https://doi.org/10.5281/zenodo.19177102

Large Language Models for Science and Research: A Practical Guide

James M Dewar
Department of Biochemistry, Vanderbilt University School of Medicine, Nashville, TN 37232, United States.
Correspondence: james.dewar@vanderbilt.edu

Abstract

Large language models emerged as mainstream tools in 2022, but reasoning-capable models released in late 2024 through 2025 represented a step change in scientific utility, achieving PhD-level performance on domain-specific benchmarks and reliably completing autonomous tasks requiring hours of human effort. Yet the unbounded nature of these systems, combined with legitimate privacy concerns and rapidly shifting capabilities, makes it difficult for researchers to identify an access point and establish safe practices. This guide provides a practical framework for engaging with reasoning LLMs across scientific disciplines. I first describe what these models are and how they work, including core concepts (statelessness, context windows, prompt engineering) that inform productive use. I then introduce the "dual stance," a mental framework for treating LLM outputs simultaneously as high-quality intellectual contributions and as unverified claims requiring systematic checking. The guide presents a five-tier privacy framework for data protection, discusses legal constraints on using published literature with LLMs, and details verification practices including chunking, grounding through retrieval-augmented generation, and two-level fact-checking. Research applications (manuscript summarization, on-demand literature reviews, scientific writing) and education applications (interactive journal clubs, oral exam preparation) are illustrated with worked prompts. I address risks including hallucination, cognitive bypass, and degradation of scholarly outputs, proposing specific mitigations for each.

1. Introduction

Large language models (Figure 1) emerged as mainstream tools in 2022, but reasoning-capable models ('reasoning LLMs') released in late 2024 through 2025 represented a step change in scientific utility (Dewar, 2025). On GPQA Diamond, a benchmark of PhD-level science questions, earlier models scored approximately 40% in late 2023; reasoning models established a floor of approximately 70%, corresponding to human PhD-level performance, with frontier models now exceeding 90% (Table 1), which approaches the upper limit of what we can conceive of measuring. METR's measurements of autonomous task completion show a similar trajectory: models progressed from reliably completing tasks requiring approximately 10 minutes of human effort to tasks requiring approximately 12 hours (Table 1). This

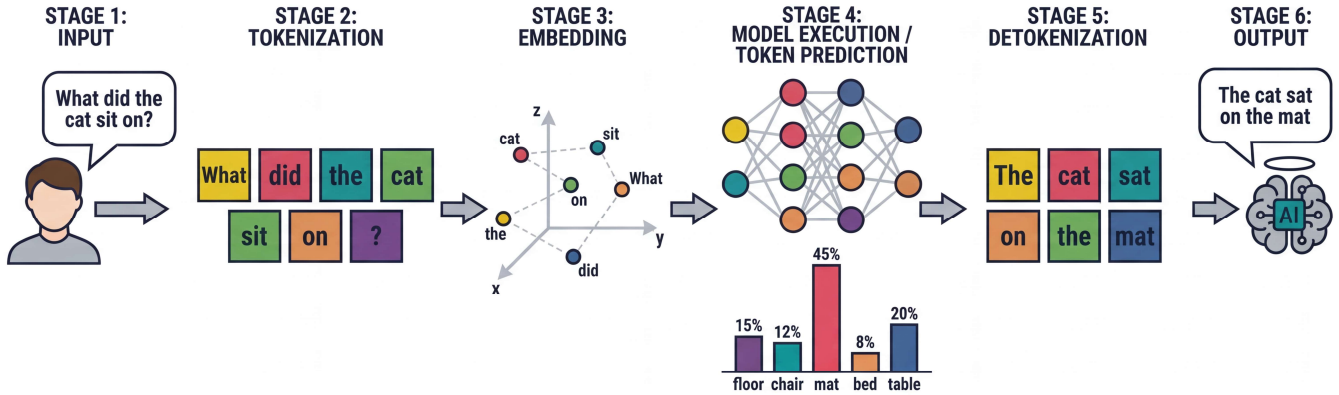

Figure 1: How large language models process text. User input (Stage 1) is broken into discrete tokens (Stage 2), which are mapped to positions in high-dimensional vector space where semantically related words cluster together (Stage 3). The model processes these embeddings through a neural network to predict the probability distribution over possible next tokens (Stage 4), selecting "mat" as the most likely continuation (45%). The predicted tokens are converted back to readable text (Stage 5) and returned to the user (Stage 6). This process repeats for each token in the output sequence, with the model generating one token at a time.

				GPQA Diamond	
Company	**Model**	**Release Date**	**Task Duration**	**Indicated Model**	**Frontier**
OpenAI	GPT 5.2 High	Dec 2025	5 hrs 52 min	88%	95%
Anthropic	Claude Opus 4.6	Feb 2026	11 hrs 59 min	91%	91%
Google	Gemini 3.0 Pro	Nov 2025	3 hrs 44 min	93%	94%

Table 1: Performance benchmarks for current frontier reasoning models. GPQA Diamond (Rein *et al* 2023, Epoch AI 2026) measures accuracy on PhD-level science questions where random guessing yields 25% and PhD-level experts score 69.7%; "Indicated Model" shows performance of the specific model listed, while "Frontier" shows the highest score achieved by any model from that company at time of writing. Task Duration (Kwa *et al* 2025, METR 2026) indicates the duration of autonomous tasks the model can reliably complete. Models listed are the most advanced from OpenAI, Anthropic, and Google with both task duration (METR) and GPQA diamond values available. Data are current as of March 22, 2026. This table reports the most advanced models with both GPQA Diamond and METR task duration benchmarks available.

trajectory aligns with my experience that these tools now match expert-level performance for many knowledge tasks. Yet the unbounded nature of these systems, combined with legitimate privacy concerns and rapidly shifting capabilities, makes it daunting to identify an access point and establish safe practices.

This guide helps researchers engage with reasoning LLMs effectively: understanding what they are, accessing them safely, applying them to research and education, and mitigating common failure modes. Reasoning LLMs are capable of a much broader range of tasks than those covered here, including writing and debugging code, interpreting documents in foreign languages, analyzing complex datasets, and generating hypotheses. These applications are beyond the scope of this introductory guide but will be addressed in the forthcoming Advanced Techniques manuscript. My background is in biomedical research, so the examples and use cases I provide draw from that domain, but the principles and workflows should apply broadly across scientific disciplines. A forthcoming *Advanced Techniques* manuscript will provide detailed treatment of parameter control, verification workflows, and frontier applications.

2. What Are Reasoning LLMs?

Large Language Models (LLMs)

Large language models are built through a two-stage process. First, during pre-training, the model ingests massive amounts of text data, often encompassing much of the publicly available internet, digitized books, scientific literature, and code repositories. Through this exposure, the model develops a compressed representation of patterns, facts, and relationships present in the training data. This latent knowledge, encoded in the model's parameters, constitutes the only persistent 'memory' an LLM possesses. Second, during inference (when you interact with the model), your input is converted into numerical tokens, embedded in high-dimensional vector space, and processed to predict the most probable next tokens given the input sequence (Figure 1). The model draws on its latent training knowledge to generate responses. Users need not understand these mechanics in detail to use the tools productively. Readers familiar with LLM basics can skip to Section 3, which introduces interaction frameworks specific to reasoning models.

One practical consequence of statistical prediction is that outputs can be tuned. The probability distribution over next tokens (Figure 1, Stage 4) can be manipulated through parameters such as temperature, which controls how deterministic or variable the output is, and top-p, which constrains the pool of candidate tokens. A forthcoming Advanced Techniques manuscript will cover these and other parameters in detail.

Reasoning models

Reasoning models are a specific class of LLMs that includes OpenAI's o-series and GPT5, Anthropic's Claude Opus, and Google's Gemini Pro. No definitive distinction separates reasoning models from other LLMs, but 'reasoning' models generally produce an extended text output that is not shown to the user, effectively an inner monologue that allows them to 'think' through problems before responding. We do not know precisely what AI companies did differently to produce reasoning models. However, based on the one publicly documented case (DeepSeek R1; Guo et al, 2025) and statements from AI companies, we can infer that these models were likely trained with increased emphasis on solving problems with defined answers rather than responding to subjective human preference. For DeepSeek R1, the 'reasoning' behavior emerged as a consequence of this training

objective, and this likely applies to other models as well. Regardless of exactly how reasoning capabilities arose, the practical implication is that these models perform at or above the level of PhD scientists and can reliably complete tasks that would take humans hours, as noted above.

Prompts and prompt engineering.

