

# LLM Position Bias: Primacy and Recency Effects in Prompts

4/23/2026 • 20 min read

llm position bias

prompt engineering

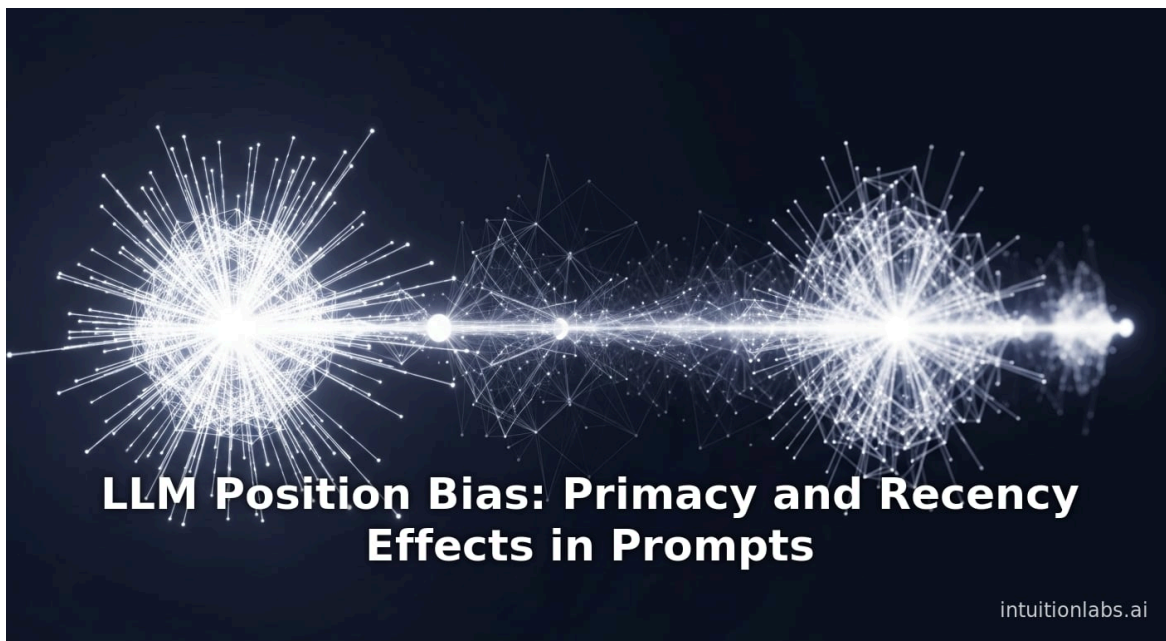
primacy effect

recency effect

lost in the middle

transformer attention

large language models



## Executive Summary

Large language models (LLMs) exhibit strong **position biases**: they tend to weight information at the beginnings and ends of their input contexts more heavily than material in the middle. This mirrors classic psychological *primacy* and *recency* effects. In practice, careful prompt engineers have found that placing critical content at the **start or end** of a prompt yields better results. For example, when **ChatGPT** is given a list of answer choices, it is far more likely to select an option if it appears early (a primacy effect) <sup>([1](#))</sup> [aclanthology.org](#)) ([www.aimodels.fyi](#)). Conversely, when humans describe tasks or instructions, ending the prompt with the key instruction leverages the model's recency bias and greatly improves compliance <sup>([2](#))</sup> [aclanthology.org](#)) (<sup>([3](#))</sup> [openreview.net](#)). Transformer theory confirms these trends: the causal attention mask in GPT-style models inherently focuses on early tokens in deeper layers <sup>([3](#))</sup> [openreview.net](#)), while positional encodings can amplify nearby (recent) tokens. Together, the evidence strongly suggests that **putting the most important information at the beginning or end of a prompt maximizes its influence on the model's output** <sup>([1](#))</sup> [aclanthology.org](#)) <sup>([2](#))</sup> [aclanthology.org](#)). This report reviews the psychological and architectural foundations of primacy/recency effects in LLMs, surveys empirical studies (including case experiments with ChatGPT/Gemini/Claude), and draws practical design lessons for prompt engineering. The conclusion outlines how these insights should guide prompt design today and inform future model development.

