminal interfaces, these tools can create and edit a variety of different file formats, including research articles, documents, and presentations, while also performing web searches, interfacing with GitHub repositories, and generating code.

Many of the prompt engineering principles described in this section are readily observable in agentic frameworks. For instance, Claude Code implements an explicit planning phase (accessible via the /plan command) prior to task execution (Figure 17). For complex tasks, rather than processing plans sequentially, specific sub-problems can be delegated to sub-agents. Sub-agents are independent model instances operating in parallel or series to the main conversation. Sub-agents are orchestrated by the primary instance, maintain separate context windows and return solutions to designated sub-problems without consuming or interfering with the main conversation's context, effectively multiplying the available working memory.

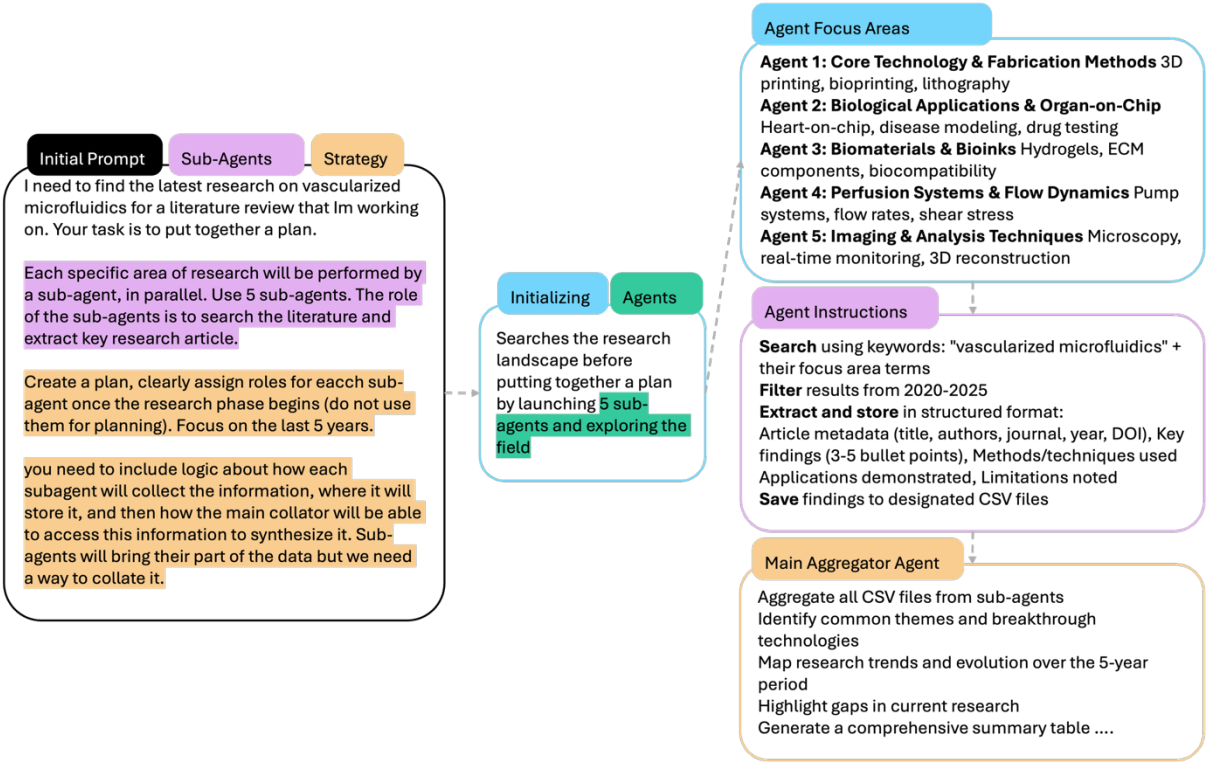

Figure 17. Multi-agent framework for performing a literature review utilizing parallelized sub-agents. Initial prompt defines the research scope (vascularized microfluidics, 2020-2025), which is then decomposed into five parallel sub-agents, each focusing on distinct research domains: core technology and fabrication methods, biological applications, biomaterials and bioinks, perfusion systems, and imaging techniques. Each sub-agent operates with independent context windows, extracting structured data (metadata, key findings, methods, applications, limitations) into CSV files. The main aggregator agent synthesizes outputs from all sub-agents to identify common themes, breakthrough technologies, and research trends across the 5-year period, effectively multiplying available working memory while maintaining task coherence.

