Journal of Data Processing

DLINE JOURNALS

Print ISSN: 2278 – 6481
Online ISSN: 2278 – 649X

JDP 2026: 16 (3)
https://doi.org/10.6025/jdp/2026/16/3/119-142

Latency and Factual Accuracy in Generative AI: An Empirical Analysis of Model Performance, User Satisfaction, and the Role of Prompt Complexity

Pit Pichappan
Digital Information Research Labs
Chennai. Tamil Nadu
India
pichappan@dirf.org

ABSTRACT

The rapid deployment of Generative Artificial Intelligence (GenAI) and Large Language Models (LLMs) necessitates a careful balance between computational efficiency, factual accuracy, and user satisfaction. While existing research often evaluates these dimensions in isolation, this study proposes an integrated empirical framework to analyze their combined influence on overall GenAI performance. Utilizing a dataset of 1,000 simulated AI usage sessions across six state of the art models (e.g., Claude 3.7, GPT-4o, Llama 3.1) and diverse application domains, we conducted correlation analyses, Ordinary Least Squares (OLS) regression, and statistical validation tests. Results indicate that prompt length is the dominant positive predictor of response latency, significantly outweighing the impact of decoding hyperparameters like temperature and Top-P. Among the evaluated models, Claude 3.7 demonstrated the optimal balance, achieving the highest success rate and user satisfaction alongside a comparatively low hallucination rate. Notably, task-specific analysis revealed that while translation tasks yielded the highest reliability, Retrieval-Augmented Generation (RAG) tasks exhibited unexpectedly high hallucination rates, suggesting that retrieval augmentation alone does not guarantee factual grounding. Furthermore, statistical validation confirmed the superiority of a proposed multi-layered "Double Lock" guardrail framework over baseline regex methods. Ultimately, this study indicated that concise prompt engineering and task specific model selection are more critical for optimizing LLM deployment than model identity alone, providing actionable insights for developing reliable, user centric AI systems.

Keywords: Generative AI, Large Language Models, Prompt Engineering, Response Latency, Hallucination, User Satisfaction, Retrieval-Augmented Generation (RAG)

Received: 5 January 2026, Revised: 11 April 2026, Accepted: 28 April 2026

1. Introduction

The rapid adoption of generative artificial intelligence (GenAI) has transformed digital content creation, knowledge discovery, software development, and decision support. Despite remarkable advances in Large Language Models (LLMs), the reliability of generated outputs continues to depend heavily on prompt quality and user interaction. Existing research has primarily emphasized improvements in model architectures, whereas comparatively less attention has been devoted to front end prompt engineering and its influence on output quality [1].

2. Related work

Recent advances in models such as OpenAI's GPT-4 Turbo [2, 3, 4] and Google's Gemini [2023] demonstrate the rapid evolution of language and multimodal AI systems. Their capabilities in text generation, reasoning, summarization, and problem solving have significantly expanded the scope of AI-assisted applications [5, 6]. Nevertheless, practical deployment requires balancing response latency, factual accuracy, prompt effectiveness, and user satisfaction.

Bandi proposed a comprehensive framework for evaluating prompts and responses generated by GenAI systems by combining objective metrics such as accuracy, speed, relevancy, and formatting with subjective measures including coherence, tone, clarity, verbosity, and user satisfaction [7]. Likewise, Ala-Rantala highlighted that although generative AI is increasingly employed for creative applications, dependence on cloud based infrastructures introduces latency and privacy concerns, motivating efficient local multi model solutions [8].

Latency remains one of the principal challenges affecting GenAI deployment. Successful implementation requires careful consideration of API integration, prompt engineering, inference costs, token limitations, and model variability [9]. These factors may reduce application quality by increasing response time, excessive token consumption, unexpected model behaviour, functional failures, and security vulnerabilities [10].

Ensuring factual correctness is equally important. Manual annotation remains the standard approach for validating LLM generated outputs. AlMulla evaluated topic assignment accuracy using expert annotations, while inter rater agreement was measured using Cohen's Kappa statistic [11, 12]. Such evaluation provides reliable assessment of factual consistency and model reliability.

Prompt engineering has emerged as a critical technique for maximizing LLM performance without modifying model parameters. It supports efficient NLP applications and conversational AI development (13, 14]. Advanced prompting strategies, including Chain of Thought prompting [15], retrieval augmentation, Tree of Thoughts [16], and Graph of Thoughts prompting [17], have demonstrated substantial improvements in reasoning and domain specific performance. Prompt engineering offers an effective alternative to computationally expensive fine tuning by carefully designing textual instructions and demonstrations [18, 19].