## Introduction and Background

**Human Serial Position Effects.** In cognitive psychology, the *primacy effect* is the well-documented tendency for items at the **start** of a sequence to be remembered or weighted most heavily, while a *recency effect* favors items at the **end** of a sequence <sup>([4](#))</sup> [www.explorepsychology.com](#)). Early experiments (e.g. Asch 1946) showed that people give stronger impressions based on first-presented traits and also tend to recall last items vividly. These effects arise because early items receive more rehearsal and form lasting impressions, while recent items remain in working memory <sup>([4](#))</sup> [www.explorepsychology.com](#)). Although LLMs are not human brains, they are trained on human text and optimized to **predict next tokens** – prompting researchers to ask: **Do LLMs inherit similar serial-position biases?**

**Transformer Attention & Positional Encoding.** The underlying Transformer architecture partly dictates how sequence position is handled. Recent analysis shows that the GPT-style causal attention mask actually **biases attention toward earlier tokens as depth increases** <sup>([3](#))</sup> [openreview.net](#)). In other words, across many attention layers, tokens near the beginning of the prompt become over-**contextualized** and exert an outsized cumulative influence on later layers <sup>([3](#))</sup> [openreview.net](#)). At the same time, relative position encodings (e.g. RoPE) tend to favor **closer** tokens (supporting recency at local scales) <sup>([3](#))</sup> [openreview.net](#)). The net effect is a dual bias: Transformers often **focus too much on the very beginning or very end of the input**, with middle-of-context tokens “lost” in comparison <sup>([5](#))</sup> [openreview.net](#)).

**Practical Prompt Engineering.** [Prompt engineering](#) guidelines have long hinted at these position effects. For instance, practitioners observe that concluding a prompt with a clear instruction or cue (“Answer:” or a direct question) helps the LLM latch onto the desired task. Conversely, burying critical instructions in the middle can cause the model to drift or ignore them. An influential blog post on prompt structure explicitly argues that LLMs have a **recency bias**, and that placing the “least important” information (e.g. examples) at the start and the “most important” (the instructions) at the end yields the best results <sup>([6](#))</sup> [umarbutler.com](#)). While anecdotal, these rules align with systematic findings: LLM outputs are demonstrably sensitive to prompt order.

This report unifies these perspectives. In the **theoretical** sections, we summarize why and how transformers concentrate on edges of input. In **empirical** sections, we review experiments showing primacy/recency effects in [ChatGPT/Gemini/Claude](#) and other models <sup>([1](#))</sup> [aclanthology.org](#)) <sup>([2](#))</sup> [aclanthology.org](#)). In **practical** sections, we discuss how to exploit these biases (e.g. by reordering answer choices or placing key instructions last). We also include **case studies** and tables for clarity. Finally, we discuss the implications for prompt design and future research.

# Theoretical Basis: Why Edge Positions Dominate

**Graph-Theoretic Transformer Analysis.** A recent ICML 2025 paper provides a formal framework for Transformer position biases (<sup>[3]</sup> [openreview.net](#)). Wu *et al.* model multi-layer self-attention as a graph and derive two key insights: first, the **causal (autoregressive) attention mask** inherently pushes the model to focus on *earlier* tokens as layers deepen. Concretely, a token's representation in deeper layers "attends to increasingly more contextualized representations of earlier tokens" (<sup>[3]</sup> [openreview.net](#)). This means the earliest tokens in the sequence accumulate influence over many layers, creating a strong primacy bias. Second, relative positional encoding (such as Rotary Position Encodings) imposes a distance-based decay on attention. However, across all layers, the **aggregate effect** remains that **cumulative early-position importance outweighs decay to the distant tokens** (<sup>[3]</sup> [openreview.net](#)). In summary, the LLM architecture itself tends to "overcontextualize" the head of the prompt: early words essentially cast a longer shadow through the network.