6.1 On Prompt Templates

Prompt templates can be effective for capturing the logic behind specific data extraction patterns, maintaining stylistic control over document editing, or as a starting point for designing more tailored prompts. They are especially important in industrial settings (Mao et al. 2025), where consistent LLM performance is critical to business function. The prompts presented here serve as foundational frameworks for understanding prompt engineering principles rather than prescriptive templates, enabling researchers to develop domain-specific prompts based on the systematic approaches outlined throughout this review. We encourage users to develop a key set of principles for their LLM to follow consistently and to deploy these principles in prompts that may be used across multiple documents or applications. Further, we encourage using the LLM itself to help brainstorm and create task-specific prompts. For example, prompting Claude Opus 4.1 to create a prompt template for extracting data from electrophysiology experiments resulted in a highly detailed prompt containing problem specifics, multiple few-shot examples, output examples, JSON extraction patterns and instructions to cover edge cases.

7 Conclusion

This review was our attempt to synthesize and explore various high-impact prompt engineering ideas, with our observations and reflections while providing specific use cases relevant to academics within the life science. The Prompt Report describes 58 different prompt engineering techniques, here we focus on a small subset, providing rich examples, recommendations and references to other studies where prompt engineering techniques were developed and utilized. While it's possible that next-generation of models may reduce the need for prompt engineering, so far the opposite has actually held true. Agentic systems like Claude Code, while highly capable, require explicit decomposition of tasks to stay on track. We hope that the case studies and examples explored in this review will help others to develop their own high-quality, field specific prompts that generate reliable, trustworthy and reproducible outputs for their field-specific research applications.

Acknowledgements
V.R. would like to acknowledge Prof. Emeritus Cristobal Dos Remedios for suggesting publishing the various ideas that we have presented here.

Conflict of Interest

The authors have no relevant financial or non-financial interests to disclose.

References

Agarwal, Rishabh, Avi Singh, Lei M. Zhang, et al. 2024. ‘Many-Shot In-Context Learning’. arXiv:2404.11018. Preprint, arXiv, October 17. https://doi.org/10.48550/arXiv.2404.11018.

Aggarwal, Pranjal, Aman Madaan, Yiming Yang, and Mausam. 2023. ‘Let’s Sample Step by Step: Adaptive-Consistency for Efficient Reasoning and Coding with LLMs’. arXiv:2305.11860. Preprint, arXiv, November 16. https://doi.org/10.48550/arXiv.2305.11860.

Albert, Emőke, Péter Basa, Bálint Fodor, et al. 2025. ‘Experimental and Computational Synthesis of TiO2 Sol–Gel Coatings’. *Langmuir* 41 (1): 704–18. https://doi.org/10.1021/acs.langmuir.4c03959.

Alkaissi, Hussam, and Samy I McFarlane. 2023. ‘Artificial Hallucinations in ChatGPT: Implications in Scientific Writing’. *Cureus*, ahead of print, February 19. https://doi.org/10.7759/cureus.35179.

Amad, Harry, Nicolás Astorga, and Mihaela van der Schaar. 2025. ‘Continuously Updating Digital Twins Using Large Language Models’. arXiv:2506.12091. Preprint, arXiv, July 21. https://doi.org/10.48550/arXiv.2506.12091.

An, Shengnan, Zexiong Ma, Zeqi Lin, Nanning Zheng, and Jian-Guang Lou. 2024. ‘Make Your LLM Fully Utilize the Context’. arXiv:2404.16811. Preprint, arXiv, April 26. https://doi.org/10.48550/arXiv.2404.16811.

Andersen, Jens Peter, Lise Degn, Rachel Fishberg, et al. 2025. ‘Generative Artificial Intelligence (GenAI) in the Research Process – A Survey of Researchers’ Practices and Perceptions’. *Technology in Society* 81 (June): 102813. https://doi.org/10.1016/j.techsoc.2025.102813.

