

What LLM Benchmarks Actually Say About Prompt Format?

Author: Kevin Anderson

Date: 03/14/2026

Topic: A Whitepaper on Markdown, JSON, XML, YAML, Plain Text, and the Measurable Role of Structure in Language-Model Performance

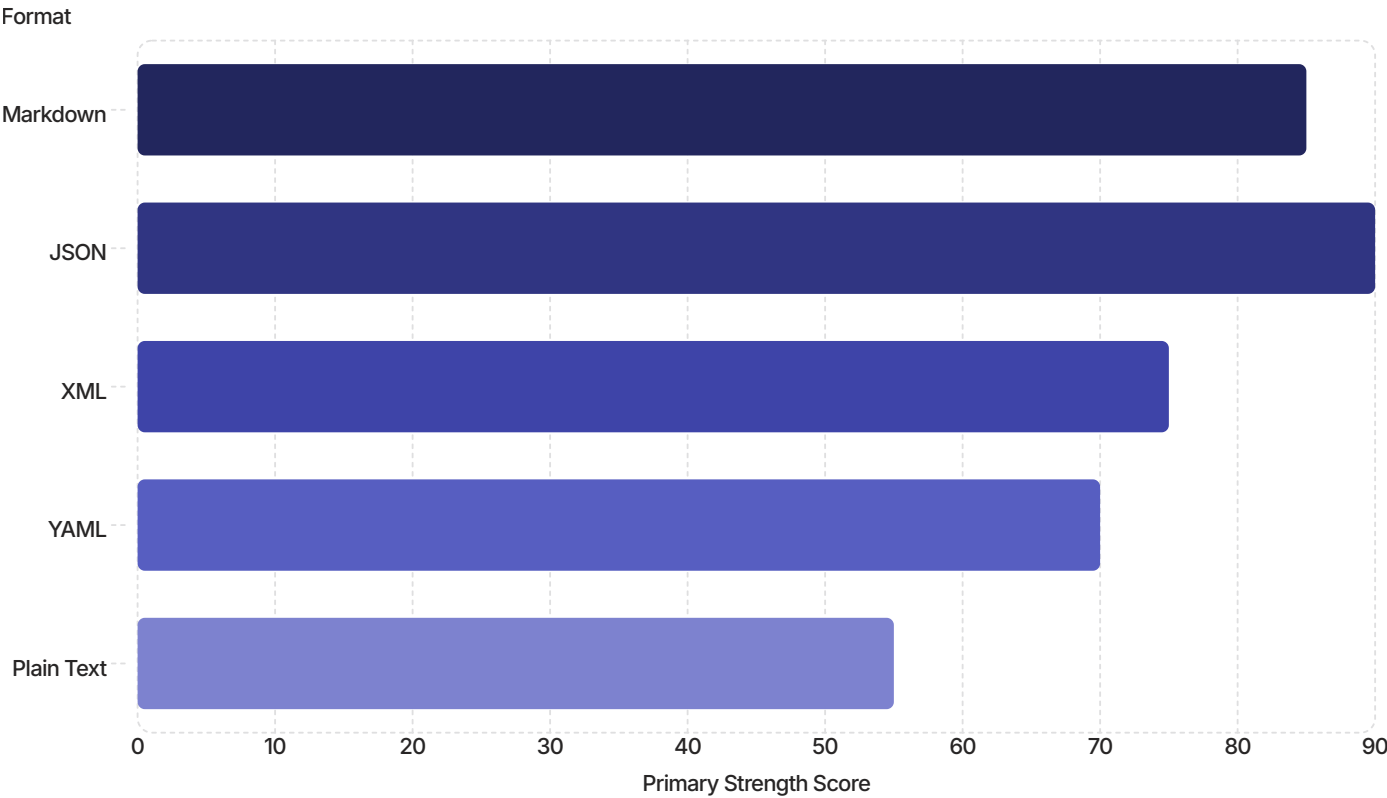
Abstract

Prompt format is often treated as a stylistic choice. The benchmark literature suggests otherwise. Across recent studies, the way a prompt or output is structured can measurably affect model behavior, including task accuracy, structural compliance, readability, and workflow reliability. The strongest direct evidence comes from a 2024 study showing that semantically equivalent prompts formatted as plain text, Markdown, JSON, and YAML can produce materially different results across reasoning, code generation, and translation tasks, with especially large variance in smaller GPT models.

The benchmark landscape also shows that "format quality" is no longer a vague concept. It has been split into distinct, measurable capabilities. MDEval evaluates Markdown Awareness as a readable-structure capability for human-facing outputs. JSONSchemaBench evaluates constrained decoding and schema compliance for structured outputs. LLMStructBench evaluates structured data extraction and valid JSON generation across many models and prompting strategies. Together, these benchmarks make one point clear: format is part of system performance, not just presentation.

This paper argues for a practical conclusion rather than a universal winner. Benchmarks do not show that one format is best everywhere. They show that different formats optimize different failure modes. Markdown tends to support readability, long-form organization, and human-model collaboration. JSON tends to support schema fidelity and machine consumption. XML tends to support explicit segmentation in complex prompts. YAML sits between document and config, often useful when prompts are manifest-like rather than narrative. Plain text remains effective for simple interaction, but usually loses ground as complexity and reuse increase.

The central thesis of this synthesis is therefore simple: **benchmark evidence does not justify format dogma; it justifies format-task alignment.** In production LLM systems, format choice should be treated as an engineering variable and evaluated against the actual objective function of the workflow, especially when speed and quality are the primary decision criteria.