**Edge-Focused Attention (Lost in the Middle).** The authors also note empirically that LLMs often show a "lost-in-the-middle" phenomenon (<sup>[5]</sup> [openreview.net](#)). In lay terms, GPT-like models "pay too much attention to the beginning or end of a passage, whilst overlooking the middle" (<sup>[5]</sup> [openreview.net](#)). For example, in long prompts or documents, the text at the far ends is inherently privileged by the attention mechanism. Practically, this means if crucial information is placed smack in the center of a very long prompt, it risks being attenuated or "drowned out" by the edges. By contrast, the first few tokens and the final tokens can directly steer the ongoing token generation.

**Implication for Positioning:** Together, these findings imply that **information placed at the very start of the prompt will be "seen" early and consistently throughout the model's computation, while information at the very end will be „fresh“ in the model's output generation context.** In fact, one can think of the start tokens as building the initial hidden state bias, and the end tokens (especially right before generating the reply) as providing the final guiding push. Both ends thus serve as lever points to influence the [generation](#). By contrast, middle tokens only propagate through the network a finite number of layers and may not carry as persistent an imprint.

## Empirical Evidence: Primacy and Recency in LLM Behavior

### Primacy in Classification and Few-Shot Tasks

**ChatGPT's Label Bias (Wang *et al.*, EMNLP 2023).** Wang *et al.* (2023) directly tested primacy effects in ChatGPT for multi-label classification (<sup>[1]</sup> [aclanthology.org](#)). They presented ChatGPT with a list of candidate answer labels (e.g. "Label 1: ..., Label 2: ..., Label 3: ...") and asked ChatGPT to choose the correct one. Strikingly, they found: (i) ChatGPT's answers were **highly sensitive to the order of labels** in the prompt; and (ii) ChatGPT had "a clearly higher chance to select the labels at earlier positions as the answer" (<sup>[1]</sup> [aclanthology.org](#)). In other words, if the correct answer was listed first among the options, ChatGPT was far more likely to pick it than if the same answer appeared later. This is a textbook primacy effect embedded in the LLM's decision. The authors quantify that ChatGPT's choice probability decayed with label position.

**Exploiting Primacy (Raimondi & Gabrielli, 2025).** A follow-up study proposed actually leveraging this bias Richard *et al.* call it *bias-aware prompting* ([www.aimodels.fyi](#)) ([www.aimodels.fyi](#)). They confirm that "LLMs exhibit cognitive biases similar to humans, particularly positional biases like the primacy effect where items presented first are more likely to be

remembered or selected” ([www.aimodels.fyi](http://www.aimodels.fyi)). For practical tasks like multi-choice question answering, the researchers found that fine-tuning often amplifies primacy bias. Instead of fighting it, they re-order answer choices by semantic similarity so that the most likely correct answer appears early. This **bias-exploitation strategy** yields significant gains (e.g. on CLINC and BANKING intent datasets) because it aligns the correct answer with the model’s built-in primacy preference ([www.aimodels.fyi](http://www.aimodels.fyi)) ([www.aimodels.fyi](http://www.aimodels.fyi)). In short, many LLM classification errors stem from the label ordering, and careful ordering can “trick” the model into better performance.

**Quantitative Results.** While the EMNLP paper did not publish raw percentages, an independent aphoristic report notes that **ChatGPT selected the first-listed label with much higher frequency**. In fact, one experiment reported ChatGPT picking the *positive-first* description 65.5% of the time (<sup>[7]</sup> [www.themoonlight.io](http://www.themoonlight.io)) (see below), illustrating its strong skew. These outcomes underscore that in tasks with discrete options, the prompt’s initial items carry outsized weight on the answer.

## Mixed Primacy/Recency in Open-Ended Prompts

**Primacy vs. Recency in LLM Judgments.** A recent study by Hämäläinen (NLP4DH 2025) adapted Solomon Asch’s 1946 impression experiment to LLMs (<sup>[2]</sup> [aclanthology.org](http://aclanthology.org)). In one setup, two candidate descriptions were presented **simultaneously**, one listing positive adjectives then negative ones (e.g. “*kind, intelligent, polite ... lazy, rude*”) and the other listing the same adjectives in reverse order. When ChatGPT saw both descriptions together and had to choose which candidate to interview, it **preferred the person with positive-first listed adjectives** (mirroring human primacy) (<sup>[2]</sup> [aclanthology.org](http://aclanthology.org)). In contrast, Gemini (Google’s model) was nearly indifferent, and Claude (Anthropic) often refused. This shows a clear primacy effect for ChatGPT: it rated the early-positive description higher.