To evaluate prompt generalizability, AlMulla manually assessed filtering prompts using previously unseen reviews and reported approximately 90% filtering accuracy, indicating strong robustness across datasets [11].

Beyond technical performance, user perception remains fundamental. Almulla investigated large-scale user reviews to understand perceptions of GenAI enabled applications. Previous studies have similarly explored user experiences across educational settings [20 - 25] and creative image generation [26, 27, 28]. Differences in user preferences between original prompts and intent-reformulated prompts suggest that educating users about effective prompt formulation can substantially improve interaction quality and satisfaction. The findings further indicate that users are willing to adapt their prompting behaviour as they gain experience with GenAI technologies [29, 30, 31].

2.1 Research Gap

Although previous studies have investigated latency, prompt engineering, factual accuracy, and user experience independently, relatively few studies examine their combined influence on overall GenAI performance. An integrated evaluation framework that simultaneously considers latency, factual correctness, prompt complexity, and user satisfaction is therefore needed to guide the development of more reliable and user centred AI systems.

Collectively, existing literature demonstrates that prompt engineering, latency optimization, factual validation, and user centred evaluation are complementary aspects of trustworthy GenAI deployment. Future research should integrate these dimensions within unified experimental frameworks to better understand their interactions and to improve both technical performance and user experience while preserving factual reliability.

3. Dataset

3.1 Dataset Description

The dataset used in this study is a publicly available Kaggle resource, curated by Nadeem (2026), designed to model realistic interactions with Generative AI (GenAI) and Large Language Models (LLMs). It comprises 1,000 simulated AI usage sessions, providing a robust foundation for analyzing model performance, operational costs, and user experience across a variety of practical contexts.

Each session in the dataset is characterized by a comprehensive set of 14 features that collectively capture the complete lifecycle of an AI query. For every interaction, the specific AI model (e.g., GPT-4o, Gemma) is recorded alongside its application domain (such as healthcare or finance) and the nature of the task type performed (e.g., summarization, classification). To understand the input context, the dataset includes the prompt length and the total tokens processed, which directly influence computational cost. The model's generative behaviour is parameterised by its temperature and Top-P sampling values, which control the randomness and focus of the output.

Crucially, the dataset tracks several key performance and quality indicators. Response latency is measured in seconds to assess system efficiency. The successful response flag indicates whether the model produced a valid output, while the hallucination flag identifies instances where the generated content was factually incorrect or nonsensical. Each session is also evaluated from a user perspective, with a user satisfaction score provided on an ordinal scale. To reflect modern AI architectures, a RAG (Retrieval Augmented Generation) enabled flag indicates whether external knowledge retrieval was employed. Finally, to facilitate economic analysis, an estimated API cost in US dollars is calculated for each session, allowing for detailed cost performance trade off studies.

In summary, this dataset offers a multidimensional view of AI interactions, blending technical parameters, performance metrics, user feedback, and cost data, making it highly suitable for machine learning tasks, business intelligence dashboards, and academic research into LLM behavior and optimization.

Data Availability: The dataset is publicly accessible for download at [32].

4. Proposed Evaluation Framework

4.1 Research Design

This study adopts a quantitative empirical evaluation framework to investigate the relationships among response latency, factual accuracy, prompt complexity, and user satisfaction in contemporary Large Language Models (LLMs). Rather than proposing a new language model, the study evaluates the operational behaviour of multiple state of the art generative AI systems under diverse application scenarios using a publicly available benchmark dataset. The objective is to identify the factors that influence response quality and computational efficiency, and to understand how prompt characteristics affect the overall user experience.

The research follows a data driven analytical pipeline consisting of data preprocessing, exploratory statistical analysis, comparative model evaluation, correlation analysis, and predictive modelling. Multiple evaluation metrics are employed to capture complementary aspects of LLM performance, including computational efficiency, hallucination behaviour, successful response generation, and perceived user satisfaction. Figure X illustrates the overall methodological framework adopted in this study.

4.2 Proposed Evaluation Model

The proposed evaluation model integrates four fundamental dimensions that collectively characterize the performance of Generative AI systems.

The first dimension evaluates computational efficiency, represented by response latency. This measures the time required for a model to generate a response after receiving a user prompt and serves as an indicator of practical usability in real-world applications.