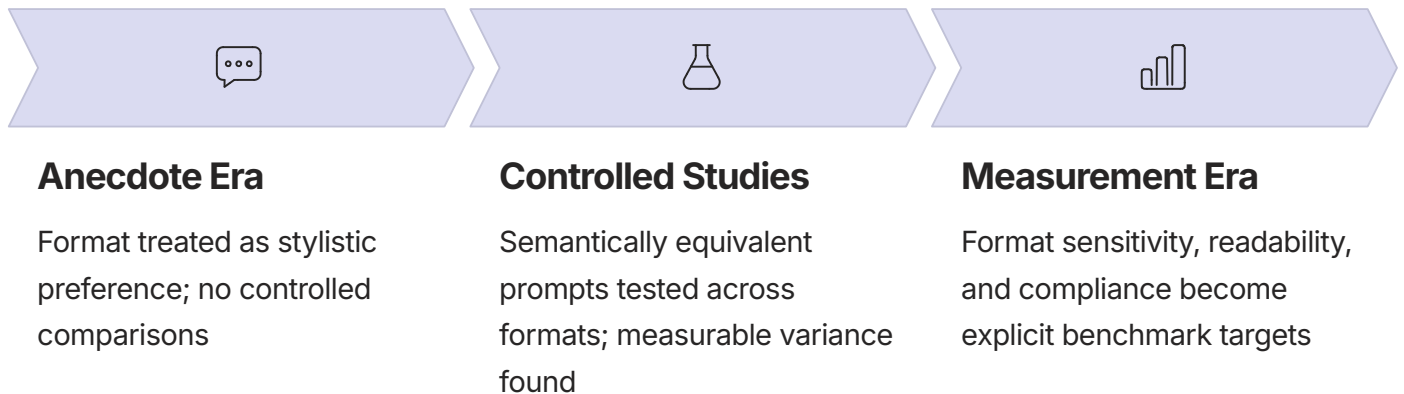
Each format optimizes a different failure mode. The chart above illustrates relative primary strength scores across the five formats discussed in this whitepaper, based on benchmark synthesis.

1. Introduction

As LLM systems mature, the industry is moving past the idea that prompting is only about "wording." In practice, modern prompts often include instructions, context, examples, policies, constraints, retrieved passages, and output requirements. Once prompts reach that level of complexity, structure becomes part of the interface itself. OpenAI's prompt engineering documentation explicitly recommends using Markdown and XML to help the model understand logical boundaries in prompts and context data. Anthropic's prompting guidance similarly recommends XML tags to separate instructions, context, examples, and variable inputs when prompts become complex.

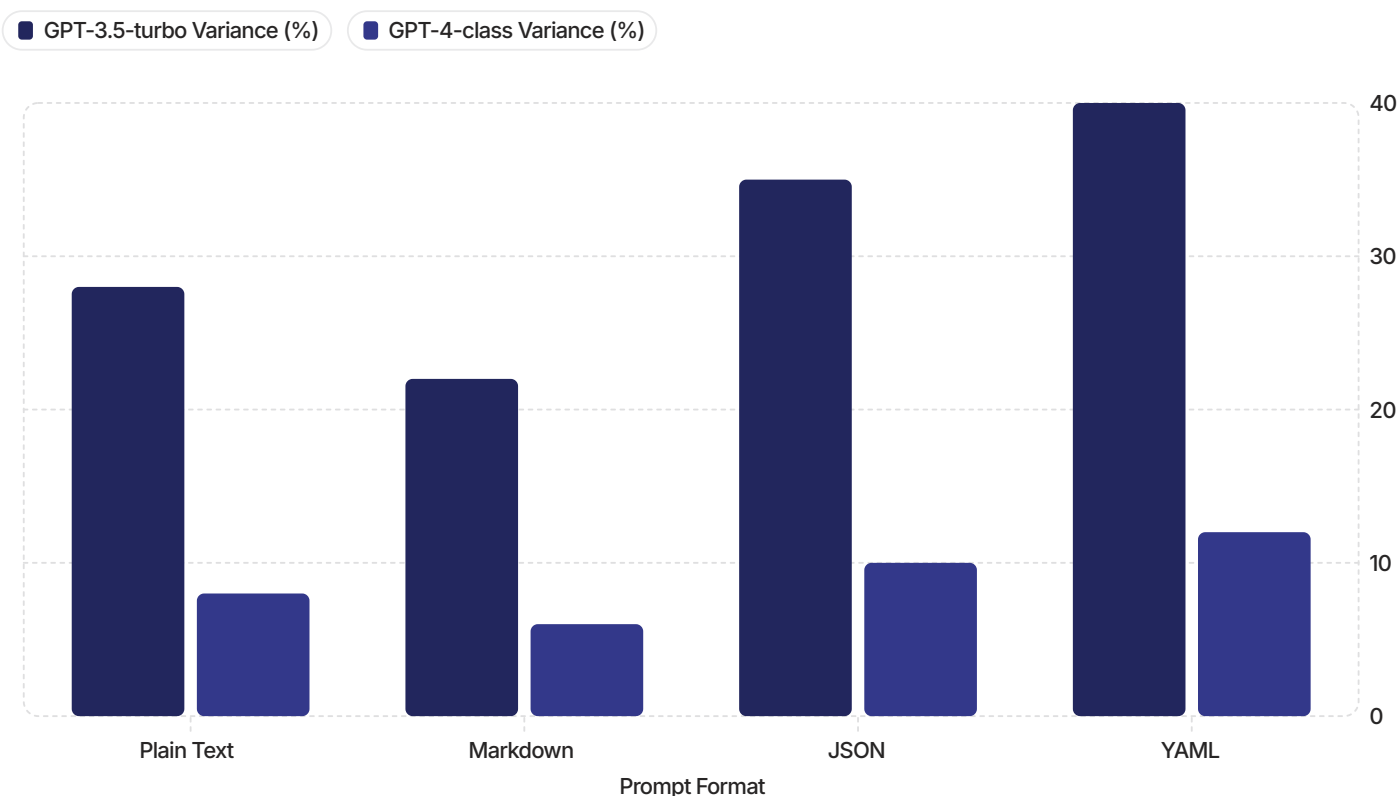
This matters because **prompt failures are often structural failures**. A model may confuse instructions with examples, treat source material as policy, or ignore constraints that are buried in prose. These are not always failures of intelligence. They are often failures of boundary clarity. That is why format comparisons such as Markdown versus JSON, Markdown versus XML, Markdown versus YAML, and Markdown versus plain text are not merely aesthetic debates. They are debates about how to reduce ambiguity inside the context window.