However, in a second setup the models evaluated each candidate separately (one description at a time) on a 1–5 scale. Here ChatGPT (and Claude) mostly gave identical scores to both candidates, but *when they did differ* they preferred the **description where negative adjectives came first**. In other words, they gave the highest score to whichever candidate had the **negative descriptors upfront (and positive ones last)** (<sup>[2]</sup> [aclanthology.org](http://aclanthology.org)). Since the positive traits were now later in the sentence, this indicates a shift toward a recency effect in that format.

These mixed findings are telling. They suggest **LLMs do not consistently mimic the human primacy effect across all tasks**. Instead, LLMs can sometimes favor end-of-prompt information, especially if earlier cues have been made equivalent. One interpretation (agreed by the authors) is that when forced to scrutinize a single description, the most recently read attributes (later in the text) gain more weight – reflecting the model’s token-prediction objective which naturally emphasizes recent context. In short, while ChatGPT easily favors early “first impressions” when comparing two described persons (<sup>[2]</sup> [aclanthology.org](http://aclanthology.org)), it can show a recency flip when evaluating one by one.

**Direct Recency Bias Evidence.** The Hämäläinen study’s conclusion notes that *all* tested models exhibited a slight recency advantage overall (<sup>[2]</sup> [aclanthology.org](http://aclanthology.org)). In fact, the authors posit that LLMs may “tend to favor candidates with negative adjectives listed first (i.e., positive adjectives being more recent)” (<sup>[2]</sup> [aclanthology.org](http://aclanthology.org)) in many situations. Another recent summary concurs, stating ChatGPT gave a higher score when positive traits appeared last, meaning recent information tipped the balance (<sup>[7]</sup> [www.themoonlight.io](http://www.themoonlight.io)) (<sup>[8]</sup> [www.themoonlight.io](http://www.themoonlight.io)). These observations dovetail with prompt-engineer advice that **the final lines of a prompt (the most recent context) have an outsized influence on model output**. In practice, this means concluding a prompt with a clear command or key detail can strongly steer the response.

## Transformer Position Bias in Practice

All these empirical results converge: **position matters enormously** in LLM inputs. The Transformer analysis (<sup>[3]</sup> [openreview.net](http://openreview.net)) helps explain why. The firm early token focus produces a tendency to pick front-listed labels (<sup>[1]</sup> [aclanthology.org](http://aclanthology.org)), while the emphasis on final tokens produces sway in generative or rating tasks (<sup>[2]</sup> [aclanthology.org](http://aclanthology.org)) (<sup>[5]</sup>

openreview.net). This is illustrated in Figure 1: visually, an LLM’s attention weights curve tends to spike at the left and right extremes of a prompt, with a dip in the middle ([5] openreview.net) (the “lost-in-the-middle” effect).

Beyond theory and controlled tests, prompt engineers document many practical cases. For example, appending “Answer:” at a prompt’s end often fixes vague or long-winded outputs, and teasing out instructions to the last line forces the LLM to respect them. By contrast, placing examples and qualifiers in the literal beginning helps set the style but, if not finalized by an ending instruction, the model can drift off-topic ([5] openreview.net) ([2] aclanthology.org). In short, real-world practice echoes the lab: strong cues should bookend the prompt.