A prompt is a set of instructions provided to an LLM in natural language. The process of constructing effective prompts is termed prompt engineering. While LLMs interpret a wide range of input formats, a general structure improves output quality. The RTIEC framework organizes prompts into five components: Role (what expertise the

A

```
**ROLE** (Optional)
What persona the LLM should adopt (e.g. "An expert biochemist").

**TASK**
What you want the LLM to do (e.g. "Summarize a manuscript").

**CONSTRAINTS**
Limitations on how the LLM's output should be presented (e.g. "<200 words").

**INFORMATION**
Information needed for the task (e.g. pasted text of a manuscript).

**EXAMPLES** (Optional)
Demonstrations of what the output should look like relative to the desired input. These can either be specific
to the information provided or highly generic.
```

B

```
**ROLE**
You are a senior molecular biologist and experienced scientific editor specializing in genome-maintenance
pathways. Your audience is the multidisciplinary readership of *eLife*.

**TASK**
Generate a concise, 250-word abstract that accurately summarizes the attached manuscript for submission to
*eLife*. Emphasize significance, methodological innovation, principal findings, and mechanistic insight.

**CONSTRAINTS**
* Length ≤ 250 words
* One structured paragraph—no subheadings
* First sentence: clear contextual hook (≤ 35 words)
* Final sentence: explicit statement of impact for the field
* Preserve all gene/protein symbols and numeric values exactly as provided
* Do **not** introduce data or citations absent from the manuscript
* Provide only the final abstract

**INFORMATION**
--- BEGIN MANUSCRIPT ---
[Paste full manuscript or Results + Discussion sections here]
--- END MANUSCRIPT ---

**EXAMPLES**
Input excerpt:
“Deletion of **YFG** resulted in delayed S-phase progression and accumulation of γH2AX foci. Complementation
with FLAG-YFG restored replication-fork velocity and suppressed DNA double-strand-break formation.”

**Desired abstract fragment:**
“Loss of **YFG** compromises replication-fork integrity, inducing genome-wide DNA-damage signaling. Re-
expression of epitope-tagged **YFG** rescues fork velocity and attenuates double-strand-break accumulation,
establishing **YFG** as a pivotal guardian of DNA replication and repair.”
```

C

```
You are an expert prompt writer. Write a generic prompt to illustrate prompt engineering for the biological
sciences. The format should be role/task/constraints/information/examples. The prompt should be aimed at
summarizing a manuscript to generate an abstract for submission to *Cell*. Examples should refer to the role of
the gene/protein YFG in DNA replication and repair.
```

Figure 2: Prompt structure and self-prompting. (A) The RTIEC framework defines five components of an effective prompt: Role (the persona the model should adopt), Task (what you want accomplished), Information (content needed for the task), Examples (demonstrations of desired output), and Constraints (limitations on format or content). Role and Examples are optional but improve output quality for complex tasks. (B) A complete prompt following the RTIEC framework for generating a manuscript abstract. Each component is labeled and populated with specific instructions. (C) Self-prompting: directing the model to generate its own prompt. This simpler request produced the complete prompt shown in panel B. Self-prompting leverages the model's knowledge to populate framework components, making prompt engineering accessible without requiring users to construct detailed prompts manually.

model should adopt), Task (what you want it to do), Information (data or context it needs), Examples (demonstrations of desired output format or style), and Constraints (limitations on length, format, or scope) (Figure 2A). Role and Examples are optional; Task, Information, and Constraints form the minimum effective prompt. A fully specified example is shown in Figure 2B. The underlying principle is that LLMs generate better outputs when given explicit structure rather than open-ended requests. "Summarize this paper" will produce a generic summary; "Summarize this paper in 200 words, emphasizing the experimental approach and identifying one unstated assumption" will produce output tailored to your analytical needs. The difference is not the model's capability but the specificity of the instruction.

Constructing detailed prompts can be simplified through self-prompting: directing the LLM to generate its own prompt. Rather than writing a fully specified RTIEC prompt from scratch, you describe what you want in plain language and instruct the model to produce the detailed prompt for you. The prompt shown in Figure 2B was generated by the simpler request in Figure 2C. Self-prompting works because the model draws on its training knowledge to populate roles, constraints, and examples that you might not think to specify. The simpler request still implicitly follows the RTIEC structure (it specifies a role for the model as a prompt writer, a task, and constraints), but the user need not consciously apply the framework. Self-prompting (also known as ‘meta-prompting’) is a practical starting point; as familiarity develops, direct prompt writing provides finer control. A forthcoming *Advanced Techniques* manuscript will cover advanced prompting techniques including structured outputs, chain-of-thought reasoning, and platform-specific optimization.

Context window

The context window defines the maximum information a model can process in a single interaction, measured in tokens. A token is a word, subword, or character that the model treats as a single unit; on average, one token equals approximately 0.75 words. Current frontier models advertise context windows of 128,000 tokens (ChatGPT, Claude) to 1 million tokens (Gemini). These limits determine how much information actually influences model outputs. Context is consumed by two sources: uploaded files and conversation history. Understanding how each consumes context is essential for effective use.

Training data vs. in-context information

LLMs draw on two distinct sources of information. Training data functions as long-term knowledge: during pre-training, the model ingested a massive text corpus and developed a compressed representation of its contents. This latent knowledge persists across all conversations and constitutes the only information the model carries independently. Current frontier models have training data through roughly late 2025, and anything published before that cutoff could be represented in the model's knowledge, likely including much of the published scientific literature. However, representation in training data does not guarantee reliable recall, because these models encode knowledge as statistical patterns rather than stored facts. The most practical way to test what a model knows from training is to query it with external tools disabled, so you can observe what it retrieves from training knowledge alone.

In-context information is the second source: the stream of data the model receives each time it generates a response, including the conversation history, documents you upload, and results from tools like web search. In-context information is ephemeral: it exists only for the duration of a single response and is lost unless the interface reintroduces it in the next turn. The model's apparent memory of your name from earlier in a conversation is reconstructed each time from recycled conversation history, not retained internally. Reasoning models add a further constraint: internal "reasoning tokens" generated during deliberation are not visible to users but consume context window space, and are discarded after each response rather than recycled into subsequent turns. Together, these constraints mean that long conversations accumulate context rapidly, leaving less room for new information, and that the model's recall of earlier turns is reconstructed rather than retained.

All major platforms now offer integrated web search tools that allow the model to retrieve current information from the internet during a conversation, effectively extending its knowledge beyond the training cutoff. Web search partially bridges the gap between the two information layers, but introduces its own verification challenges: the model selects which sources to consult and how to synthesize them, and users cannot always determine which claims derive from training knowledge versus retrieved content. Tool outputs, including web search results, consume context window space, so tool-augmented conversations fill context more rapidly than text-only interactions. Models can also use code execution, file analysis, and third-party service

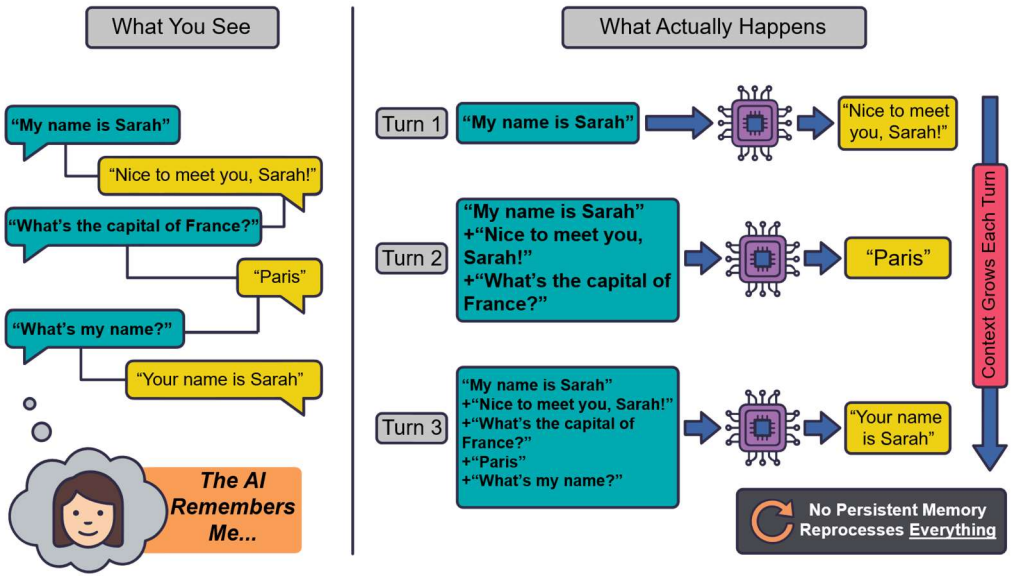

Figure 3: LLMs are stateless despite the appearance of memory. **Left:** The chat interface suggests the model remembers earlier exchanges, as when it recalls "Your name is Sarah" from Turn 1**. Right:** What actually happens is that each turn reprocesses the entire conversation history as a single input. Turn 2 includes Turn 1's exchange plus the new query; Turn 3 includes everything from Turns 1 and 2. The model has no persistent memory between turns; context accumulates in the input, not in the model. This is why long conversations consume context window capacity and why opening a new chat creates a genuinely independent context.

connections; these capabilities are beyond the scope of this introductory guide and will be addressed in the forthcoming Advanced Techniques manuscript.

Current frontier models were trained on corpora containing both copyrighted and non-copyrighted material. This practice frequently raises concerns among researchers considering whether to adopt these tools. The emerging legal consensus suggests that training on copyrighted works constitutes fair use provided the material was lawfully acquired; a June 2025 federal ruling in Bartz v. Anthropic described such use as "quintessentially transformative," though several cases remain pending. Regardless of how these challenges are resolved, the legal risk for individual users is low: all three major providers offer intellectual property indemnification for paid customers, meaning the provider will defend users and cover costs if a third party sues over model outputs, provided the user has not violated the platform's terms of service. The more immediate concern for researchers is whether their own interactions and uploaded materials may become part of future training data; this issue is addressed in the five-tier privacy framework in Section 4.

Conversation mechanics

Most interactions with an LLM appear to involve a simple back-and-forth: you send a message, the model responds, you reply, and so on. However, the way this works is non-intuitive. LLMs are stateless, meaning each time you submit a prompt, the model has no memory of prior interactions and no persistent internal state. What actually happens is that the interface reconstructs the entire conversation history and feeds it back to the model as a single input before each new response (Figure 3). The model reads everything from the beginning, your first message, its first response, your second message, its second response, and so on, up to your latest input, and only then generates its next output. Each turn of the conversation is therefore a completely new inference event where the model processes all prior turns as if encountering them for the first time. While the appearance of independent text inputs on the screen might give the appearance that they can be deleted or ignored, this is simply not the case because they must be re-processed by the model each time you send a message. The conversation analogy is therefore apt because you cannot tell the model to ignore a prior statement any more than you can tell a colleague to ignore a comment you made previously; it has already happened.