The benchmark literature now supports taking that problem seriously. Format sensitivity, Markdown readability, structured output reliability, and format-following all appear as explicit evaluation targets in recent papers. The field has effectively moved from anecdote to measurement.



2. The Benchmark Shift: From "Looks Better" to "Performs Differently"

The most important turning point in this area is the move from intuition to controlled comparison. The paper *Does Prompt Formatting Have Any Impact on LLM Performance?* tested semantically equivalent prompts in plain text, Markdown, JSON, and YAML across six benchmarks using GPT-family models. Its main conclusion is direct: prompt format can significantly affect performance, and the effect varies by task and model. In one highlighted case, GPT-3.5-turbo varied by up to **40 percent** on a code translation task depending on the prompt template, while GPT-4-class models were more robust but still not format-invariant.



Illustrative performance variance by prompt format across GPT model families on a code translation task. GPT-3.5-turbo showed up to 40% variance depending on template, while GPT-4-class models were more robust but still not format-invariant.

That result matters more than any single head-to-head winner. It means format is an experimental variable. A system benchmarked in one format and deployed in another may not preserve its performance characteristics. It also means that "the best prompt" is incomplete as a concept unless it specifies the structural template in which the prompt is expressed.

A second important shift is that output formatting itself is now being benchmarked as a standalone capability. MDEval frames Markdown Awareness as an overlooked but important dimension because readable structure directly affects how useful chatbot outputs are to humans. JSONSchemaBench and LLMStructBench do something similar on the structured-output side by treating valid JSON and schema compliance as benchmark-worthy tasks in their own right. These papers collectively show that output structure is no longer a secondary concern; it is part of the quality surface of modern LLM systems.

3. What the Main Benchmarks Measure

- 1

Prompt-Format Sensitivity

The 2024 formatting paper is still the clearest broad benchmark for the core question of whether human-readable formatting changes model behavior. Its value lies in breadth rather than in choosing a universal champion. It demonstrates that plain text, Markdown, JSON, and YAML can all lead to different outcomes under otherwise equivalent semantic conditions. The paper therefore supports a meta-conclusion: **format should be benchmarked alongside wording, examples, and model choice.**
- 2

Markdown Awareness (MDEval)

MDEval measures whether LLMs can produce readable, structurally appropriate Markdown. It introduces a 20,000-instance benchmark spanning English and Chinese, and reports strong alignment with human judgment, including a Spearman correlation of 0.791 and accuracy of 84.1 percent. The key takeaway is conceptual as much as numerical: Markdown readability is treated as a genuine model capability, not just a UI nicety.
- 3

Structured-Output Compliance (JSONSchemaBench)

JSONSchemaBench focuses on constrained decoding over 10,000 real-world JSON schemas and evaluates approaches across efficiency, coverage, and output quality. Its importance is that it grounds the structured-output debate in real schemas and real operational constraints, rather than toy examples. This is especially relevant for production systems where exact machine-readable output matters more than explanatory elegance.
- 4