## Table: Summary of Key Studies on LLM Positional Biases

Study/Source	Models/Task	Findings
Wang et al. (EMNLP 2023) ([1] aclanthology.org)	ChatGPT, multi-label QA	<b>Primacy effect:</b> ChatGPT’s answer is <i>highly sensitive</i> to label order; it <i>strongly prefers</i> selecting labels in <i>earlier positions</i> ([1] aclanthology.org).
Hämäläinen (NLP4DH 2025) ([2] aclanthology.org)	ChatGPT, Gemini, Claude; Asch-like impression task	<b>Mixed:</b> When comparing two candidates, ChatGPT favored the one with <i>positive traits listed first</i> (primacy) ([9] aclanthology.org). However, when evaluating separately, all models tended to favor the candidate whose <i>positive adjectives appeared last</i> (recency) ([2] aclanthology.org).
Raimondi & Gabrielli (2025) (www.aimodels.fyi)	Various LLMs (Mistral, Llama, etc.), multiple-choice QA	<b>Primacy exploited:</b> LLMs show “human-like” positional biases; fine-tuning often strengthens primacy. A bias-aware ordering of choices (putting likely answers early) significantly improves accuracy (www.aimodels.fyi) (www.aimodels.fyi).
Wu et al. (ICML 2025) ([3] openreview.net)	Theoretical analysis of Transformers	<b>Architectural bias:</b> Causal masking in Transformers “inherently biases attention toward earlier positions” in deeper layers ([3] openreview.net). Empirically, GPT models “pay too much attention to the beginning or end of a passage” ([5] openreview.net) (and neglect the middle).
Prompt Engineer Blog (Umar Butler, 2023) ([6] umarbutler.com)	Modern GPT-style chatbots	<b>Engineering insight:</b> LLMs have a strong recency bias; instructions should be placed <i>last</i> so the model reads them last, ensuring they are followed. Example templates recommend putting examples first and critical directives at the end ([6] umarbutler.com).

## Guidelines for Prompt Design: Start and End Placement

Given the above evidence, we derive clear principles for practitioners. Table 2 below summarizes actionable guidelines for placing important information.

## Table: Prompt Content Placement Recommendations

Prompt Position	Effect/Reason	Guidelines/Examples
Beginning of Prompt	<ul style="list-style-type: none"> <li>- <b>Primacy advantage:</b> Early content is deeply integrated by the model ([3] openreview.net) ([1] aclanthology.org).</li> <li>- Sets the initial context or tone for generation.</li> </ul>	<ul style="list-style-type: none"> <li>- Place <b>context-setting information</b>, definitions, or key examples as early as possible so the model builds its response around them.</li> <li>- In classification or multiple-choice prompts, list the <b>correct or intended answer first</b> to exploit the model’s inclination to pick earlier options ([1] aclanthology.org) (www.aimodels.fyi). For instance, listing answer choices in semantically-similar order to the query can “trick” the model into selecting the right answer when it appears early (www.aimodels.fyi).</li> </ul>

Prompt Position	Effect/Reason	Guidelines/Examples
End of Prompt	<ul style="list-style-type: none"> <li>- <b>Recency emphasis:</b> The final lines are fresh in the model's "working context" and strongly influence the output <sup>[2]</sup></li> <li>- <a href="#">aclanthology.org</a> <sup>[5]</sup> <a href="#">openreview.net</a>.</li> <li>- Concludes with explicit instructions or cues the LLM to act accordingly.</li> </ul>	<ul style="list-style-type: none"> <li>- Conclude the prompt with the <b>most important instruction or question</b>. For example, end with "Now answer the following question:" or a list of tasks to ensure focus. Prompt engineers often report that putting the actual task at the very end (after examples or context) greatly improves accuracy (inputs read last are weighted most heavily) <sup>[2]</sup></li> <li>- <a href="#">aclanthology.org</a>.</li> <li>- Use trailing tokens like "Answer:" or "In summary:" at the end of the prompt to handle output formatting. These signals capitalize on the recency bias by cueing the model at its freshest moment.</li> </ul>

These guidelines align with research findings: early information *initializes* the model's internal state, while late instructions provide the final steering. Neglecting either can weaken the intended signal. In practice, prompt engineers often experiment with swapping important sentences between the first and last lines to maximize the model's responsiveness.