Anthropic. 2024a. ‘Prompt Generator’. https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/prompt-generator.

Anthropic. 2024b. ‘System Prompts’. https://docs.anthropic.com/en/release-notes/system-prompts.

Anthropic. 2025a. ‘Claude 4 Prompt Engineering Best Practices’. https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/claude-4-best-practices.

Anthropic. 2025b. *Claude Code: Agentic Coding Tool*. Anthropic, released. https://github.com/anthropics/claude-code.

Anthropic. 2025c. ‘How We Built Our Multi-Agent Research System’. https://www.anthropic.com/engineering/multi-agent-research-system.

Araujo, Pedro Henrique Luz de, and Benjamin Roth. 2025. 'Helpful Assistant or Fruitful Facilitator? Investigating How Personas Affect Language Model Behavior'. *PLOS ONE* 20 (6): e0325664. https://doi.org/10.1371/journal.pone.0325664.

Aydin, Abdulkerim, Süleyman Eren Yürük, İlknur Reisoğlu, and Yuksel Goktas. 2023. 'Main Barriers and Possible Enablers of Academicians While Publishing'. *Scientometrics* 128 (1): 623–50. https://doi.org/10.1007/s11192-022-04528-x.

Becker, Jonas, Lars Benedikt Kaesberg, Andreas Stephan, Jan Philip Wahle, Terry Ruas, and Bela Gipp. 2025. 'Stay Focused: Problem Drift in Multi-Agent Debate'. arXiv:2502.19559. Preprint, arXiv, May 21. https://doi.org/10.48550/arXiv.2502.19559.

Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. 'On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜'. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '21, March 1, 610–23. https://doi.org/10.1145/3442188.3445922.

Bhope, Rahul Atul, Praveen Venkateswaran, K. R. Jayaram, Vatche Isahagian, Vinod Muthusamy, and Nalini Venkatasubramanian. 2025. 'OptiSeq: Ordering Examples On-The-Fly for In-Context Learning'. arXiv:2501.15030. Preprint, arXiv, February 18. https://doi.org/10.48550/arXiv.2501.15030.

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, et al. 2022. 'On the Opportunities and Risks of Foundation Models'. arXiv:2108.07258. Preprint, arXiv, July 12. https://doi.org/10.48550/arXiv.2108.07258.

Brown, Tom, Benjamin Mann, Nick Ryder, et al. 2020. 'Language Models Are Few-Shot Learners'. *Advances in Neural Information Processing Systems* 33: 1877–901. https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Capstick, Alexander, Rahul G. Krishnan, and Payam Barnaghi. 2025. 'AutoElicit: Using Large Language Models for Expert Prior Elicitation in Predictive Modelling'. arXiv:2411.17284. Preprint, arXiv, May 28. https://doi.org/10.48550/arXiv.2411.17284.

Carli, Francesco, Pierluigi Di Chiaro, Mariangela Morelli, et al. 2025. 'Learning and Actioning General Principles of Cancer Cell Drug Sensitivity'. *Nature Communications* 16 (1): 1654. https://doi.org/10.1038/s41467-025-56827-5.

Chan, Kuang Wen, Farhan Ali, Joonhyeong Park, et al. 2025. 'Automatic Item Generation in Various STEM Subjects Using Large Language Model Prompting'. *Computers and Education: Artificial Intelligence* 8 (June): 100344. https://doi.org/10.1016/j.caeai.2024.100344.

Chang, Yung-Chun, Ming-Siang Huang, Yi-Hsuan Huang, and Yi-Hsuan Lin. 2025. 'The Influence of Prompt Engineering on Large Language Models for Protein–Protein

Interaction Identification in Biomedical Literature'. *Scientific Reports* 15 (1): 15493. https://doi.org/10.1038/s41598-025-99290-4.

DeHaan, Soren, Yuanze Liu, Johan Bollen, and Sa'ul A. Blanco. 2025. 'GPT Editors, Not Authors: The Stylistic Footprint of LLMs in Academic Preprints'. arXiv:2505.17327. Preprint, arXiv, May 22. https://doi.org/10.48550/arXiv.2505.17327.