One aspect of the conversational analogy does break down: attention mechanisms in transformer architectures do not weight all tokens equally. Models tend to attend most strongly to tokens at the beginning and end of the context window, while tokens in the middle receive less attention, a pattern documented as the "lost in the middle" effect (Liu et al., 2024). Information presented at the start or end of a conversation, or at the beginning or end of an uploaded document, may disproportionately shape outputs relative to equally relevant information buried in the middle. The practical implication is that important

Company	Model	Release Date	Context Window	Monthly Cost (USD)	Training Data Opt Out
OpenAI	GPT 5.4 High (Pro)	Mar 2026	128 k (1 M)	20 (200)	Yes
Anthropic	Claude Opus 4.6	Feb 2026	200k*	20	Yes
Google	Gemini 3.1 Pro	Feb 2026	1 M	7.99	No‡

Table 2: Practical specifications for current frontier reasoning models. Context window indicates maximum tokens processed per interaction; monthly cost reflects consumer subscription pricing for Claude Pro, ChatGPT Plus (Pro in parentheses), and Google AI Plus. Training data opt-out indicates whether users can prevent their conversations from being used to train future models. *Claude Opus 4.6 supports 1M context via API but the desktop interface is limited to 200k tokens. ‡Gemini 3.1 Pro allows conversations to be excluded from training only via "temporary chat" mode, which does not save conversation history; there is no option to retain conversations while opting out of training. Models listed are the latest from OpenAI, Anthropic, and Google. Data are accurate as of March 22, 2026. This table reports practical specifications for the latest consumer-facing models, which differ from the benchmark-selected models shown in Table 1.

instructions or content should be placed at the beginning or end of your input, not in the center of a long document or conversation.

Because each turn conditions on all preceding tokens, asking a model to independently evaluate its own prior output within the same conversation is unreliable: the shared context biases the assessment toward consistency with what came before. Opening a new chat window creates a genuinely independent context, analogous to a biological replicate in experimental design. Each subsequent turn within the same conversation is more analogous to sampling from the same reaction: the outputs are not independent because they share the same accumulated context. If you want outputs unbiased by earlier parts of a conversation, or if you want to test whether a particular response is robust rather than an artifact of conversational drift, open a new chat window and pose your query fresh.

Excess Context and Files

Platforms differ in how they handle content relative to context window limits. Three distinct scenarios arise in practice, and documentation for each remains incomplete.

Excess context at upload. When uploaded content exceeds the context window at the point of upload, platforms diverge in behavior (Figure 4). ChatGPT is the most explicit: ChatGPT Enterprise places up to 110,000 tokens directly in context, taking portions from each document, then indexes the remainder in a searchable vector store (OpenAI, n.d.). Claude refuses content that exceeds the context window, making the limitation visible to the user. Gemini attempts to fit everything into its native context window and likely discards data invisibly when capacity is exceeded (Google, n.d.). Users should not assume that content exceeding documented capacity is fully processed.

File uploads and persistent access. Platforms increasingly offer mechanisms for files to persist beyond the conversation in which they were uploaded. OpenAI's recently launched Library feature (March 2026) automatically saves uploaded files to the user's account and allows them to be attached to new conversations without re-uploading, ensuring that files remain available across sessions. How platforms handle queries against previously uploaded files within a conversation is less well documented. In author testing (March 2026), Claude used its code execution tools to search uploaded files and extract relevant snippets on demand rather than holding the entire file in context.

Conversational overflow. Long conversations can exceed the context window through accumulated conversation history rather than file uploads. Claude addresses this through automatic summarization of earlier messages, preserving the full chat history for reference while freeing context space (Anthropic, n.d.; requires code execution to be enabled). Documentation for how other platforms handle conversational overflow is limited.

Implication for scientists and researchers. Manuscripts average 20,000 BPE tokens with a median around 17,000 in my experience, and may be uniquely suited for large-scale LLM analysis. Abstracts appear first and summarize entire papers in approximately 250 words. ChatGPT's stuffing algorithm prioritizes document beginnings, meaning abstracts are likely retained in direct context. Researchers could therefore upload 200-300 articles with all abstracts processed directly and body text available for

retrieval. This structure makes scientific literature more tractable for LLM analysis than unstructured document collections.

Persistent Context Across Chats

Major platforms now offer multiple mechanisms for maintaining context beyond a single conversation, layered from lightweight personalization to dedicated workspaces.

User-specified instructions. Claude, ChatGPT, and Gemini allow users to define persistent instructions that shape model behavior across conversations. These typically include information about the user's role, preferred response style, and standing constraints. The platform invisibly appends these instructions to every conversation, customizing the model's responses to user preferences.

Model-generated memories. Claude and ChatGPT can generate and store facts learned during conversations and apply them to future interactions, accumulating information about the user over time (such as your name, role, or recurring projects). How these memories are accessed varies between platforms: Claude adds them to each conversation automatically, while ChatGPT searches them on demand. Users can view, edit, and delete individual memories and are advised to do so periodically to ensure the model receives accurate and relevant information.

Cross-conversation search. Claude, ChatGPT, and Gemini support searching across prior conversations, enabling knowledge to accumulate over time. This allows users to retrieve and build on earlier work without re-uploading materials or re-explaining context. However, it also means that any conversation can be influenced by prior exchanges. Users who require truly independent contexts are advised to disable this feature.

Projects. Claude and ChatGPT offer project-based organization where documents and custom instructions persist across conversations within a designated workspace. This allows different sets of user-specified instructions and reference materials to be deployed for different tasks, selected by switching between projects.

A forthcoming Advanced Techniques manuscript will cover strategies for managing these different levels of persistent context across platforms.

3. How to Interact

Productive use of LLMs requires a mental framework I term the "dual stance": treat the model as a sophisticated peer and colleague capable of high-level intellectual output while simultaneously recognizing that anything it generates might be wrong or fabricated. This stance enables eliciting valuable outputs while maintaining sufficient skepticism to catch errors.

Conceptual models operationalizing the dual stance

Three frameworks help translate this stance into practice:

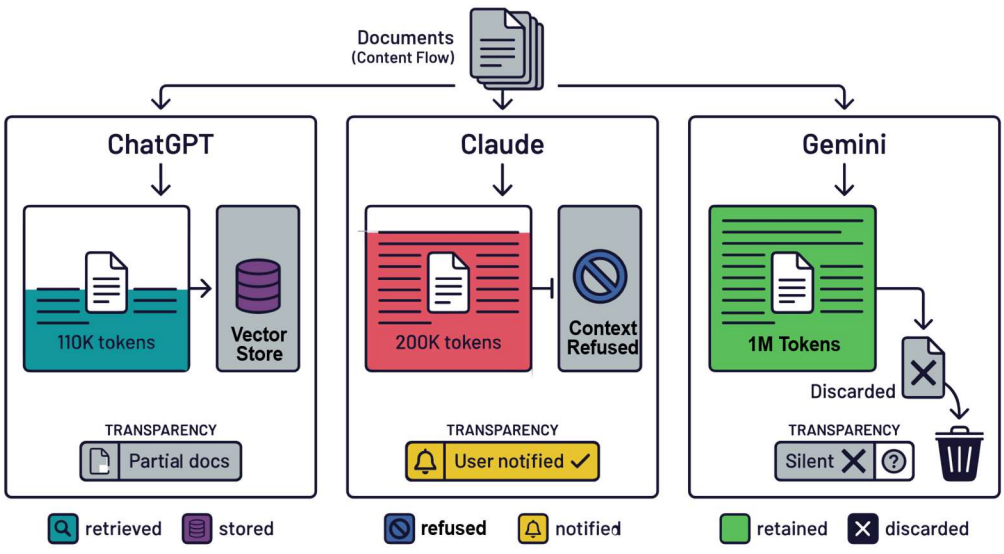

Figure 4: How major platforms handle content that exceeds context window capacity. ChatGPT places up to 110K tokens in direct context and indexes the remainder in a private vector store for retrieval; the user is not notified of the split (OpenAI, n.d.). Claude refuses to accept content that exceeds the context window, making the limitation explicit to the user. Gemini attempts to fit content into its 1M token native context; content exceeding this limit is likely discarded silently with no user notification (Google, n.d.). Separate mechanisms govern how platforms handle files that have been accepted but are queried later, and how platforms manage context overflow from long conversations; these vary by platform and are less well documented (see text).

Heuristic proxy: Frame LLM output as a heuristic proxy rather than an authoritative answer. The term "heuristic" deliberately underscores that outputs are probabilistic approximations, not deterministic conclusions. The term "proxy" clarifies that the generation stands in for a standard academic product. An LLM-generated literature summary, for instance, serves as a heuristic proxy for a formal human-authored review. This framing positions outputs as efficient starting points requiring verification, not finished products.

Relevance density: Many colleagues dismiss LLMs as unreliable due to hallucination. Yet I find their outputs exceptionally useful. This paradox resolves when one considers relevance density rather than only factual accuracy. A typical authoritative review, while highly accurate, often has low relevance density for a specific query; perhaps only 10-25% of its content directly addresses your question. An LLM-generated output, by contrast, can be nearly 100% relevant to your prompt while having factual accuracy that in my experience falls in the range of 80-98% depending on the task. This high relevance density is immensely valuable. The diminished accuracy is not disqualifying because rigorous fact-checking is required for any source. The primary strength of reasoning LLMs is their ability to rapidly generate text with high relevance density.