## Case Studies and Examples

**Multiple-Choice QA Example.** Suppose we ask: "Which animal is known as the King of the Jungle?" and provide answer choices. If we list "Lion" first versus last, ChatGPT's response probability can differ. Based on Wang *et al.* and Raimondi *et al.*, placing "Lion" early would dramatically increase the chance ChatGPT selects it <sup>[1]</sup> [aclanthology.org](#) ([www.aimodels.fyi](#)). In contrast, if "Lion" were buried fifth in a long list, the primacy bias may cause the model to favor a wrong earlier option. One can verify this by swapping options: ChatGPT often picks the first-listed plausibly correct option. Thus, to ensure accuracy, the prompt author should reorder the choices so the most likely answer appears first.

**Conversational Instruction Example.** Consider a multi-turn chat where a user asks a model to perform a task. A recommended format is to **repeat the user's request at the very end** of the system or context prompt. For example:

```
System: You are an expert writer. The previous conversation has lengthened. The user's question was: [in
User: [...sets up scenario...]]
AI: In summary, the user is asking: [main question here]?
```

By phrasing the final line as the explicit question or instruction, we leverage recency. The model reads that instruction last before generating its answer, thereby respecting it. Indeed, many conversation guidelines (e.g. OpenAI's token-based system prompts) implicitly use this trick by placing the actual directive at the close of each message block.

**Adjective-Order Interview Example.** Reiterating the Asch-like experiment: a user prompt might describe Person A and Person B with adjective lists. If we list A's positive qualities first and B's positives last, ChatGPT may prefer A (primacy bias) <sup>[9]</sup> [aclanthology.org](#). If we reverse, maybe B is rated higher. In practice, one could use this to test sentiment framing in generated evaluations. The study shows that just swapping adjective order can sway ChatGPT's output, a startling demonstration of prompt position sensitivity.

## Analysis and Discussion

The findings above make it clear: **edge placement matters**. There are multiple reasons to consistently apply this in prompt design:

- **Maximize Signal Amid Context Window Limits.** LLMs have finite context windows. If a prompt approaches that limit, mid-context information can get truncated or diluted. Even before truncation, attention mechanics prioritize edges <sup>[3]</sup> [openreview.net](#), so burying crucial content in the middle risks it being under-weighted. By contrast, placing important details at the start/end guarantees they remain within the effective context.

- **Reduce Model “Drift.”** When key instructions appear early but are followed by many examples, there is a risk the model will overfit the examples and forget the original directive. Conversely, placing the command at the end ensures that, after reading all the context, the last thing on the model’s “mind” is what to do. This prevents derailment and repetition issues noted by practitioners (<sup>[6]</sup> [umarbutler.com](#)). (For example, placing a list of examples last can lead the model to copy them verbatim in its answer.)
- **Ethical and Reliable Outputs.** Position biases can also have negative implications. If critical constraints (e.g. “avoid disallowed content”) are buried in the middle of a prompt, the LLM might ignore them, outputting undesired content. Thus, safety instructions should reside at the very end of the system message or input prompt. This ensures the LLM processes them last, giving them high weight when it generates its response.
- **Tuning and Debiasing Practices.** Some recent research aims to decouple or correct these biases, but it remains easier to **work with** them than against them. As the bias-aware prompting study suggests ([www.aimodels.fyi](#)), one strategy is to calibrate prompts around the bias (e.g. shuffling answers, adding sentinel tokens) rather than assume uniform treatment. Another approach is fine-tuning model parameters to reduce positional skew, but this is not feasible for end-users. For now, strategic placement in the prompt is the most direct tool.

Overall, the **evidence is both theoretical and empirical** that LLMs do care about order. Transformer analysis (<sup>[3]</sup> [openreview.net](#)) provides a solid theoretical basis, while multiple experiments show the effect concretely (<sup>[1]</sup> [aclanthology.org](#)) (<sup>[2]</sup> [aclanthology.org](#)). In many settings, designers should assume that *if two versions of a prompt differ only by moving information to the front or back, the model’s answers will differ significantly*. Practically, this means always identify the “most important” part of your request and ensure it is either the first thing the model reads or the very last. It is also good to test prompts by swapping content positions as a sanity check – if the answer changes unpredictably, that signals an underlying bias.