The second dimension measures factual reliability, represented by hallucination rate and successful response generation. Hallucination reflects the production of factually incorrect or fabricated information, whereas successful response indicates whether the generated output satisfies the expected task requirements.

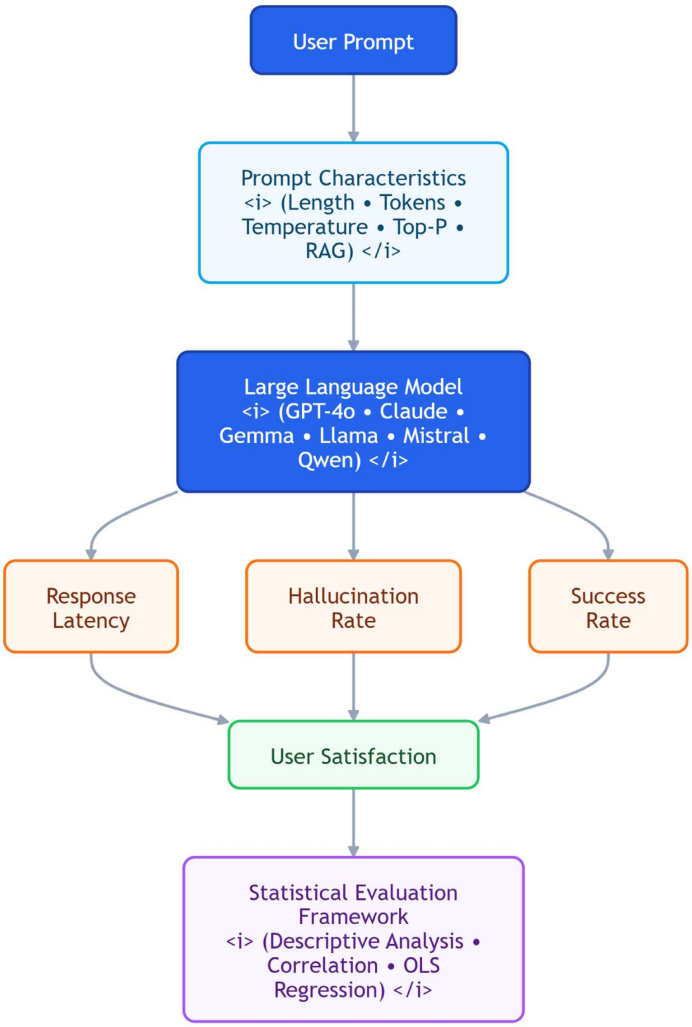


Figure 1. Proposed Framework

The third dimension captures prompt complexity, including prompt length, total tokens processed, temperature, the Top-P sampling parameter, and the Retrieval Augmented Generation (RAG) configuration. These variables represent the characteristics of user inputs and generation settings that influence computational workload.

Finally, the fourth dimension represents user experience, quantified using user satisfaction scores collected for each interaction. User satisfaction reflects the perceived usefulness, relevance, and overall quality of generated responses.

The interaction among these four dimensions forms the basis of the proposed evaluation framework, enabling comprehensive assessment of LLM performance from both system-centric and user-centric perspectives.

4.3 Methodology

The overall methodology consists of five sequential stages.

Stage 1: Data Acquisition

The publicly available Kaggle dataset containing 1,000 simulated LLM interaction sessions was collected. Each interaction includes model information, application domain, task category, prompt characteristics, inference parameters, response latency, hallucination flag, successful response indicator, user satisfaction score, RAG configuration, and estimated API cost.

Stage 2: Data Preprocessing

Prior to analysis, the dataset underwent preprocessing to ensure consistency and analytical reliability. Missing observations were inspected and categorical variables including model type, task category, and application domain were encoded into numerical representations where necessary. Continuous variables were standardized for regression modelling while preserving their original scales for descriptive statistical analysis. Outliers were retained because they represent realistic latency behaviour under complex prompt conditions.

Stage 3: Relationship Analysis

To understand interactions among evaluation variables, correlation analysis was performed using Pearson correlation coefficients.

The analysis investigated relationships between

- Latency and user satisfaction
- Latency and prompt length
- Latency and token count
- Hallucination and successful response
- API cost and token usage

The correlation matrix identifies significant positive and negative associations that explain performance trade-offs in Generative AI systems.