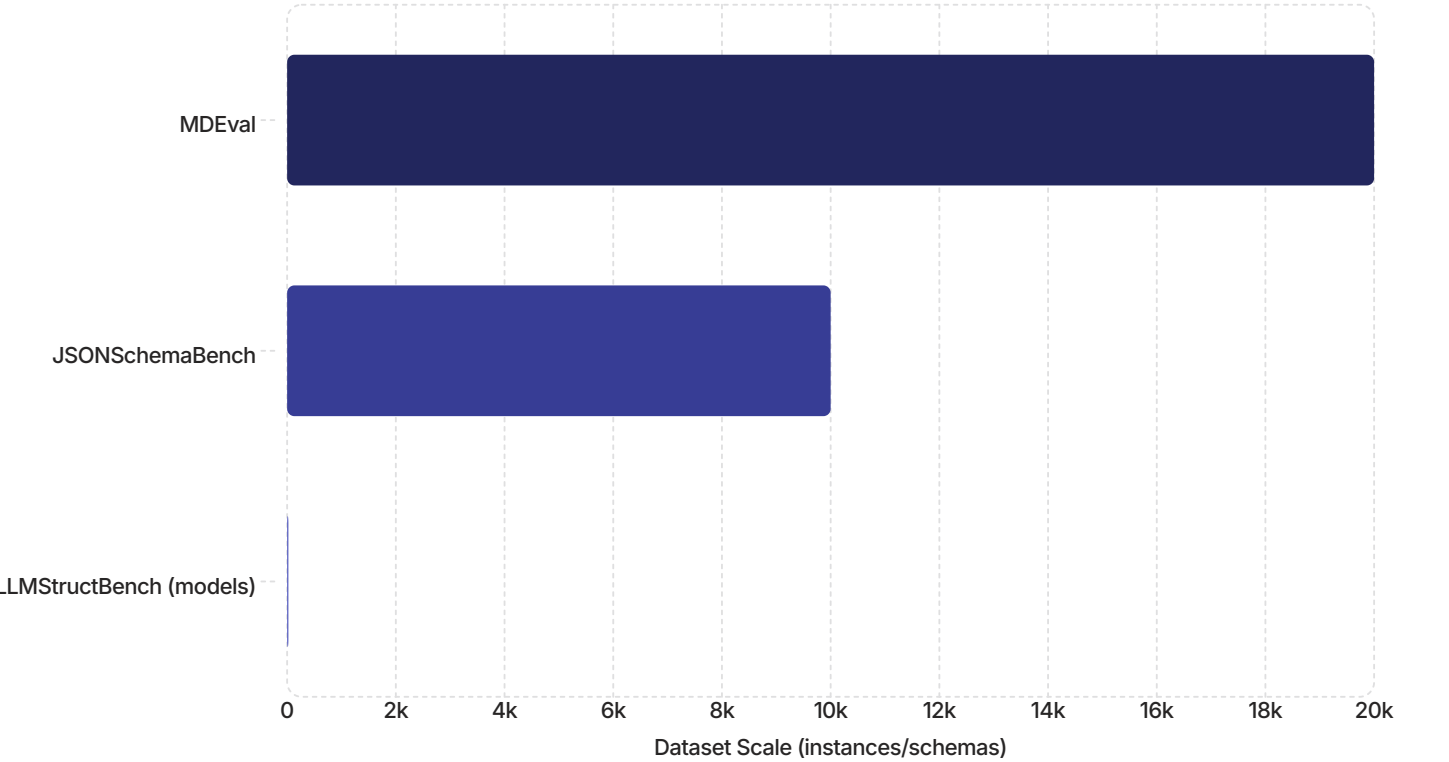
Structured Extraction & Valid JSON (LLMStructBench)

LLMStructBench extends the structured-output evaluation focus into 2026 by benchmarking structured data extraction and valid JSON generation across 22 models and five prompting strategies. The paper reinforces that reliable structured generation is difficult enough, and important enough, to warrant dedicated evaluation at scale.
- 5

Format-Following as a General Capability

Other work broadens the lens further. FoFo studies real-world format-following as a general capability, while StructEval benchmarks model ability to generate structured formats across multiple representations. These papers support a larger interpretation of the field: formatting behavior is becoming part of the core competency profile of advanced language models.

Benchmark



Scale of the three primary benchmarks discussed. MDEval covers 20,000 instances, JSONSchemaBench covers 10,000 real-world schemas, and LLMStructBench spans 22 models and five prompting strategies.

4. What the Benchmarks Say About Markdown

The benchmark evidence does not say that Markdown is universally best. It says something more precise: **Markdown is especially strong when the target output is meant for humans to read, navigate, edit, or publish.** MDEval makes that point explicit by tying Markdown quality to readability and output structure. In human-facing systems such as chat assistants, documentation agents, content workflows, and research synthesis tools, that matters directly.

OpenAI's official guidance supports this practical role for Markdown. The prompt engineering guide states that Markdown formatting can help models understand logical boundaries, and the Custom GPT instruction guidelines say that Markdown and structured formatting can improve readability and precision. GPT-5 guidance also notes that hierarchical Markdown answers can be induced reliably with explicit prompting, even though Markdown is not always the default output style in the API.

MDEval Key Metrics

0.791

**Spearman
Correlation**

Alignment with human
judgment

84.1%

Accuracy

On Markdown
awareness benchmark

Where Markdown Wins

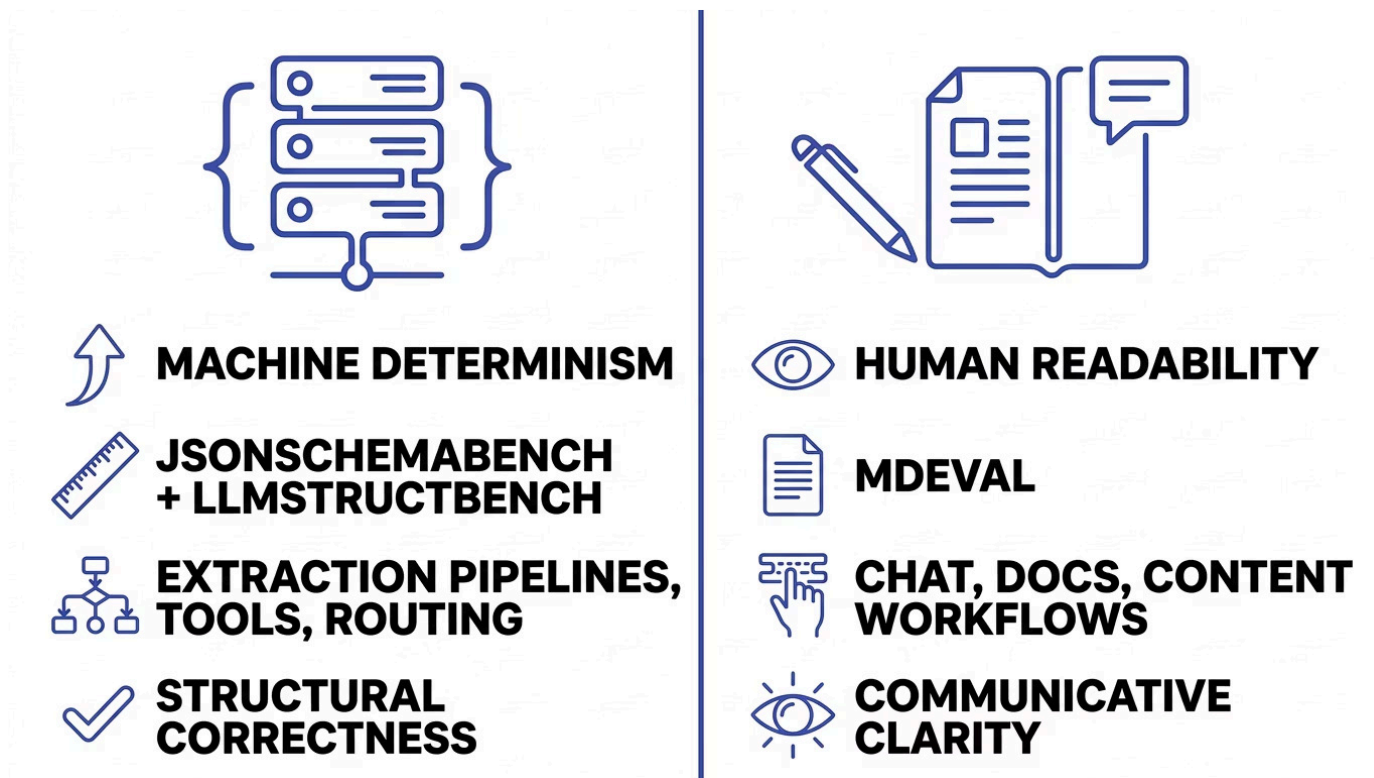
- Chat assistants and documentation agents
- Content workflows and research synthesis
- Long prompts with mixed content
- Collaborative review workflows
- Human-facing output organization