Du, Mingxuan, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. 2025. 'DeepResearch Bench: A Comprehensive Benchmark for Deep Research Agents'. arXiv:2506.11763. Preprint, arXiv, June 13. https://doi.org/10.48550/arXiv.2506.11763.

Emsley, Robin. 2023. 'ChatGPT: These Are Not Hallucinations–They're Fabrications and Falsifications'. *Schizophrenia* 9 (1): 52.

Evstafev, Evgenii. 2025. 'Token-by-Token Regeneration and Domain Biases: A Benchmark of LLMs on Advanced Mathematical Problem-Solving'. arXiv:2501.17084. Preprint, arXiv, January 28. https://doi.org/10.48550/arXiv.2501.17084.

Fang, Chenhao, Xiaohan Li, Zezhong Fan, et al. 2024. 'LLM-Ensemble: Optimal Large Language Model Ensemble Method for E-Commerce Product Attribute Value Extraction'. arXiv:2403.00863. Preprint, arXiv, June 20. https://doi.org/10.48550/arXiv.2403.00863.

Google. 2025. *Gemini CLI: Open-Source AI Agent for Terminal*. Google, released June. https://github.com/google-gemini/gemini-cli.

'Google AI Studio'. 2024. https://aistudio.google.com/.

'GPT-5 Prompting Guide'. 2025. https://cookbook.openai.com/examples/gpt-5/gpt-5_prompting_guide.

Hanson, Mark A., Pablo Gómez Barreiro, Paolo Crosetto, and Dan Brockington. 2024. 'The Strain on Scientific Publishing'. *Quantitative Science Studies* 5 (4): 823–43. https://doi.org/10.1162/qss_a_00327.

Hu, Ke, Zhehuai Chen, Chao-Han Huck Yang, et al. 2025. 'Chain-of-Thought Prompting for Speech Translation'. *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April, 1–5. https://doi.org/10.1109/ICASSP49660.2025.10890560.

Huang, Jiaxin, Shixiang Shane Gu, Le Hou, et al. 2022. 'Large Language Models Can Self-Improve'. arXiv:2210.11610. Preprint, arXiv, October 25. https://doi.org/10.48550/arXiv.2210.11610.

Hwang, Taesoon, Nishant Aggarwal, Pir Zarak Khan, et al. 2024. 'Can ChatGPT Assist Authors with Abstract Writing in Medical Journals? Evaluating the Quality of

Scientific Abstracts Generated by ChatGPT and Original Abstracts'. *PLOS ONE* 19 (2): e0297701. https://doi.org/10.1371/journal.pone.0297701.

Imani, Shima, Liang Du, and Harsh Shrivastava. 2023. 'MathPrompter: Mathematical Reasoning Using Large Language Models'. arXiv:2303.05398. Preprint, arXiv, March 4. https://doi.org/10.48550/arXiv.2303.05398.

'Introducing Deep Research'. 2025. https://openai.com/index/introducing-deep-research/.

Jin, Haolin, Linghan Huang, Haipeng Cai, Jun Yan, Bo Li, and Huaming Chen. 2025. 'From LLMs to LLM-Based Agents for Software Engineering: A Survey of Current, Challenges and Future'. arXiv:2408.02479. Preprint, arXiv, April 13. https://doi.org/10.48550/arXiv.2408.02479.

Jin, Hongye, Xiaotian Han, Jingfeng Yang, et al. 2024. 'LLM Maybe LongLM: Self-Extend LLM Context Window Without Tuning'. arXiv:2401.01325. Preprint, arXiv, July 11. https://doi.org/10.48550/arXiv.2401.01325.

Jin, Mingyu, Haochen Xue, Zhenting Wang, et al. 2024. 'ProLLM: Protein Chain-of-Thoughts Enhanced LLM for Protein-Protein Interaction Prediction'. arXiv:2405.06649. Preprint, arXiv, July 12. https://doi.org/10.48550/arXiv.2405.06649.

Kalai, Adam Tauman, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. 'Why Language Models Hallucinate'. arXiv:2509.04664. Preprint, arXiv, September 4. https://doi.org/10.48550/arXiv.2509.04664.