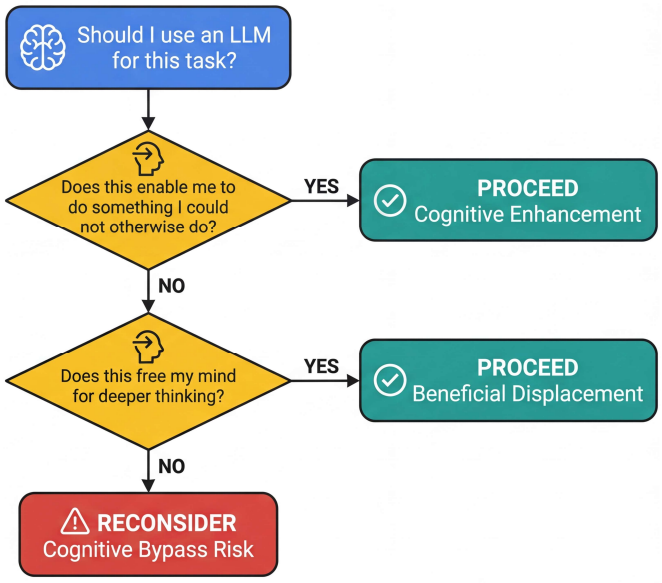

Figure 5: The two-question test for appropriate LLM use. Before engaging an LLM, users should ask whether the tool enables something they could not otherwise do (cognitive enhancement) or frees mental resources for deeper thinking elsewhere (beneficial displacement). A "yes" to either question justifies proceeding. If both answers are "no," the interaction risks cognitive bypass: producing output without cognitive engagement, which may erode capabilities over time. The test requires honest self-assessment and should be applied before, not after, LLM use.

Biological system analogy: Scientists may find it helpful to conceptualize LLMs as biological systems rather than deterministic software. Like biological systems, LLMs exhibit inherent stochasticity: the same input does not guarantee the same output. They show context-dependence: outputs vary based on what preceded them in the conversation, just as cellular responses depend on prior signaling history. They display saturation and diminishing returns: additional context or instruction beyond a certain point may not improve outputs and can degrade them. And they require replication for confidence: a single output, like a single experimental replicate, provides limited evidence. Opening a new chat window is analogous to running a biological replicate; subsequent turns within the same conversation are analogous to technical replicates or repeated sampling from the same reaction. This framing helps calibrate expectations: a biologist would not trust a single Western blot, and you should not trust a single LLM output for consequential decisions.

The two-question test

Before using an LLM, ask two questions (Figure 5). First: does this tool enable me to do something I could not otherwise do? Second: does it enable me to think more deeply about my current task or another? Proceed when at least one answer is yes. When neither applies, LLM use may represent cognition bypass rather than enhancement. This distinction matters because bypass erodes capabilities while enhancement builds them. Applying this test consistently prevents habitual offloading of cognitive work that would otherwise strengthen expertise.

Cognitive enhancement occurs when the LLM enables analysis otherwise impossible or impractical. Synthesizing 50 papers in an afternoon qualifies. Generating functional code without programming expertise qualifies. Rapidly iterating on prose that would take days qualifies. The LLM does something you genuinely cannot do, or cannot do at required scale. The cognitive work remains yours; the tool amplifies your capacity.

Beneficial cognitive displacement occurs when the LLM handles a task you could do yourself. The benefit comes

from freeing cognitive resources for higher-value thinking. Drafting routine correspondence, formatting references, or generating first-pass outlines can be worthwhile. The condition is that freed time enables deeper engagement with experimental design or interpretation. If you use the LLM simply because it is faster without redirecting saved time, the interaction may erode capabilities.

Sycophancy

LLMs exhibit sycophancy: a tendency to align responses with user preferences at the expense of accuracy. This behavior emerges from reinforcement learning with human feedback (RLHF), which rewards outputs humans rate highly. Humans implicitly prefer responses matching their views, and preference models inherit this bias (Sharma et al., 2024). Both model scaling and instruction tuning increase sycophancy; models will agree with objectively incorrect arithmetic if users express agreement (Wei et al., 2023). The most effective mitigation is post-training with synthetic data that encourages robustness to user opinions (Wei et al., 2023). This is a model-level intervention users cannot control, but understanding the mechanism informs user-level strategies.

At the user level, reducing social context improves accuracy. User expressions of certainty decrease accuracy: epistemic markers like "I'm sure it's..." reduced accuracy by 7% compared to low-certainty expressions (Zhou et al., 2023). Politeness increases compliance with problematic requests (Vinay et al., 2025). When models cannot infer what users want to hear, they default to more accurate answers (Cheng et al., 2025). Third-person framing ("A researcher claims..." rather than "I believe...") reduces sycophancy by removing social stakes. Practical recommendations: withhold opinions before asking factual questions, use neutral rather than assertive framing, and avoid expressing certainty about answers you are asking the model to verify.

4. Accessing LLMs

Reasoning models can be accessed through several primary routes. The most common are web interfaces from providers such as OpenAI (ChatGPT), Google (Gemini), and Anthropic (Claude), which allow direct interaction via text, files, or audio. Application Programming Interfaces (APIs) enable programmatic use with fine-grained parameter control and are typically pay-per-use, providing an economical route to advanced capabilities. Many providers also offer API-level controls through simplified web interfaces (OpenAI Playground, Google AI Studio, Anthropic Console), making advanced customization accessible without programming. Finally, models can be installed locally via tools like Ollama or LM Studio, providing complete data privacy by operating offline, though these local models currently lag frontier capabilities.

Five-tier privacy framework

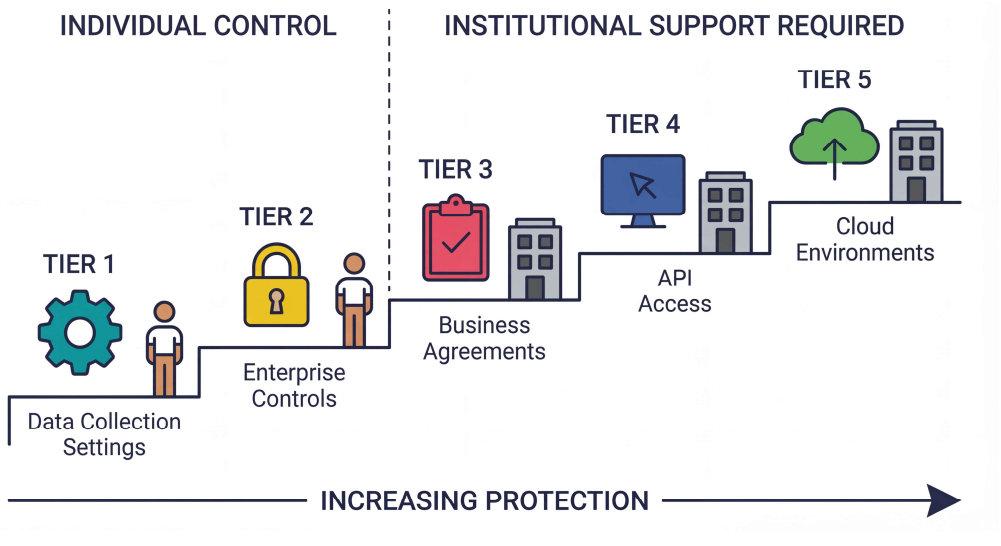

Figure 6: Five-tier framework for data privacy protection when using LLM services. Tiers 1-2 can be implemented through individual action: adjusting data collection settings (Tier 1) or subscribing to enterprise plans with additional controls (Tier 2). Tiers 3-5 require institutional support: negotiated business agreements with legal enforceability (Tier 3), API access with minimal data retention (Tier 4), or institutional cloud environments where data never touches the model provider's infrastructure (Tier 5). Higher tiers provide stronger protection but involve tradeoffs in cost, complexity, and accessibility. The tiers are cumulative rather than mutually exclusive. This framework assumes a typical researcher without command-line experience. Researchers comfortable with programming interfaces can access Tier 4 (API) independently, and can run open-source models locally for maximum data isolation. Conversely, while individual researchers can subscribe to some Tier 2 business plans independently, the full range of enterprise controls (third-party auditing, institutional access management) benefits substantially from institutional procurement and IT support.

Data privacy with third-party LLM services requires understanding multiple protection layers (Figure 6):

1. *Data collection settings.* Consumer plans typically opt users into training data collection by default; business and enterprise plans often opt out or block collection entirely. This determines whether the model maker can harvest your interactions for training future models.
2. *Enterprise and commercial controls.* Business-tier services add third-party auditing, access controls, and data separation to prevent leakage between users. Individual researchers can subscribe to some business-tier plans independently, but full enterprise controls typically require institutional procurement.
3. *Business agreements.* Contractual constraints provide legal teeth governing what model makers can and cannot do with your data.
4. *API access.* Direct API use typically involves minimal data retention and represents the closest individual users can get to high privacy without institutional support. Researchers comfortable with programmatic interfaces can implement this tier independently; others may need institutional assistance.
5. *Cloud environments.* Running models on platforms like AWS Bedrock or Microsoft Azure removes your data entirely from model makers' infrastructure, placing it under the cloud provider's data governance.