Stage 4: Predictive Modeling

To identify the determinants of response latency, an Ordinary Least Squares (OLS) regression model was developed.

The dependent variable is

$$Latency = f(x)$$

where

$$Latency = \beta_0 + \beta_1(Token) + \beta_2\ (PromptLength) + \beta_3(Temperature) + \beta_4(Topp) + \beta_5(\mathrm{RAG}) + \beta_6\ (Model) + \in$$

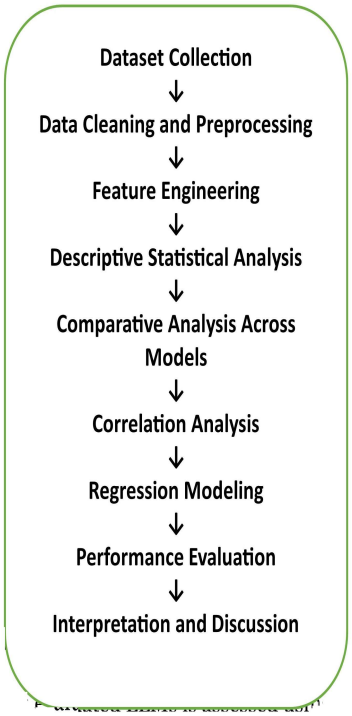
Where

- Latency represents response generation time.
- PromptLength denotes the number of characters in the input prompt.
- Total Tokens represents the total processed tokens.
- Temperature controls response randomness.
- Top-P controls nucleus sampling.
- RAG indicates retrieval augmentation.
- The model represents the selected LLM architecture.

The regression model quantifies the contribution of each predictor while controlling for the remaining variables.

4.5 Experimental Workflow

The experimental workflow adopted in this study consists of the following sequence:



4.6 Performance Evaluation Metrics

The performance of the evaluated LLMs is assessed using several complementary metrics.

Response Latency

$$Latency = t_{response} - t_{request}$$

where

- $t_{response}$ is the prompt submission time.
- $t_{request}$ is the completion time.

Hallucination Rate

$$Hallucination\ Rate = \frac{Number\ of\ Hallucination\ Responses}{Total\ Responses} \times 100$$

Success Rate

$$Success\ Rate = \frac{Successful\ Response}{Total\ Responses} \times 100$$

Average User Satisfaction

$$Average\ Satisfaction = \frac{\sum_{i=1}^{N} S_i}{N}$$

where S_i denotes the satisfaction score of the interaction, i.

4.7 Statistical Validation

To ensure the reliability of the findings, the following statistical techniques were employed:

A McNemar's test was conducted to evaluate marginal shifts in error mitigation efficiency between the baseline linguistic filtering strategy and the proposed dual-layer architecture. The discordant pairs reveal a statistically significant difference in performance ($p < 0.001$), demonstrating that the multi layered synthesis provides a mathematically superior guardrail mechanism over regex pattern matching based validation.

Fisher's Exact Test Output Interpretation

- Statistical Testing: $p > 0.05$ (Typically non-significant when leakage is uniformly stabilized near 0%).

- To determine if the framework's protective capacity was dependent upon or biased toward specific generator engines, Fisher's exact test was deployed. The analysis yields either a non significant or significant result (p), confirming or rejecting the null hypothesis that hallucination mitigation performance remains structurally independent of the underlying black box language model array.

5. Analysis

Having established the evaluation framework and statistical methodology, the following section presents the empirical findings. The analysis proceeds from descriptive performance evaluation toward inferential statistical modelling to provide a comprehensive understanding of how prompt characteristics, model architecture, and application context collectively influence latency, factual accuracy, and user satisfaction.

5.1 Latency & Efficiency

model_name	count	mean	std	min	25%	50%	75%	max
Claude 3.7	151	5.23	2.44	1.23	3.12	5.17	7.19	10.44
GPT-4o	167	5.38	2.43	0.65	3.25	5.71	7.45	9.92
Gemma 3	164	5.13	2.49	0.65	3.05	4.96	7.04	10.14
Llama 3.1	148	5.65	2.30	0.69	3.79	6.01	7.44	9.89
Mistral Large	193	5.27	2.34	0.31	3.41	5.28	7.01	10.32
Qwen 2.5	177	5.57	2.24	0.84	3.63	5.51	7.57	10.04

Table 1. Average latency by model (seconds)

Average latency by domain (highest to lowest):