Kobak, Dmitry, Rita González-Márquez, Emőke-Ágnes Horvát, and Jan Lause. 2025. 'Delving into LLM-Assisted Writing in Biomedical Publications through Excess Vocabulary'. *Science Advances* 11 (27): eadt3813. https://doi.org/10.1126/sciadv.adt3813.

Laban, Philippe, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. 'LLMs Get Lost In Multi-Turn Conversation'. arXiv:2505.06120. Preprint, arXiv, May 9. https://doi.org/10.48550/arXiv.2505.06120.

Li, Ang, Haozhe Chen, Hongseok Namkoong, and Tianyi Peng. 2025. 'LLM Generated Persona Is a Promise with a Catch'. arXiv:2503.16527. Preprint, arXiv, March 18. https://doi.org/10.48550/arXiv.2503.16527.

Li, Guohao, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. 'CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society'. arXiv:2303.17760. Preprint, arXiv, November 2. https://doi.org/10.48550/arXiv.2303.17760.

Li, Jia, Ge Li, Yongmin Li, and Zhi Jin. 2025. 'Structured Chain-of-Thought Prompting for Code Generation'. *ACM Trans. Softw. Eng. Methodol.* 34 (2): 37:1-37:23. https://doi.org/10.1145/3690635.

Liang, Jessica E. 2025. ‘Chain-of-Thought Reasoning for Math: Theoretical Foundation and Applications’. Paper presented at 2nd AI for Math Workshop @ ICML 2025. July 16. https://openreview.net/forum?id=G3eCchXqke&noteId=G3eCchXqke.

Liu, Jiachang, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. ‘What Makes Good In-Context Examples for GPT-3?’ arXiv:2101.06804. Preprint, arXiv, January 17. https://doi.org/10.48550/arXiv.2101.06804.

Liu, Ryan, Jiayi Geng, Addison J. Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L. Griffiths. 2025. ‘Mind Your Step (by Step): Chain-of-Thought Can Reduce Performance on Tasks Where Thinking Makes Humans Worse’. arXiv:2410.21333. Preprint, arXiv, June 13. https://doi.org/10.48550/arXiv.2410.21333.

Lu, Albert, Hongxin Zhang, Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2023. ‘Bounding the Capabilities of Large Language Models in Open Text Generation with Prompt Constraints’. arXiv:2302.09185. Preprint, arXiv, February 17. https://doi.org/10.48550/arXiv.2302.09185.

Madaan, Aman, Niket Tandon, Prakhar Gupta, et al. 2023. ‘Self-Refine: Iterative Refinement with Self-Feedback’. arXiv:2303.17651. Preprint, arXiv, May 25. https://doi.org/10.48550/arXiv.2303.17651.

Mao, Yuetian, Junjie He, and Chunyang Chen. 2025. ‘From Prompts to Templates: A Systematic Prompt Template Analysis for Real-World LLMapps’. *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, June 23, 75–86. https://doi.org/10.1145/3696630.3728533.

Megahed, Fadel M., Ying-Ju Chen, L. Allision Jones-Farmer, Younghwa Lee, Jiawei Brooke Wang, and Inez M. Zwetsloot. 2025. ‘Reliable Decision Support with LLMs: A Framework for Evaluating Consistency in Binary Text Classification Applications’. arXiv:2505.14918. Preprint, arXiv, May 20. https://doi.org/10.48550/arXiv.2505.14918.

Mishra, Vishala, Ashish Sarraju, Neil M. Kalwani, and Joseph P. Dexter. 2024. ‘Evaluation of Prompts to Simplify Cardiovascular Disease Information Generated Using a Large Language Model: Cross-Sectional Study’. *Journal of Medical Internet Research* 26 (1): e55388. https://doi.org/10.2196/55388.

Molenaar, Stef R.A., Bram van de Put, Jessica S. Desport, Saer Samanipour, Ron A.H. Peters, and Bob W.J. Pirok. 2022. ‘Automated Feature Mining for Two-Dimensional Liquid Chromatography Applied to Polymers Enabled by Mass Remainder Analysis’. *Analytical Chemistry* 94 (14): 5599–607. https://doi.org/10.1021/acs.analchem.1c05336.