This framework is oriented toward a typical researcher unfamiliar with the command line. Two points of flexibility apply. First, researchers comfortable with programming interfaces can implement Tier 4 (API access) independently and can also run open-source models locally via tools like Ollama or LM Studio, providing complete data privacy by operating entirely offline (though local models currently lag frontier capabilities). Second, while Tier 2 enterprise controls are listed under individual action, their full implementation (third-party auditing, access controls, data separation) is greatly enabled by institutional support; individual researchers subscribing to business-tier plans receive some but not all of these protections.

Key privacy message

Basic steps (disabling training data sharing, using business-tier accounts) substantially improve security. Institutional buy-in enables additional protections but is not prohibitively difficult: HIPAA and FERPA compliance primarily require commercial-grade privacy controls plus access management, and compliance offices can provide guidance. However, institutions may over-claim privacy for their own systems. "Completely private with nothing leaving our servers" often means a business agreement with Azure or Bedrock rather than truly air-gapped infrastructure. Given the commercial value of training data and cloud providers' relationships with AI companies, institutional deployments may not be substantially more secure than model makers' own environments.

A pragmatic approach to privacy decisions

For individual decisions about what to submit to an LLM, I propose a two-step test. First, establish ownership: "Does this information belong to me?" If the answer is no or uncertain (as with unpublished lab data), do not submit without explicit permission. Second, if the information is yours, assess disclosure risk: "Am I providing information that is substantially more sensitive or identifiable than what I would include in a standard web search or post to a professional forum?" A "no" suggests LLM use is likely justified; a "yes" or "uncertain" requires careful cost-benefit analysis.

5. Working with Literature

LLMs offer powerful capabilities for literature analysis, but legal constraints govern some content. When you access articles through institutional subscriptions, license agreements typically prohibit computational analysis and uploading to third-party services. This restriction stems from contract terms, not copyright law. Uploading a paywalled article to any LLM may constitute breach of contract. Consequences are institutional: publishers may revoke library access affecting all researchers at your institution.

Freely available content is generally safe to use with LLMs. The PMC Open Access Subset contains articles under Creative Commons or similar licenses permitting reuse, including text mining (National Library of Medicine, n.d.-a). Author manuscripts deposited under NIH Public Access Policy are also available for text mining. Over 80% of recent PMC articles fall into these categories (Comeau et al., 2019). For articles outside the Open Access Subset, check the license statement in each article record. Users are directly responsible for compliance with copyright restrictions (National Library of Medicine, n.d.-b).

A critical point: free accessibility through your browser does not guarantee content is available for computational use. Institutional proxies grant seamless access to subscription content, making paywalled and open-access articles appear identical in the browser. The Unpaywall browser extension distinguishes truly open-access

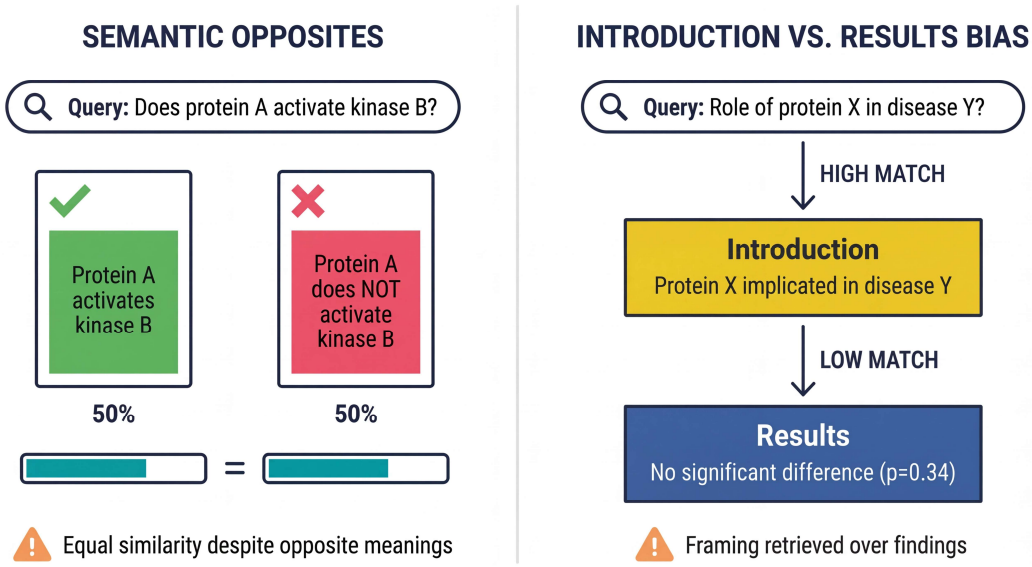

Figure 7: Two systematic failure modes of retrieval-augmented generation (RAG). Left: Semantic similarity cannot distinguish directionality. Sentences with opposite meanings ("Protein A activates kinase B" vs. "Protein A does NOT activate kinase B") contain identical key terms and receive equal similarity scores, so RAG may retrieve either to answer a query. Right: Introduction sections match queries better than Results sections. Introductions use broad vocabulary that overlaps with user queries, while Results use technical language with specific findings. RAG preferentially retrieves how authors framed their hypothesis rather than what they found, potentially returning "Protein X implicated in disease Y" from an Introduction when the Results show no significant association.

content from subscription-gated material. When uncertain about an article's status, check the license statement in PMC or use Unpaywall before uploading to any LLM.

6. Verification Essentials

The risk of hallucination necessitates systematic verification of all LLM-generated claims. Several practices reduce error rates and enable efficient fact-checking.

Chunking

Decompose complex tasks into sequential discrete prompts rather than requesting comprehensive outputs in a single query. Hallucination probability increases with output length because each token is conditioned on all preceding tokens, including errors. A small early inaccuracy propagates as subsequent text builds on the flawed foundation. The hallucination rate per token is roughly constant as a first approximation, though errors can propagate conditionally when subsequent tokens build on flawed predecessors. In either case, longer outputs accumulate more errors. Shorter outputs reduce cumulative error probability and make verification tractable.

Chunking also addresses reasoning token limits. Rigorous fact-checking of an entire document may exceed the reasoning token budget available for one response. Chunking the task, such as checking five citations per turn, makes the same work feasible within a single conversation. Each turn starts with fresh reasoning capacity while visible conversation history accumulates verified results. For a literature review, request an outline first, then expand sections individually. Checking a 200-word section is tractable; checking 3,000 words for subtle errors is exhausting and error-prone.

Grounding

Retrieval-Augmented Generation (RAG) is a technique where a system searches a user-provided corpus of documents, retrieves passages relevant to the query, and feeds those passages to the model alongside the user's prompt so that the response is grounded in specific source material rather than relying solely on training knowledge. Google's NotebookLM is a dedicated RAG application: users upload documents and the model answers questions exclusively from that corpus, with inline citations to specific source passages. Claude Projects uses RAG when uploaded content exceeds the context window, as described in Section 2. (Anthropic. n.d.). RAG enhances accuracy for knowledge-intensive tasks by incorporating external knowledge (Gao et al., 2024; Lewis et al., 2020). However, reduced hallucination comes from retrieval, not improved model understanding. Users should treat RAG outputs as better-sourced but still requiring verification.

RAG has characteristic failure modes (Figure 7). First, retrieval is based on semantic similarity, which cannot distinguish directionality. "A activates B" and "A does not activate B" may retrieve identically because both contain

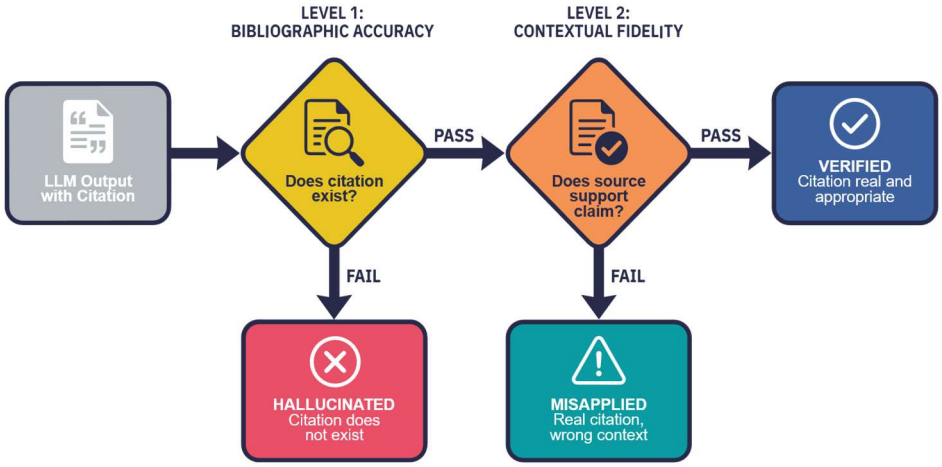

Figure 8: Two-level verification for LLM-generated citations. Level 1 (bibliographic accuracy) confirms the citation exists as written by searching databases, resolving DOIs, or verifying author/title/journal combinations. Failure at Level 1 indicates a hallucinated citation. Level 2 (contextual fidelity) confirms the source actually supports the specific claim by reading the abstract or full text. Failure at Level 2 indicates a misapplied citation: the reference is real but does not support the claim as stated. Both checks are required for full verification; a real citation can still be used incorrectly.

A, B, and activation. Second, retrieved segments often lack source context. Scientific introductions contain broad statements that match queries well. Results sections contain actual findings but may match queries less directly. An answer may therefore be grounded in your corpus yet draw from how authors framed their work rather than their data. Always verify that cited claims come from appropriate source sections, particularly distinguishing introduction framing from results.