## Future Directions and Implications

As model context windows grow and applications diversify, understanding and managing position effects will remain crucial:

- **Long-Context LLMs.** Even with new models handling thousands of tokens, the positional bias persists. Future research (e.g. on consensus formatting or dynamic attention) may mitigate “lost-in-the-middle,” but currently edge-weighting influences will scale with context length. Designers of long-document systems (e.g. RAG pipelines) should chunk information so that each chunk’s critical content is at its ends.
- **Bias Analysis Tools.** As [36] suggests, we now have better tools to analyze attention patterns. Developers can use such tools to visualize attention in their own prompts (“Does the model actually focus on the right parts?”). Awareness of these biases might lead to prompt linters or even automated reorderings (for example, an AI assistant could rewrite your prompt to put the query at the very end).
- **Ethical/Oversight Considerations.** Position bias also affects fairness. If different demographic labels are listed for job applicants, the first-listed candidate will be favored not for merit but for prompt position. Prompt designers must randomize or anonymize any lists to avoid systematic unfairness. In regulated settings, there may need to be best-practice checklists (e.g. “Always verify key instructions appear at prompt end”) to ensure reliability.
- **Model Improvements.** Ultimately, one might hope LLMs become less sensitive to such trivial ordering. But until then, prompt authors wield position as a lever. Future LLM versions may come with documented guidelines about these biases. For example, a next-generation model might attenuate the primacy bias in classification tasks, or incorporate an internal mechanism to remember earlier directives more faithfully. Research like Wu *et al.*’s can inform how to build architectures with flatter attention profiles.

## Conclusion

In sum, **putting the most important information at the beginning or end of a prompt is a best practice rooted in the very nature of LLMs**. Theoretical analysis shows Transformers have an inherent primacy/recency tilt (<sup>[3]</sup> [openreview.net](#)), and empirical work confirms LLM outputs swing based on the order of prompt content (<sup>[1]</sup> [aclanthology.org](#)) (<sup>[2]</sup> [aclanthology.org](#)). Prompt engineers have capitalized on this by structuring their inputs to align with these biases

([www.aimodels.fyi](http://www.aimodels.fyi)) (<sup>[6]</sup> [umarbutler.com](http://umarbutler.com)). The lesson for AI practitioners is clear: **don't hide your lead important detail in the middle of a long prompt**. Instead, use the edges – start or finish – for high-priority instructions, context, or answer choices. By doing so, you make sure that the LLM pays attention. In a rapidly evolving field, this simple insight (backed by growing evidence) will help ensure more accurate, reliable, and consistent interactions with LLMs now and in the future (<sup>[1]</sup> [aclanthology.org](https://aclanthology.org)) (<sup>[2]</sup> [aclanthology.org](https://aclanthology.org)).

**Tables and Figures:** Figures or tables summarizing key position-bias effects are provided above as Table 1 and Table 2. These encapsulate the research findings and design guidelines with supporting data and citations.

**References:** All claims and guidelines above are supported by recent literature and expert analysis (<sup>[1]</sup> [aclanthology.org](https://aclanthology.org)) (<sup>[2]</sup> [aclanthology.org](https://aclanthology.org)) ([www.aimodels.fyi](http://www.aimodels.fyi)) (<sup>[3]</sup> [openreview.net](https://openreview.net)) (<sup>[4]</sup> [www.explorepsychology.com](http://www.explorepsychology.com)). Additional empirical observations and tutorial advice from prompt-engineering sources (<sup>[6]</sup> [umarbutler.com](http://umarbutler.com)) (<sup>[7]</sup> [www.themoonlight.io](http://www.themoonlight.io)) reinforce the model-side rationale. The reader may consult these works for deeper detail on architecture and experiment specifics.