Niederer, Steven A., Joost Lumens, and Natalia A. Trayanova. 2019. ‘Computational Models in Cardiology’. *Nature Reviews Cardiology* 16 (2): 100–111. https://doi.org/10.1038/s41569-018-0104-y.

Oh, Jeong-seok, and Jay-yoon Lee. 2025. ‘Latent Self-Consistency for Reliable Majority-Set Selection in Short- and Long-Answer Reasoning’. arXiv:2508.18395. Preprint, arXiv, August 25. https://doi.org/10.48550/arXiv.2508.18395.

OpenAI. 2025a. *Codex CLI: Coding Agent for Local Development*. OpenAI, released. https://github.com/openai/codex.

OpenAI. 2025b. *OpenAI O3 and O4-Mini System Card*. Technical Report. OpenAI. https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf.

Peters, Uwe, and Benjamin Chin-Yee. 2025. ‘Generalization Bias in Large Language Model Summarization of Scientific Research’. *Royal Society Open Science* 12 (4): 241776. https://doi.org/10.1098/rsos.241776.

Polak, Maciej P., and Dane Morgan. 2024. ‘Extracting Accurate Materials Data from Research Papers with Conversational Language Models and Prompt Engineering’. *Nature Communications* 15 (1): 1569. https://doi.org/10.1038/s41467-024-45914-8.

Rajaraman, Nived, Jiantao Jiao, and Kannan Ramchandran. 2025. ‘Toward a Theory of Tokenization in LLMs’. arXiv:2404.08335. Preprint, arXiv, April 10. https://doi.org/10.48550/arXiv.2404.08335.

Rasal, Sumedh, and E. J. Hauer. 2024. ‘Navigating Complexity: Orchestrated Problem Solving with Multi-Agent LLMs’. arXiv:2402.16713. Version 1. Preprint, arXiv, February 26. https://doi.org/10.48550/arXiv.2402.16713.

Riegler, Michael A., Kristoffer Herland Hellton, Vajira Thambawita, and Hugo L. Hammer. 2025. ‘Using Large Language Models to Suggest Informative Prior Distributions in Bayesian Statistics’. arXiv:2506.21964. Preprint, arXiv, June 27. https://doi.org/10.48550/arXiv.2506.21964.

Ruan, Yixiang, Chenyin Lu, Ning Xu, et al. 2024. ‘An Automatic End-to-End Chemical Synthesis Development Platform Powered by Large Language Models’. *Nature Communications* 15 (1): 10160. https://doi.org/10.1038/s41467-024-54457-x.

Rudolph, Jürgen, Samson Tan, and Shannon Tan. 2023. ‘ChatGPT: Bullshit Spewer or the End of Traditional Assessments in Higher Education?’ *Journal of Applied Learning and Teaching* 6 (1): 342–63.

Schulhoff, Sander, Michael Ilie, Nishant Balepur, et al. 2025. ‘The Prompt Report: A Systematic Survey of Prompt Engineering Techniques’. arXiv:2406.06608. Preprint, arXiv, February 26. https://doi.org/10.48550/arXiv.2406.06608.

Shaikh, Omar, Hussein Mozannar, Gagan Bansal, Adam Fourney, and Eric Horvitz. 2025. ‘Navigating Rifts in Human-LLM Grounding: Study and Benchmark’. arXiv:2503.13975. Preprint, arXiv, June 1. https://doi.org/10.48550/arXiv.2503.13975.

Sharma, Aditi, Tejas Medapalli, Micaella Alexandrou, Emmanouil Brilakis, and Anand Prasad. 2024. 'Exploring the Role of ChatGPT in Cardiology: A Systematic Review of the Current Literature'. *Cureus*, ahead of print, April 24. https://doi.org/10.7759/cureus.58936.

Shinn, Noah, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. 'Reflexion: Language Agents with Verbal Reinforcement Learning'. arXiv:2303.11366. Preprint, arXiv, October 10. https://doi.org/10.48550/arXiv.2303.11366.