So the benchmark-backed case for Markdown is not "it always improves raw task accuracy." It is that Markdown often improves the conditions under which high-quality human-facing output is produced. It helps organize long prompts and outputs, reduces ambiguity in mixed-content tasks, and lowers review friction in collaborative workflows.

5. What the Benchmarks Say About JSON

If Markdown's strength is human-facing structure, JSON's strength is machine-facing determinism. JSONSchemaBench and LLMStructBench both reinforce that reliable structured output is difficult and operationally central. These benchmarks matter because many real applications do not merely want a good answer; they want valid fields, known keys, parsable nesting, and schema-constrained behavior.

OpenAI's Structured Outputs guidance aligns cleanly with this benchmark trend. It recommends schema-constrained output when exact structure is required, and explicitly positions this capability as the solution to missing keys, invalid enum values, and unreliable machine consumption. That is a different objective function from Markdown readability, and benchmarks show it deserves separate evaluation.



The broader lesson is that JSON tends to win when quality is defined by structural correctness rather than communicative clarity. In extraction pipelines, tool invocation, workflow routing, and application logic, a terse valid JSON object can be higher quality than a beautifully organized Markdown answer if the latter still requires cleanup or parsing heuristics.

6. What the Benchmarks Suggest About XML

XML is less represented in public benchmark literature than JSON or Markdown, but vendor guidance gives it strong practical credibility. Anthropic's documentation explicitly recommends XML tags for complex prompts because they help separate instructions, context, examples, and variable inputs unambiguously. The same guidance recommends consistent tag names and hierarchical nesting when the content has natural structure.

What XML Addresses

What does that mean in benchmark terms? It suggests XML is best understood as a **control-oriented format** rather than a general-purpose one. XML can reduce a class of failures related to prompt-region confusion, especially in templated or multi-component prompts. That is not identical to the tasks emphasized in JSONSchemaBench or MDEval, but it addresses a real structural problem that those benchmarks do not fully capture.

The safest conclusion is that XML currently has stronger support from vendor prompting practice than from broad academic head-to-head benchmarks. Even so, the overall benchmark trend toward format sensitivity makes that guidance more credible, not less. If structure affects performance, then explicit prompt segmentation has a real theoretical basis.

XML Prompt Segmentation

- **Instructions** — clearly delimited from context
- **Context** — separated from examples
- **Examples** — isolated from variable inputs
- **Variable Inputs** — unambiguously tagged

Anthropic explicitly recommends this pattern for complex, multi-component prompts.

Vendor Credibility

Anthropic's documentation explicitly recommends XML tags for complex prompts — stronger vendor support than academic benchmarks currently provide.

Theoretical Basis

If structure affects performance (as benchmarks confirm), then explicit prompt segmentation via XML has a real, defensible theoretical basis.