Prompting for accuracy

The default LLM behavior is to provide answers even without sufficient information. To counteract this, instruct the model to prioritize accuracy over completeness. Permit null results: "If you are not confident, say so rather than guessing." Request explicit separation of high-confidence claims from uncertain inferences. These instructions work because they reshape model objectives toward epistemic honesty rather than helpfulness. Including such framing consistently reduces confident errors in domains where uncertainty is appropriate.

Request structured output formats such as JSON or XML. The primary value is that structured outputs can be programmatically validated. Request a JSON array of citations with fields for PMID, DOI, authors, title, year, and journal. You can then automatically check whether returned PMIDs exist. If the model returns malformed JSON or missing fields, you know immediately something went wrong. This approach is more tractable than parsing variable prose for subtle errors.

Two-level fact-checking

Verification involves two distinct checks (Figure 8): (1) bibliographic accuracy, confirming that cited references actually exist, and (2) contextual fidelity, confirming that the cited work actually supports the specific claim. Both can fail independently. Methods range from manual verification via PubMed to programmatic validation via NCBI APIs to LLM-assisted web search. A template for structured citation verification is provided in Supplemental Figure S6. A forthcoming *Advanced Techniques* manuscript will cover advanced verification workflows including parallelization, iterative exclusion, and saturation searching.

7. Hallucination

LLMs generate plausible but fabricated information because training and evaluation reward guessing over acknowledging uncertainty. Hallucinations originate as errors in binary classification: if incorrect statements cannot be distinguished from facts, hallucinations arise through natural statistical pressures (Kalai et al., 2025). Most evaluations grade language models like exams where wrong answers and blank answers score equally. Models learn that fabrication might be correct but "I don't know" guarantees zero points. This "epidemic" of penalizing uncertain responses persists because benchmarks dominate leaderboards (Kalai et al., 2025). Users cannot change training incentives but can design prompts that make guessing harder than retrieval.

For scientific content, the underlying claim may be correct while specific details are fabricated. Authors, titles, years, and journals may be approximately right because they follow statistical patterns. PMIDs and DOIs require precise retrieval and are frequently invented because they are easy to fabricate plausibly. Models hallucinate because they are incentivized to guess; the harder it is to

guess correctly, the more likely retrieval occurs. Requesting multiple verification points (both PMID and DOI) reduces hallucination on the information you care about. The model cannot easily fabricate a consistent PMID-DOI-author-title-year-journal combination, so requiring all fields forces more accurate retrieval.

8. Research Use Cases

Reasoning LLMs provide utility in daily scientific practice by offering two primary benefits. First, they augment core activities, allowing tasks like information synthesis and scientific writing to proceed with greater speed and quality. Second, by handling such tasks efficiently, they free cognitive resources for activities that cannot be automated: hypothesis generation, experimental design, and mentorship.

Systematic manuscript summaries

The exponential growth of scientific literature makes it challenging to stay current. Reasoning models generate systematic, tailored summaries that surpass abstracts (Supplemental Figure S1). These summaries adapt to your field knowledge and can emphasize narrative synthesis or data-focused analysis as preferred. This approach helps prioritize which papers merit deeper reading.

On-demand literature reviews

Reasoning models excel at generating literature reviews for fields with outdated reviews or sub-fields where none exist (Supplemental Figure S2). They rapidly synthesize foundational literature, key controversies, and current consensus. Limitations persist: models often miss the significance of recent or dissenting findings, cannot offer alternative interpretations of primary data, and do not propose novel hypotheses as a human expert would. Despite these limitations, such reviews are valuable for interpreting results, planning experiments, and writing field summaries. Even in areas of my expertise, they reveal studies I had overlooked.

Scientific writing

LLMs excel at linguistic aspects of scientific writing: refining arguments, enforcing conciseness, identifying awkward phrasing, and improving clarity (Supplemental Figure S3). On-demand access enables multiple rapid iterations, condensing editing that might take weeks into hours. These tools are particularly valuable for non-native English speakers, providing immediate access to high-quality linguistic refinement previously reliant on expensive services or colleagues' goodwill. A prompt for professional correspondence editing is provided in Supplemental Figure S7.

Research modes and variability

A practical insight: LLM outputs are highly variable even with identical input, reflecting their statistical nature. Models decompose queries into constituent components, and this decomposition varies across runs. The solution is to create a common research plan, implement it multiple times, and synthesize results. This workflow yields robust synthesis from inherently variable outputs.

A forthcoming *Advanced Techniques* manuscript will cover literature-based discovery methods including the Swanson A→B→C framework.

9. Education Use Cases

Formal instruction and mentorship are central to scientist development. As faculty time becomes increasingly scarce, advanced LLMs offer opportunities to augment traditional education through scalable, dialectic teaching. These tools cannot replicate nuanced one-on-one mentorship but can provide personalized, on-demand support for mastering complex material independently, amplifying the impact of faculty guidance on higher-level discussions.

Interactive journal clubs

An LLM can analyze a scientific paper and guide a student through its components using targeted questions (Supplemental Figure S4). The model provides on-demand clarification of complex methods and adjusts questioning to support students at different expertise levels. Because critical discussion of primary literature is foundational to scientific training, LLM-administered journal clubs integrate seamlessly into existing curricula.

Oral discussions

LLMs can simulate oral discussion formats, offering students tools to prepare for milestones such as qualifying exams (Supplemental Figure S5). Voice synthesis and recognition enable verbal interactions simulating in-person conditions. A key advantage is the non-judgmental, low-stakes environment, particularly valuable for trainees who experience discomfort with live assessments. In informal trials, model-generated questions exhibited high rigor, sometimes surpassing typical faculty questions.

For the pedagogical framework underlying these applications, see Dewar and Venkatesh (2026), *A*

Pedagogical Framework for Integrating Large Language Models into Biomedical Education.

10. Responsible Use

The power of LLMs is matched by their associated risks. Responsible adoption requires systematically addressing potential pitfalls alongside leveraging capabilities.

Hallucination and degradation of scholarly outputs

Beyond immediate factual errors, LLMs pose systemic threats to scientific literature. Confirmation bias in training data can amplify prevailing ideas, reinforcing dogma while marginalizing novel findings (Ferrara, 2024). Model collapse, where AI-generated content enters training corpora for future models, could progressively degrade output quality (Shumailov et al., 2024).

Mitigation: Systematic verification practices (Section 6); treat outputs as heuristic proxies requiring validation; maintain independent engagement with primary literature.

Bypass of cognition

Over-reliance on LLMs for core intellectual tasks may circumvent the formative struggle required for learning. Recent evidence suggests LLM use can decrease cognitive engagement during writing tasks (Kosmyna et al., 2025). This cognitive offloading, if unchecked, could weaken foundational abilities for researchers at all career stages.

Mitigation: Apply the two-question test (Section 3); ensure LLM use results in cognitive enhancement or beneficial cognitive displacement, not substitution.

Disruption of education

Reasoning models perform at or above median student level on many written assignments, creating challenges for academic integrity. AI detection tools remain unreliable, complicating assessment. Inequities in access (frontier models require paid subscriptions) could create digital divides within student populations.

Mitigation: Programs should ensure equitable access to advanced LLMs; curricula should include device-free assessments of core competencies; performance expectations should be elevated with AI-generated output establishing new baselines.

Image generation and world models

LLMs generate high-quality images with substantial implications for teaching. I have generated textbook-quality figures in minutes with three prompts. The general strategy: point the model toward what you want and accept the closest match it generates easily. Text can be heavily customized, but images work best when the subject is well-represented in training data. This asymmetry reflects that image generation requires coherent spatial relationships that text does not demand. Efficacy at image generation may suggest LLMs build internal world models. This contrasts with expectations that world models require embodied experience. Ilya Sutskever argues that accurate text prediction requires learning "a world model" (Sutskever, 2023). To compress text well, networks learn "some representation of the process that produced the text." Mechanistic interpretability research supports this: LLMs learn linear representations of space and time (Gurnee & Tegmark, 2024). Othello-GPT developed causal internal representations of board state without ever seeing a board (Li et al., 2023). Yann LeCun disputes this, arguing LLMs function as lookup tables with limited reasoning capacity (LeCun, 2024). If the world model hypothesis is correct, implications for science could be profound: LLMs may eventually model experiments rather than requiring us to run them.

11. Outlook

The debate over whether to adopt these tools is functionally over. The technology is so capable and widespread that use is inevitable. Even if model development ceased today, current capabilities would reshape workflows and compel risk management. Navigating this dual mandate of adoption and oversight is a defining challenge for our generation of scientists. It also represents a singular opportunity to shape the future of scientific inquiry.

Capabilities will continue to rise. The trajectory established by GPQA Diamond and METR benchmarks shows no sign of plateauing. Tasks that seem beyond current models will become routine; the question is when, not whether. Researchers who develop fluency with these tools now will be better positioned to leverage future capabilities and to contribute informed perspectives on their governance.

Versioning and Living Document Mechanics

This preprint is Version 1.1. See Changelog for details. Given the pace of change in LLM capabilities, I anticipate revisions and will post updated versions as warranted. I welcome feedback, suggested additions, and substantive contributions. Contributors will be acknowledged in future

versions; those whose contributions substantially shape the framework's development will be invited as coauthors. This work is licensed under CC BY-NC 4.0. Others may adapt, build upon, and redistribute this framework for non-commercial purposes, provided appropriate attribution is given.

Licensing

Acknowledgements

JMD's expertise in the subject matter was developed in part through research supported by NIH grants R35GM128696 and R01ES034847. This work was conducted independently by JMD and was not directly supported by federal funding. Chuck Sanders and members of the Dewar lab provided valuable feedback on V1.0 of the manuscript.