Silva, Viviane Torres da, Alexandre Rademaker, Krystelle Lionti, et al. 2024. 'Automated, LLM Enabled Extraction of Synthesis Details for Reticular Materials from Scientific Literature'. arXiv:2411.03484. Preprint, arXiv, November 5. https://doi.org/10.48550/arXiv.2411.03484.

Smith, Gabriel Reuben, Carolina Bello, Lalasia Bialic-Murphy, et al. 2024. 'Ten Simple Rules for Using Large Language Models in Science, Version 1.0'. *PLOS Computational Biology* 20 (1): e1011767. https://doi.org/10.1371/journal.pcbi.1011767.

Su, Hongjin, Jungo Kasai, Chen Henry Wu, et al. 2022. 'Selective Annotation Makes Language Models Better Few-Shot Learners'. arXiv:2209.01975. Preprint, arXiv, September 5. https://doi.org/10.48550/arXiv.2209.01975.

Sukherman, Lev, and Yetunde Folajimi. 2025. 'Enhancing Mathematical Reasoning in GPT-J Through Topic-Aware Prompt Engineering'. *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization* (New York, NY, USA), UMAP Adjunct '25, June 12, 388–92. https://doi.org/10.1145/3708319.3733807.

Sun, Yicheng, Qi Zhang, Jinsong Bao, Yuqian Lu, and Shimin Liu. 2024. 'Empowering Digital Twins with Large Language Models for Global Temporal Feature Learning'. *Journal of Manufacturing Systems* 74 (June): 83–99. https://doi.org/10.1016/j.jmsy.2024.02.015.

Taubenfeld, Amir, Tom Sheffer, Eran Ofek, et al. 2025. 'Confidence Improves Self-Consistency in LLMs'. *Findings of the Association for Computational Linguistics: ACL 2025*, 20090–111. https://doi.org/10.18653/v1/2025.findings-acl.1030.

Tewari, Jeevan, Benjamin W. Dahl, and Jeffrey J. Saucerman. 2025. 'Benchmarking of Signaling Networks Generated by Large Language Models'. Preprint, bioRxiv, July 29. https://doi.org/10.1101/2025.07.28.667217.

Toufiq, Mohammed, Darawan Rinchai, Eleonore Bettacchioli, et al. 2023. 'Harnessing Large Language Models (LLMs) for Candidate Gene Prioritization and Selection'. *Journal of Translational Medicine* 21 (1): 728. https://doi.org/10.1186/s12967-023-04576-8.

Van Noorden, Richard, and Jeffrey M. Perkel. 2023. ‘AI and Science: What 1,600 Researchers Think’. *Nature* 621 (7980): 672–75. https://doi.org/10.1038/d41586-023-02980-0.

Van Veen, Dave, Cara Van Uden, Louis Blankemeier, et al. 2024. ‘Adapted Large Language Models Can Outperform Medical Experts in Clinical Text Summarization’. *Nature Medicine* 30 (4): 1134–42. https://doi.org/10.1038/s41591-024-02855-5.

Walters, William H., and Esther Isabelle Wilder. 2023. ‘Fabrication and Errors in the Bibliographic Citations Generated by ChatGPT’. *Scientific Reports* 13 (1): 14045. https://doi.org/10.1038/s41598-023-41032-5.

Wang, Lei, Wanyu Xu, Yihuai Lan, et al. 2023. ‘Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models’. arXiv:2305.04091. Preprint, arXiv, May 26. https://doi.org/10.48550/arXiv.2305.04091.

Wang, Song, Zihan Chen, Chengshuai Shi, Cong Shen, and Jundong Li. 2024. ‘Mixture of Demonstrations for In-Context Learning’. *Advances in Neural Information Processing Systems* 37 (December): 88091–116.

Wang, Xuezhi, Jason Wei, Dale Schuurmans, et al. 2023. ‘Self-Consistency Improves Chain of Thought Reasoning in Language Models’. arXiv:2203.11171. Preprint, arXiv, March 7. https://doi.org/10.48550/arXiv.2203.11171.

Wei, Jason, Maarten Bosma, Vincent Y. Zhao, et al. 2022. ‘Finetuned Language Models Are Zero-Shot Learners’. arXiv:2109.01652. Preprint, arXiv, February 8. https://doi.org/10.48550/arXiv.2109.01652.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, et al. 2022. ‘Chain-of-Thought Prompting Elicits Reasoning in Large Language Models’. *Advances in Neural Information Processing Systems* 35: 24824–37.