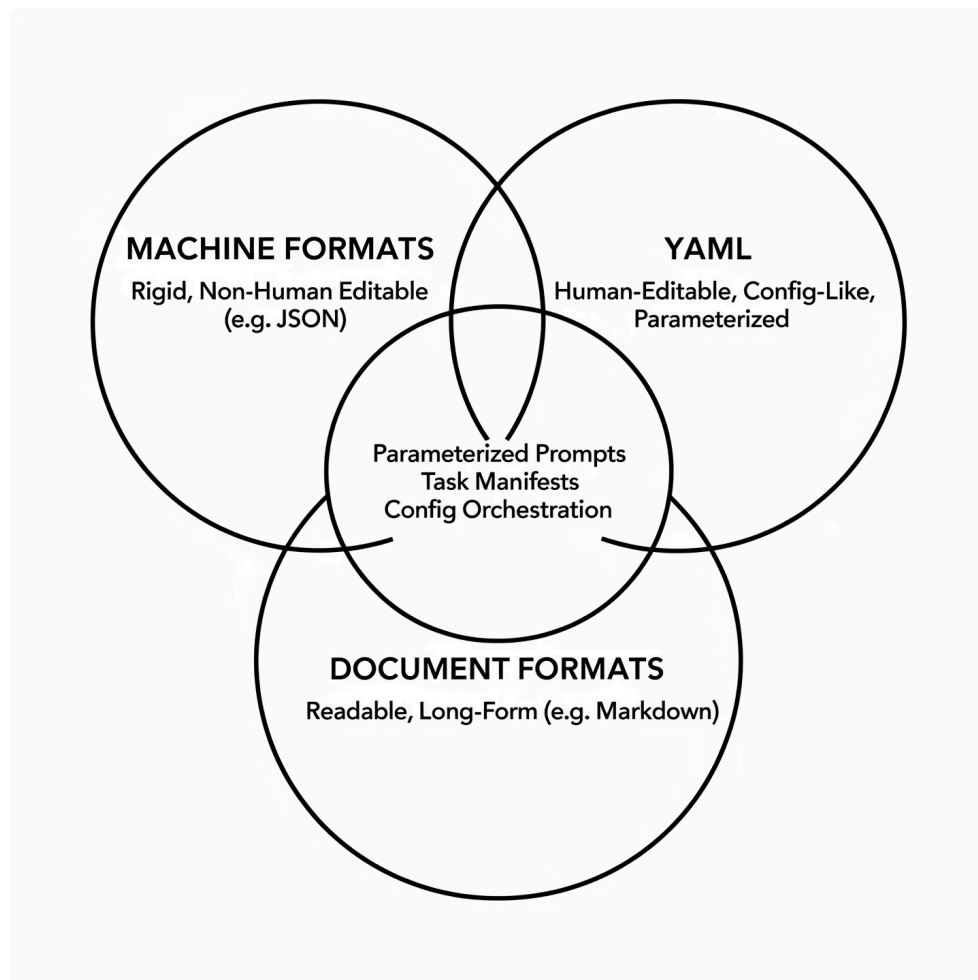
Failure Mode Addressed

XML reduces prompt-region confusion in templated or multi-component prompts — a failure class not fully captured by JSONSchemaBench or MDEval.

7. What the Benchmarks Suggest About YAML

YAML appears in the 2024 prompt-format study as one of the major human-readable alternatives to plain text, Markdown, and JSON. Its inclusion is important because it shows that YAML is not merely a software-config artifact in this conversation; it is a meaningful prompt template style whose effects can differ from the other formats.

The literature does not yet support a blanket statement that YAML is broadly better or worse than Markdown. What it does support is that YAML becomes useful when the prompt behaves more like a manifest or configuration than a document. FoFo and StructEval reinforce that structured format following matters across representations, which strengthens the case for evaluating YAML not just as a serialization language but as a prompting surface.