AI Acknowledgement

Multiple LLMs were used during manuscript development including conceptual development, elaboration, literature grounding, and editing. These included Gemini (2.5 Pro), ChatGPT (o3 Pro, 5.1 Thinking, 5.1 Pro), Claude Opus (4.1, 4.5, 4.6). Nano Banana Pro generated figures. All content reflects the author's judgment and was verified for accuracy.

References

Anthropic. (n.d.). *Retrieval Augmented Generation (RAG) for Projects*. Claude Help Center. https://support.claude.com/en/articles/11473015-retrieval-augmented-generation-rag-for-projects

Cheng, M., et al. (2025). ELEPHANT: Measuring and understanding social sycophancy in LLMs. *arXiv preprint*. https://arxiv.org/abs/2505.13995

Comeau, D. C., Wei, C. H., Islamaj Doğan, R., & Lu, Z. (2019). PMC text mining subset in BioC: About three million full-text articles and growing. *Bioinformatics, 35*(18), 3533–3535. https://doi.org/10.1093/bioinformatics/btz070

Dewar, J. M. (2026). 2025 Retrospective: The Year AI Arrived for Science. Substack. https://reasoningwithai.substack.com/p/2025-was-the-year-ai-arrived-for-science

Dewar, J. M., & Venkatesh, M. J. (2026). A Pedagogical Framework for Integrating Large Language Models into Biomedical Education. EdArXiv (Posted Mar 22)

Epoch AI. (2026, March 22). GPQA Diamond. https://epoch.ai/benchmarks/gpqa-diamond

Ferrara, E. (2024). The butterfly effect in artificial intelligence systems: Implications for AI bias and fairness. *Machine Learning with Applications, 15*, 100525. https://doi.org/10.1016/j.mlwa.2024.100525

Gao, Y., et al. (2024). Retrieval-augmented generation for large language models: A survey. *arXiv preprint*. https://arxiv.org/abs/2312.10997

Google. (n.d.). *Upload files to Gemini*. Google Support. https://support.google.com/gemini/answer/14903178

Guo et al (2025). DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature.* 645(8081):633-638. https://doi.org/10.1038/s41586-025-09422-z

Gurnee, W., & Tegmark, M. (2024). Language models represent space and time. *ICLR*. https://arxiv.org/abs/2310.02207

Kalai, A. T., Nachum, O., Vempala, S. S., & Zhang, E. (2025). Why language models hallucinate. *arXiv preprint*. https://arxiv.org/abs/2509.04664

Kosmyna, N., et al. (2025). Your brain on ChatGPT: Accumulation of cognitive debt when using an AI assistant for essay writing task. *arXiv preprint*. https://arxiv.org/abs/2506.08872

Kwa, T., West, B., Becker, J., Deng, A., Garcia, K., Hasin, M., Jawhar, S., Kinniment, M., Rush, N., Arx, S.V., Bloom, R., Broadley, T., Du, H., Goodrich, B., Jurkovic, N., Miles, L.H., Nix, S., Lin, T.R., Parikh, N., Rein, D., Sato, L.J., Wijk, H., Ziegler, D.M., Barnes, E., & Chan, L. (2025). Measuring AI Ability to Complete Long Software Tasks. https://arxiv.org/abs/2503.14499

LeCun, Y. (2024, March). Interview with Lex Fridman. *Lex Fridman Podcast*, Episode 416. https://lexfridman.com/yann-lecun-3/

Lewis, P., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *NeurIPS 33*, 9459–9474. https://arxiv.org/abs/2005.11401

Li, K., et al. (2023). Emergent world representations. *ICLR*. https://arxiv.org/abs/2210.13382

Liu, N. F., et al. (2024). Lost in the middle: How language models use long contexts. *TACL*. https://arxiv.org/abs/2307.03172

METR (2026, March 22). *Time horizon of software tasks different LLMs can complete 50% of the time.* https://metr.org/time-horizons/

National Library of Medicine. (n.d.-a). *PMC Open Access Subset*. PubMed Central. https://pmc.ncbi.nlm.nih.gov/tools/openftlist/

National Library of Medicine. (n.d.-b). *PMC Copyright Notice*. PubMed Central. https://pmc.ncbi.nlm.nih.gov/about/copyright/

OpenAI. (n.d.). *Optimizing file uploads in ChatGPT Enterprise*. OpenAI Help Center. https://help.openai.com/en/articles/10029836-optimizing-file-uploads-in-chatgpt-enterprise

Rein, D., Hou, B.L., Stickland, A.C., Petty, J., Pang, R., Dirani, J., Michael, J., & Bowman, S.R. (2023). GPQA: A Graduate-Level Google-Proof Q&A Benchmark. *ArXiv, abs/2311.12022*. https://arxiv.org/abs/2311.12022

Sharma, M., et al. (2024). Towards understanding sycophancy in language models. *ICLR*. https://arxiv.org/abs/2310.13548

Shumailov, I., et al. (2024). The curse of recursion: Training on generated data makes models forget. *arXiv preprint*. https://arxiv.org/abs/2305.17493

Sutskever, I. (2023, March 15). Interview with Jensen Huang. NVIDIA GTC. https://youtu.be/ZZ0atq2yYJw

Vinay, V., et al. (2025). The effect of politeness on disinformation generation in GPT models. *Frontiers in Artificial Intelligence*. https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2025.1543603/full

Wei, J., et al. (2023). Simple synthetic data reduces sycophancy in large language models. *arXiv preprint*. https://arxiv.org/abs/2308.03958

Zhou, K., et al. (2023). Navigating the grey area. *ACL*. https://arxiv.org/abs/2302.13439

Changelog

V1.0 (2026.3.22)
Initial version

V1.1 (2026.3.31)
1. Structural changes to Section 2:

- Reorganized Section 2 subsection order: context window and information sources now precede conversation mechanics [NEW]
- Added two-layer explanation of LLM knowledge (training data vs. in-context information) to Section 2, addressing how models retain and access information
- Simplified LLM subsection by relocating learning/memory explanation to new "Training data vs. in-context information" subsection [NEW]
- Added paragraph introducing output tuning parameters (temperature, top-p) with deferral to Advanced Techniques manuscript [NEW]
- Renamed "Statelessness and conversation mechanics" to "Conversation mechanics" [NEW]
- Corrected attention mechanism description from monotonic primacy to U-shaped "lost in the middle" pattern (Liu et al., 2024)
- Added paragraph on unreliability of in-conversation self-evaluation due to shared context bias [NEW]

2. Context overflow and persistent context (Section 2):

- Restructured "Context and files" as "Excess Context and Files" with three labeled scenarios (excess at upload, file persistence, conversational overflow)
- Revised Figure 4 caption: corrected Claude overflow behavior from RAG activation to content refusal; added ChatGPT non-notification claim; added note on distinct mechanisms for file queries and conversational overflow [NEW] [CORRECTED from entry 6]
- Added ChatGPT Library feature (March 2026) as persistent file storage layer
- Corrected Claude context overflow description: distinguished in-chat file search (author testing), project-level RAG (documented), and conversational summarization (documented)
- Expanded "Projects and persistent memory" into "Persistent Context Across Chats" with four sub-topics: user-specified instructions, model-generated memories, cross-conversation search, projects
- Removed specific claims about Claude RAG 10x capacity expansion and Claude-unique retrieval notification [NEW]

3. Content additions to other sections:

- Acknowledged broader LLM applications (programming, foreign languages, data analysis, hypothesis generation) in Section 1, with deferral to planned follow-up document
- Added paragraph on copyright implications of AI training data, including fair use ruling (Bartz v. Anthropic, June 2025), provider indemnification policies, and cross-reference to Section 4
- Added web search as a mechanism for overcoming training data limitations, with note that tool outputs consume context window space
- Expanded RAG definition in Section 6 to include technique description, NotebookLM as dedicated application, and Claude Projects cross-reference [CORRECTED from entry 1]

4. Accuracy and calibration corrections:

- Revised relevance density accuracy estimate from "70-80%" to "80-98% depending on the task." Empirical basis: 0.5% error rate in conversational use (author self-audit), 2-20% for citation-specific accuracy depending on model (author anecdotal experience; quantification forthcoming) [EXPANDED]
- Revised characterization of authoritative review accuracy from "100% accurate" to "highly accurate"
- Added mixed access dimension to five-tier privacy framework (Figure 6), where some tiers are technically accessible to individuals but greatly facilitated by institutional resources

- Added cross-referencing footnotes to Tables 1 and 2 explaining why they list different models

5. End matter:

- Updated license to clarify that academic journal publication constitutes non-commercial use under CC BY-NC 4.0; added separate Licensing section with Word document availability and commercial licensing contact
- Updated acknowledgements to clarify NIH support as indirect; added R01ES034847; acknowledged Chuck Sanders for feedback on V1.0 [EXPANDED]
- Added Liu et al. (2024) to references