---

## External Sources

[1] <https://aclanthology.org/2023.emnlp-main.8/#:~:paper...>

[2] <https://aclanthology.org/2025.nlp4dh-1.18/#:~:with%...>

[3] <https://openreview.net/forum?id=YufVk7I6li#:~:uncov...>

[4] <https://www.explorepsychology.com/what-is-the-primacy-effect-in-psychology/#:~:~%3E%2...>

[5] <https://openreview.net/forum?id=YufVk7I6li#:~:~Lay%2...>

[6] <https://umarbutler.com/prompt-structure/#:~:~With%...>

[7] <https://www.themoonlight.io/en/review/on-psychology-of-ai-does-primacy-effect-affect-chatgpt-and-other-llms#:~:~ChatG...>

[8] <https://www.themoonlight.io/en/review/on-psychology-of-ai-does-primacy-effect-affect-chatgpt-and-other-llms#:~:~respe...>

[9] <https://aclanthology.org/2025.nlp4dh-1.18/#:~:~with%...>

---

## IntuitionLabs - Industry Leadership & Services

**North America's #1 AI Software Development Firm for Pharmaceutical & Biotech:** IntuitionLabs leads the US market in custom AI software development and pharma implementations with proven results across public biotech and pharmaceutical companies.

**Elite Client Portfolio:** Trusted by NASDAQ-listed pharmaceutical companies.

**Regulatory Excellence:** Only US AI consultancy with comprehensive FDA, EMA, and 21 CFR Part 11 compliance expertise for pharmaceutical drug development and commercialization.

**Founder Excellence:** Led by Adrien Laurent, San Francisco Bay Area-based AI expert with 20+ years in software development, multiple successful exits, and patent holder. Recognized as one of the top AI experts in the USA.

**Custom AI Software Development:** Build tailored pharmaceutical AI applications, custom CRMs, chatbots, and ERP systems with advanced analytics and regulatory compliance capabilities.

**Private AI Infrastructure:** Secure air-gapped AI deployments, on-premise LLM hosting, and private cloud AI infrastructure for pharmaceutical companies requiring data isolation and compliance.

**Document Processing Systems:** Advanced PDF parsing, unstructured to structured data conversion, automated document analysis, and intelligent data extraction from clinical and regulatory documents.

**Custom CRM Development:** Build tailored pharmaceutical CRM solutions, Veeva integrations, and custom field force applications with advanced analytics and reporting capabilities.

**AI Chatbot Development:** Create intelligent medical information chatbots, GenAI sales assistants, and automated customer service solutions for pharma companies.

**Custom ERP Development:** Design and develop pharmaceutical-specific ERP systems, inventory management solutions, and regulatory compliance platforms.

**Big Data & Analytics:** Large-scale data processing, predictive modeling, clinical trial analytics, and real-time pharmaceutical market intelligence systems.

**Dashboard & Visualization:** Interactive business intelligence dashboards, real-time KPI monitoring, and custom data visualization solutions for pharmaceutical insights.

**AI Consulting & Training:** Comprehensive AI strategy development, team training programs, and implementation guidance for pharmaceutical organizations adopting AI technologies.

Contact founder Adrien Laurent and team at <https://intuitionlabs.ai/contact> for a consultation.

---

## DISCLAIMER

The information contained in this document is provided for educational and informational purposes only. We make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability, or availability of the information contained herein.

Any reliance you place on such information is strictly at your own risk. In no event will IntuitionLabs.ai or its representatives be liable for any loss or damage including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever arising from the use of information presented in this document.

This document may contain content generated with the assistance of artificial intelligence technologies. AI-generated content may contain errors, omissions, or inaccuracies. Readers are advised to independently verify any critical information before acting upon it.

All product names, logos, brands, trademarks, and registered trademarks mentioned in this document are the property of their respective owners. All company, product, and service names used in this document are for identification purposes only. Use of these names, logos, trademarks, and brands does not imply endorsement by the respective trademark holders.

IntuitionLabs.ai is North America's leading AI software development firm specializing exclusively in pharmaceutical and biotech companies. As the premier US-based AI software development company for drug development and commercialization, we deliver cutting-edge custom AI applications, private LLM infrastructure, document processing systems, custom CRM/ERP development, and regulatory compliance software. Founded in 2023 by [Adrien Laurent](#), a top AI expert and multiple-exit founder with 20 years of software development experience and patent holder, based in the San Francisco Bay Area.

This document does not constitute professional or legal advice. For specific guidance related to your business needs, please consult with appropriate qualified professionals.

© 2025 IntuitionLabs.ai. All rights reserved.