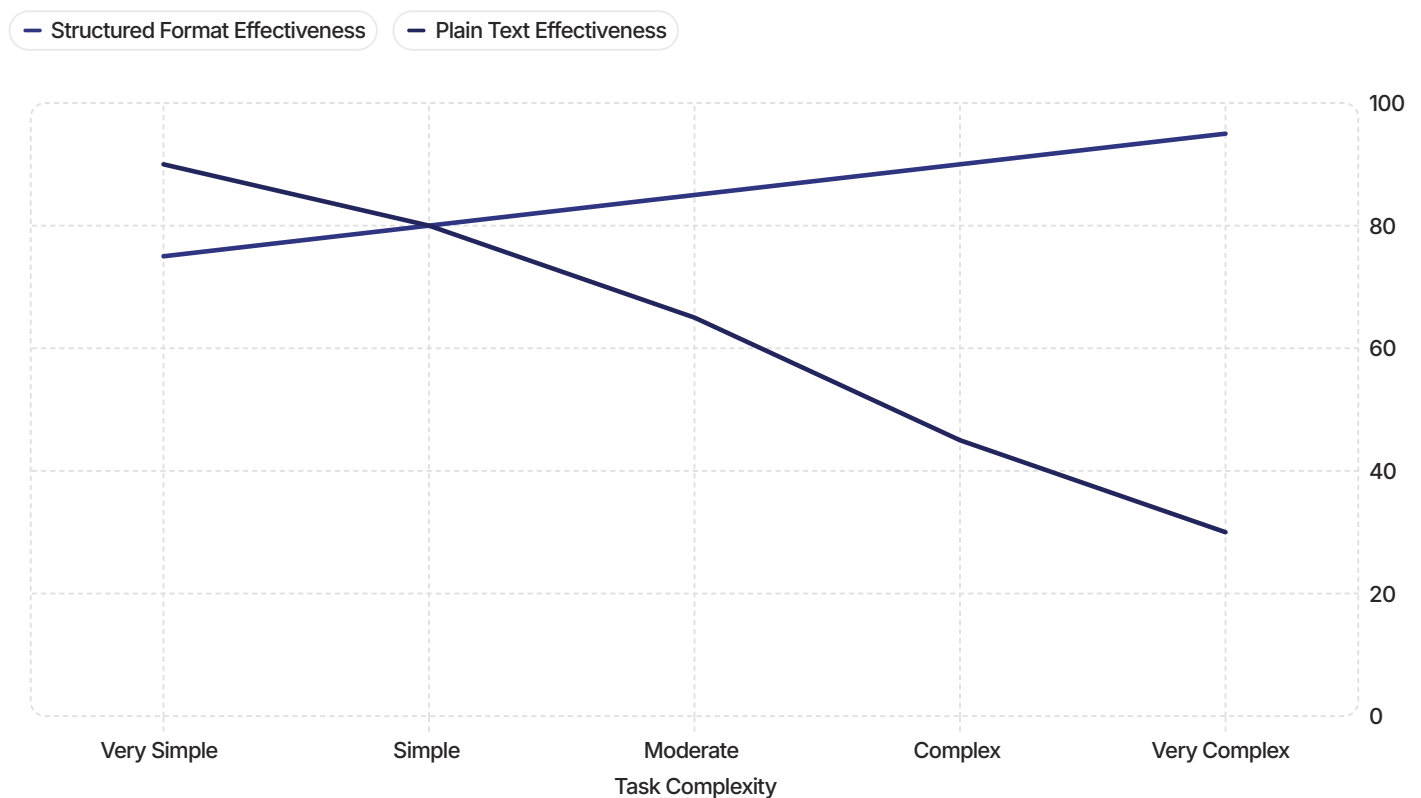


In practical terms, YAML seems to occupy a middle position. It is more human-editable than rigid machine formats, but less naturally readable than Markdown for long-form, mixed-content work. That implies a natural niche: **parameterized prompt templates, declarative task manifests, and config-like orchestration.**

8. What the Benchmarks Say About Plain Text

Plain text remains the baseline, and that matters. It is the lowest-friction format and often works well for short, simple interactions. The benchmark literature does not say plain text is useless. It says plain text loses its advantage as task complexity grows and as the system needs clearer boundaries between instruction types, context blocks, and output requirements.

In other words, plain text remains strong for immediacy, but weakens as prompts become assets rather than one-off messages. This is why official guidance increasingly encourages structured formatting. OpenAI's documentation does not tell developers to abandon plain language; it tells them to add logical boundaries when the workflow needs them.



As task complexity increases, plain text effectiveness declines while structured formats gain a decisive advantage. Plain text remains competitive only at the lowest complexity levels.

- ❏ The benchmark-backed interpretation is therefore conditional: **plain text is often good enough for simple tasks, but it is usually not the strongest default for complex, reusable, or collaborative prompting.**

9. Speed: What the Benchmarks Actually Support

The user-facing question of speed needs careful handling. The current benchmark literature supports the claim that format affects performance. It does not support a universal claim that one format is always fastest across raw model latency, time to first token, or total end-to-end completion.

What the literature does support is a more nuanced view of speed. JSON can reduce downstream latency and cleanup in schema-bound systems because it is easier to parse and validate. Markdown can reduce human drafting, review, and revision time in document-like workflows because it improves readability and navigability. XML can reduce debugging time in complex prompts when the main failure mode is boundary confusion. YAML can simplify config-driven orchestration when prompts are field-oriented.



JSON

Reduces downstream latency and cleanup in schema-bound systems — easier to parse and validate automatically.



Markdown

Reduces human drafting, review, and revision time in document-like workflows via improved readability and navigability.



XML

Reduces debugging time in complex prompts when the main failure mode is boundary confusion between prompt regions.



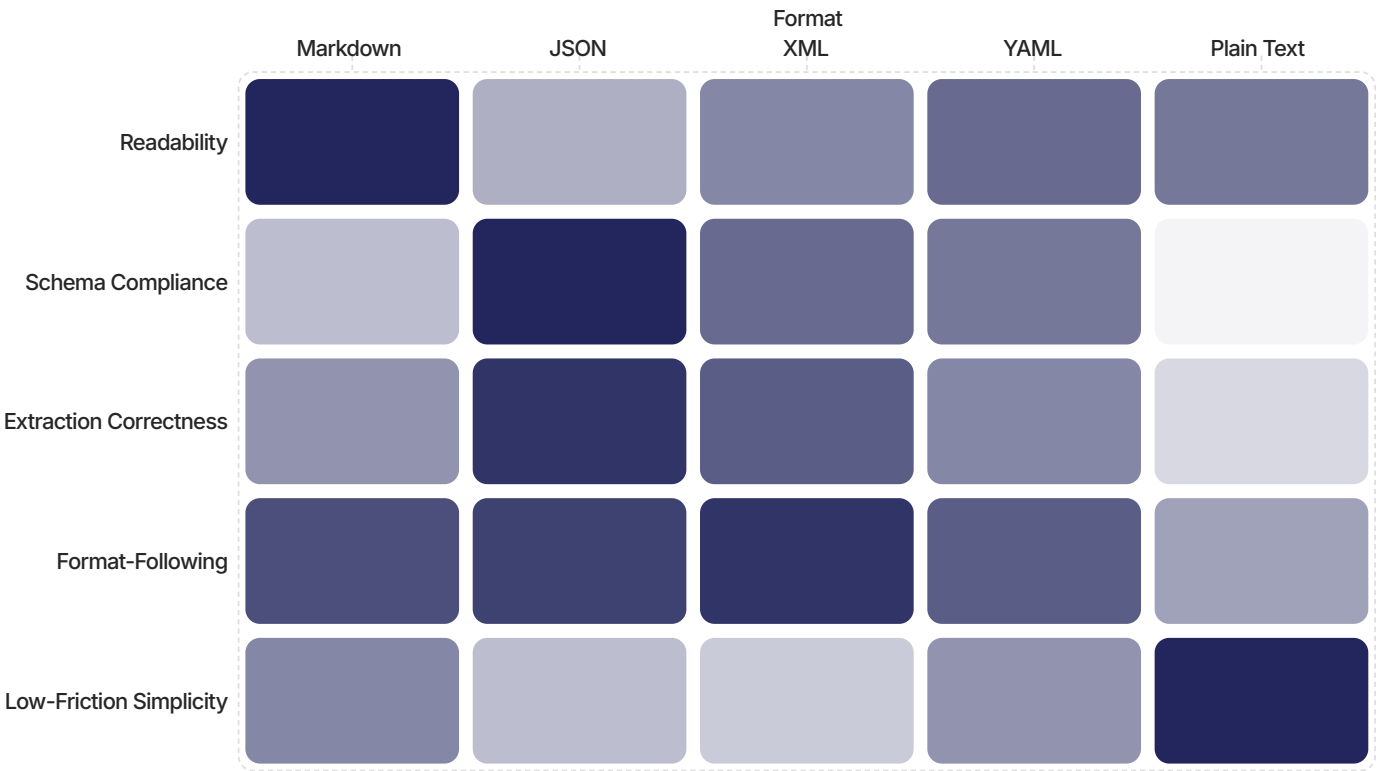
YAML

Simplifies config-driven orchestration when prompts are field-oriented and behave like manifests.