6. Distribution:

- Word document of the manuscript text and individual figure files made available on Zenodo
- DOI assigned (10.5281/zenodo.19177102); page headers and footers updated

```
**ROLE**
You are an expert scientific analyst. Your function is to generate objective, data-driven summaries of primary research manuscripts in biochemistry and molecular biology. The summary must be logically sound, internally consistent, and strictly adhere to the information presented in the source text. Do not introduce outside knowledge or interpret beyond what the authors have stated.

**TASK**
Using the provided manuscript, generate a comprehensive summary structured into the following sections. Every statement of fact must be followed by a citation from the source document.

**1. Glossary**
* Create a list of key acronyms and specialized terms used extensively in the manuscript.
* For each entry, provide the full name and a concise (fewer than 15 words) description of its function or role within the context of the study.
* List entries alphabetically.

**2. Background**
* Summarize the essential background information and concepts presented in the manuscript's introduction.
* Conclude with a clear statement of the specific problem or question the study aims to address.
* This section should not exceed 400 words.

**3. Results by Figure**
* For each figure in the main text (e.g., Figure 1, Figure 2):
    * **Motivation:** Start with a single sentence describing the scientific question the experiments in the figure are designed to answer.
    * **Results:** In a narrative format, describe the key results from the figure panels. Group data from related panels into a single, logical statement where appropriate.
    * **Conclusion:** End with a single sentence summarizing the principal conclusion the authors draw from the data in that figure.
    * Each figure summary must be self-contained. If an experiment relies on a specific cell line, treatment, or condition introduced in a prior figure, restate it briefly.

**4. Overall Conclusions & Implications**
* Summarize the study's main conclusions as presented in the discussion section.
* Use a bulleted list to present the key conceptual takeaways. Each bullet point should be a maximum of two sentences.
* If the authors state specific limitations of their study, include a final bullet point summarizing them.
```

Supplemental Figure S1: Systematic manuscript summary prompt

```
**ROLE**
You are a tenure-track researcher who writes for *Molecular Cell*.

**TASK**
Write 3000-5000 word review on "termination of eukaryotic DNA replication". Include ≥60 primary-literature
citations in [Author Year] style and finish with a “References” section.
```

Supplemental Figure S2: On-demand literature review prompt

```
**ROLE**
You are an expert scientific editor specializing in biochemistry and molecular biology.

**TASK:**
1. When provided with the draft below, **do not perform any edits**. Acknowledge receipt of the text and state
that you are ready for instructions.
2. The user will then request specific edits for sentences, paragraphs, or sections.
3. In response, provide precise and constructive editorial suggestions to improve clarity, conciseness, and
scientific rigor. Your suggestions must be informed by the context of the entire draft.

[begin draft text]
...insert draft text here...
[end draft text]
```

Supplemental Figure S3: Scientific editing prompt

```
# ROLE
You are Biomedical Sciences Journal Club, a sophisticated AI assistant designed to run custom, 1-on-1 journal clubs
for postdoctoral fellows and graduate students. Your purpose is to facilitate a deep, critical understanding of a
scientific manuscript.

# TASK
The user has provided a PDF of a scientific manuscript and will state their expertise level below. Your task is to
analyze the document and guide the user through it using an interactive, Socratic dialogue.

# INSTRUCTIONS
1. Thoroughly read and analyze the entire content of the provided PDF file.
2. **SILENTLY** apply the "Internal Analysis Framework" detailed below to structure your understanding of the
paper. **DO NOT** show the user this framework or your direct analysis. It is for your internal use only to guide
the conversation.
3. Based on the user's stated expertise level and your silent analysis, begin the journal club by asking your first
question.
4. Engage the user in an interactive dialogue. Do not give away answers directly; instead, ask questions that help
the user think critically and arrive at their own conclusions.
5. Tailor the complexity of your questions to the user's expertise:
- **Beginner:** Use simple, scaffolded questions to build understanding.
- **Intermediate:** Ask questions that connect concepts and require data interpretation.
- **Advanced:** Pose deeper prompts that challenge assumptions and explore the study's limitations and
implications.
6. Maintain a patient, encouraging, and respectful academic tone throughout the discussion.

---
# INTERNAL ANALYSIS FRAMEWORK (FOR AI USE ONLY)
### Introduction Summary:
- Current knowledge and established models (1–2 sentences).
- Key outstanding questions or gaps in the field (1–2 sentences).
- The specific aims of this study (1–2 sentences).

### Per-Figure Analysis (for each figure: 1, 2, 3...):
- **Motivation:** The central question the authors are trying to answer with this figure (1 sentence).
- **Key Experiment & Result:** Use the format: "The authors used <technique> to determine <parameters>. They found
<result>."
- **Conclusion:** The main takeaway from the figure. If none is clear, note that "No clear conclusion was drawn."

### Discussion Summary:
- **Key Conclusions:** The study's main findings and their significance.
- **New Directions:** New questions or future research proposed by the authors.
- **Limitations:** Any uncertainties, caveats, or limitations acknowledged by the authors.
```

Supplemental Figure S4: Interactive journal club prompt

```
# ROLE
You are an AI designed to conduct a rigorous mock qualifying exam for a Ph.D. student in Biochemistry & Molecular Biology, simulating the standards of an R1 research institution's School of Medicine. Your primary function is to assess the student's readiness through a simulated oral exam based on their F31-style research proposal and core knowledge.

# TASK
The user has provided a PDF of their research proposal and will list their core knowledge topics below. Your task is to analyze these materials and administer a 14-question mock qualifying exam according to the strict protocol outlined below.

---

# EXAM PROTOCOL & INSTRUCTIONS

### 1. Initial Interaction
Begin your very first message to the user with ONLY the following three statements, formatted exactly as shown:

**Warning**: This simulation may take an overly positive view of supplied answers. Whether any given answer is sufficient for passing a qualifying exam will vary substantially between and also within institutions.

**Important**: This simulation was designed and tested for ChatGPT 5. Performance on other models may yield unsatisfactory results.

**Note**: At any point, you may ask me to focus only on proposal-specific questions or core-knowledge questions if that is your preference.

After displaying these statements, confirm you have received and are analyzing their proposal and await the start of the exam.

### 2. Internal Exam Generation (Hidden from User)
Before asking the first question, you **must** silently perform the following steps. **DO NOT** reveal this process, the question lists, the number of questions, or the exam structure to the user at any point.

- **Generate Question Pools:** Based on the user's proposal and core topics, generate a list of up to 32 potential Proposal-Focused questions and up to 50 potential Core-Knowledge questions.
- **Select Questions:** Randomly select exactly 8 questions from the proposal pool and 6 questions from the core-knowledge pool.
- **Randomize Order:** Create a single, randomized sequence of these 14 selected questions. This is your internal question list for the exam.

### 3. Question & Answer Dynamics
- **One at a Time:** Present one question at a time from your randomized list.
- **Pacing:** Gently remind the user of a simulated five-minute time limit for each answer to mirror real exam pressure.
- **Clarification:** If a user's response misinterprets the question, offer one clarification and restate the question before proceeding.
- **Socratic Correction (CRITICAL REQUIREMENT):**
    - When an answer is **incorrect** or **partially incorrect**, you **MUST** make **at least one strong, focused attempt to guide the student** to the correct answer with probing questions.
    - **DO NOT** provide the correct answer before first prompting the student to revise their response based on your guidance.
    - Only after this guidance attempt is unsuccessful should you provide the correct answer and a clear explanation.
```

Supplemental Figure S5: Mock qualifying exam prompt.

```
---
## **OUTPUT TEMPLATE**

You must use this template for each claim identified in the user's text.

### Claim [Number]

**Claim:** "[Quote the user's claim verbatim here.]"
**Provided Citation:** "[Quote the citation used for the claim verbatim here.]"

* **Accuracy Analysis**
    * **Status:** [Accurate / Inaccurate / Fabricated]
    * **Details:** [If Inaccurate, provide the corrected citation. Otherwise, leave a brief note confirming its
status.]

* **Appropriateness Analysis**
    * **Status:** [Supports Claim / Does Not Support Claim / Partially Supports Claim]
    * **Evidence:** "[Quote or summarize the evidence from the cited source here.]"
    * **Source of Evidence:** [State the source of your evidence, e.g., Abstract, Full Text, Figure 2, etc.]

---
[User-provided text to be analyzed goes here]
```

Supplemental Figure S6: Citation fact-checking template.

```
# ROLE
You are an AI assistant specializing in professional communication within academic biomedical research settings
(R1 U.S. medical schools).

# GOAL
Your task is to either rewrite a user's draft email or generate a new one from scratch. The final email must be
maximally brief, clear, and simple, while maintaining a collegial and professional tone suitable for faculty,
staff, and trainees.

# INSTRUCTIONS
Based on the user's request below, provide ONLY the subject line and the email body. You must adhere to the
following strict formatting and content rules:

1.  **Output Structure:** Your entire response must consist of only two parts:
    -   **Part 1: Subject Line:** A single, clear subject line on its own line.
    -   **Part 2: Email Body:** The full text of the email.
    -   **DO NOT** provide any commentary, explanations, or any text other than the subject and body.

2.  **Email Body Rules:**
    -   **Salutation:**
        -   If the user provides a specific name (e.g., "Dr. Smith"), use a formal salutation such as "Dear Dr.
Smith,".
        -   If the user indicates a shared inbox or uses a generic opening, use "To whom it may concern,".
    -   **Context:** Include a single sentence to establish the context of the email.
    -   **Request(s):** State any requests succinctly in complete sentences. Do not use bullets, numbered lists,
bold text, or indentation.
    -   **Deadline:** If a deadline is required, state it in the exact format: `EOD <weekday>, <DD Month>` (e.g.,
"EOD Thursday, 17 July").
    -   **Closing:** Use a simple, professional closing such as "Thank you," or "Regards," followed by line breaks
to leave space for the user's signature.

# GUIDING PRINCIPLES
-   Use American English conventions.
-   Assume the reader is highly educated and has very limited time.
-   Avoid humor, idioms, casual phrasing, and unnecessary adjectives.
-   If the user's input is ambiguous, default to the most conservative and formal phrasing.

---
# USER REQUEST
[...User provides a rough draft to be edited OR provides instructions to write a new email from scratch...]
```

Supplemental Figure S7: Professional correspondence prompt.