Wei, Jerry, Jason Wei, Yi Tay, et al. 2023. ‘Larger Language Models Do In-Context Learning Differently’. arXiv:2303.03846. Preprint, arXiv, March 8. https://doi.org/10.48550/arXiv.2303.03846.

Wong, Ryan, Jiawei Wang, Junjie Zhao, et al. 2025. ‘WideSearch: Benchmarking Agentic Broad Info-Seeking’. arXiv:2508.07999. Preprint, arXiv, August 28. https://doi.org/10.48550/arXiv.2508.07999.

Yan, Tianqiang, Ziqiao Lin, Lin Zhang, Zhenglong Sun, and Yuan Gao. 2025. ‘Entrospect: Information-Theoretic Self-Reflection Elicits Better Response Refinement of Small Language Models’. In *Findings of the Association for Computational Linguistics: ACL 2025*, edited by Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar. Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.findings-acl.1261.

Ye, Qinyuan, Maxamed Axmed, Reid Pryzant, and Fereshte Khani. 2024. 'Prompt Engineering a Prompt Engineer'. arXiv:2311.05661. Preprint, arXiv, July 3. https://doi.org/10.48550/arXiv.2311.05661.

Yoo, Kang Min, Junyeob Kim, Hyuhng Joon Kim, et al. 2022. 'Ground-Truth Labels Matter: A Deeper Look into Input-Label Demonstrations'. arXiv:2205.12685. Preprint, arXiv, October 24. https://doi.org/10.48550/arXiv.2205.12685.

Zamfirescu-Pereira, J.D., Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. 'Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts'. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, April 19, 1–21. https://doi.org/10.1145/3544548.3581388.

Zhang, Zhuosheng, Yao Yao, Aston Zhang, et al. 2025. 'Igniting Language Intelligence: The Hitchhiker's Guide from Chain-of-Thought Reasoning to Language Agents'. *ACM Computing Surveys* 57 (8): 1–39. https://doi.org/10.1145/3719341.

Zhang, Zhuosheng, Aston Zhang, Mu Li, and Alex Smola. 2022. 'Automatic Chain of Thought Prompting in Large Language Models'. arXiv:2210.03493. Preprint, arXiv, October 7. https://doi.org/10.48550/arXiv.2210.03493.

Zheng, Mingqian, Jiaxin Pei, Lajanugen Logeswaran, Moontae Lee, and David Jurgens. 2024. 'When "A Helpful Assistant" Is Not Really Helpful: Personas in System Prompts Do Not Improve Performances of Large Language Models'. arXiv:2311.10054. Version 3. Preprint, arXiv, October 9. https://doi.org/10.48550/arXiv.2311.10054.

Zheng, Yizhen, Huan Yee Koh, Maddie Yang, et al. 2024. 'Large Language Models in Drug Discovery and Development: From Disease Mechanisms to Clinical Trials'. arXiv:2409.04481. Preprint, arXiv, September 6. https://doi.org/10.48550/arXiv.2409.04481.

Zheng, Zhiling, Oufan Zhang, Christian Borgs, Jennifer T. Chayes, and Omar M. Yaghi. 2023. 'ChatGPT Chemistry Assistant for Text Mining and the Prediction of MOF Synthesis'. *Journal of the American Chemical Society* 145 (32): 18048–62. https://doi.org/10.1021/jacs.3c05819.

Zheng, Zhiling, Oufan Zhang, Ha L. Nguyen, et al. 2023. 'ChatGPT Research Group for Optimizing the Crystallinity of MOFs and COFs'. *ACS Central Science* 9 (11): 2161–70. https://doi.org/10.1021/acscentsci.3c01087.

Zhou, Denny, Nathanael Schärli, Le Hou, et al. 2023. 'Least-to-Most Prompting Enables Complex Reasoning in Large Language Models'. arXiv:2205.10625. Preprint, arXiv, April 16. https://doi.org/10.48550/arXiv.2205.10625.