The benchmark-consistent answer is not "format X is fastest." It is "the fastest format depends on where the bottleneck is: model generation, machine parsing, human review, or prompt maintenance." That is a more defensible and more useful interpretation of the evidence.

10. Quality: What the Benchmarks Actually Support

The same caution applies to quality. Benchmarks do not identify one master format that dominates across all dimensions. They separate quality into different measurable properties: semantic accuracy, readability, schema compliance, extraction correctness, and format-following consistency.



This heatmap illustrates how each format scores across five distinct quality dimensions. Darker cells indicate higher performance. No single format dominates all dimensions — each excels in its own domain.

This leads to the most important conclusion in the whole literature: **format choice is inseparable from the definition of quality in the task**. If quality means readable, well-organized, human-facing output, Markdown gets stronger support. If quality means exact structured compliance, JSON gets stronger support. If quality means clear separation of heterogeneous prompt regions, XML becomes attractive. If quality means lightweight declarative configuration, YAML gains relevance. If quality means low-friction immediate interaction, plain text still has a role.

- ❏ The benchmark synthesis therefore points away from one-dimensional scorekeeping and toward **format-task matching**. That is the core pattern across the literature.

11. Recommended Evaluation Framework for Teams & Conclusion

Recommended Evaluation Framework for Teams

Teams building real LLM workflows should stop asking only "which format is best?" and start asking "best for which task family and failure mode?" A sound internal benchmark should test formats across at least five categories: short direct interaction, long-form explanation, retrieval-based synthesis, structured extraction, and reusable multi-part prompt templates. These categories expose the tradeoffs surfaced by the literature.

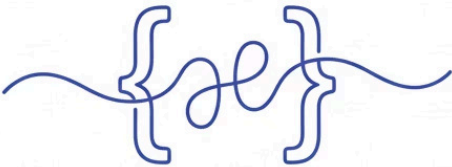
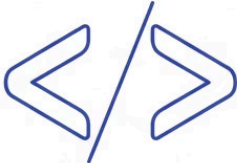
Speed Metrics • Model latency • Token count • Parser success rate • Human drafting time • Human revision time	Quality Metrics • Semantic correctness • Readability • Structural compliance • Consistency • Downstream task success	Task Families to Test • Short direct interaction • Long-form explanation • Retrieval-based synthesis • Structured extraction • Reusable multi-part templates
--	---	---

The strongest internal hypothesis is not that one format will win globally. It is that different formats will dominate in different layers of the workflow. That expectation is exactly what the current benchmark literature supports.

Conclusion

The benchmark evidence is now strong enough to reject a casual view of prompt formatting. Format is not decoration. It is part of the performance envelope of LLM systems. The literature shows that prompt templates can materially change results, that Markdown readability is measurable, that structured-output reliability deserves its own benchmarks, and that format-following itself is an emerging capability area.

At the same time, the benchmarks do not justify simplistic format tribalism. They do not show that Markdown, JSON, XML, YAML, or plain text is universally best. They show that each format is a tool for optimizing a different kind of structure. Markdown is strongest as a human-readable workspace. JSON is strongest as a machine-readable contract. XML is strongest when prompt regions must be sharply delimited. YAML is strongest when prompts resemble manifests or configs. Plain text is strongest when simplicity is enough.

 MARKDOWN: HUMAN WORKSPACE	 JSON: MACHINE CONTRACT	 XML: PROMPT DELIMITER
 YAML: MANIFEST CONFIG	 PLAIN TEXT: SIMPLICITY BASELINE	

The most defensible production conclusion is therefore architectural: **use the format that matches the failure mode you most need to prevent.** That is what the benchmarks actually say.

References

He, J., Rungta, M., Koleczek, D., Sekhon, A., Wang, F. X., and Hasan, S. **Does Prompt Formatting Have Any Impact on LLM Performance?** arXiv:2411.10541.

Chen, Z., Liu, Y., Shi, L., Wang, Z.-J., Chen, X., Zhao, Y., and Ren, F. **MDEval: Evaluating and Enhancing Markdown Awareness in Large Language Models.** arXiv:2501.15000.

Geng, S., Cooper, H., Moskal, M., Jenkins, S., Berman, J., Ranchin, N., West, R., Horvitz, E., and Nori, H. **Generating Structured Outputs from Language Models: Benchmark and Studies.** arXiv:2501.10868.

Tenckhoff, S., Koddenbrock, M., and Rodner, E. **LLMStructBench: Benchmarking Large Language Model Structured Data Extraction.** arXiv:2602.14743.

OpenAI. **Prompt engineering.**

OpenAI. **Key Guidelines for Writing Instructions for Custom GPTs.**

Anthropic. **Prompting best practices — Structure prompts with XML tags.**

Anthropic. **Console prompting tools — prompt templates and variables.**

FoFo. **A Benchmark to Evaluate LLMs' Format-Following Capability.**

StructEval. **Benchmarking LLMs' Capabilities to Generate Structured Formats.**