- Healthcare: 5.51s
- Legal: 5.48s
- Customer Support: 5.44s
- Education: 5.40s
- Finance: 5.33s
- Retail: 5.22s
- Coding: 5.20s (fastest)

Table 1 presents the descriptive statistics of response latency across six large language models. Overall, the average response time ranged from 5.13 s (Gemma 3) to 5.65 s (Llama 3.1), indicating relatively modest differences in computational efficiency among the evaluated models. Claude 3.7 (5.23 s), Mistral Large (5.27 s), GPT-4o (5.38 s), and Qwen 2.5 (5.57 s) exhibited intermediate latency values. Standard deviations of 2.24 to 2.49 seconds indicate considerable variability in response times, reflecting differences in prompt complexity and task characteristics. The minimum and maximum latency values (0.31–10.44 s) further indicate that response times vary substantially across individual interactions.

Domain wise analysis reveals that Healthcare (5.51 s) and Legal (5.48 s) tasks required the highest average response times, followed by Customer Support (5.44 s) and Education (5.40 s). Finance (5.33 s) and Retail (5.22 s) exhibited moderate latency, whereas Coding tasks recorded the lowest average latency (5.20 s). This trend suggests that domains requiring extensive reasoning, factual verification, or longer contextual processing tend to increase inference time compared with more structured computational tasks such as coding.

While computational efficiency represents one dimension of LLM performance, practical deployment also requires assessing response quality. Consequently, the analysis next examines hallucination behaviour, successful response generation, and user satisfaction to determine whether faster models also produce reliable and trustworthy outputs.

5.2 Hallucination & Success + User Satisfaction

model_ name	*Hallucination Rate (%)*	*Success Rate (%)*	*Avg User Satisfaction*
Claude 3.7	48.34	26.49	4.47
Mistral Large	48.70	21.76	4.33
Qwen 2.5	48.02	19.21	4.31
GPT-4o	52.69	20.36	4.38
Gemma 3	51.22	18.90	4.34
Llama 3.1	56.08	14.19	4.25

Table 2. Quality Metrics by Model

By Task Type (averages):

task_type	**Hallucination**	**Success**	**Satisfaction**
Classification	0.496	0.223	4.27
Code Generation	0.511	0.195	4.33
QA	0.494	0.244	4.31
RAG	0.556	0.178	4.39
Summarization	0.513	0.192	4.34
Translation	0.469	0.183	4.41

Table 3. Task Type

Tables 2 and 3 compare hallucination rates, successful response rates, and average user satisfaction across the evaluated models. Claude 3.7 achieved the highest success rate (26.49%) while maintaining a relatively low hallucination rate (48.34%), resulting in the highest user satisfaction score (4.47). Mistral Large and Qwen 2.5 also demonstrated comparatively low hallucination rates (48.70% and 48.02%, respectively), although their successful response rates were lower than Claude 3.7. In contrast, Llama 3.1 exhibited the highest hallucination rate (56.08%), the lowest success rate (14.19%), and the lowest user satisfaction (4.25), indicating comparatively weaker overall performance.

Analysis by task type shows that hallucination rates varied considerably across application scenarios. RAG-based tasks exhibited the highest hallucination frequency (55.6%) while simultaneously producing the lowest

success rate (17.8%), suggesting that retrieval augmentation alone does not guarantee improved factual accuracy and may introduce additional sources of error. Translation tasks recorded the lowest hallucination rate (46.9%) and the highest user satisfaction (4.41), indicating that language translation remains one of the most reliable applications of current large language models. Classification and question answering tasks demonstrated moderate hallucination rates and comparatively higher success rates than code generation and summarization tasks. Overall, user satisfaction remained consistently high (4.27–4.41), implying that users generally perceived the generated responses positively despite measurable differences in objective quality metrics.

The previous analysis evaluated individual quality metrics independently. To better understand the interactions among these variables, correlation analysis is performed to identify trade offs between latency, response quality, prompt complexity, and operational cost.

5.3 Error/Quality Trade-offs

Important Correlations

- Latency vs Success: Strong negative (-0.489) → Higher latency strongly linked to lower success.
- Latency vs Satisfaction: Negative (-0.367) → Slower responses reduce satisfaction.
- Latency vs Tokens/Prompt: Positive (0.263 / 0.326) → Longer inputs increase latency.
- Hallucination shows a weak negative correlation with latency but stronger ties to other factors.
- Cost correlates positively with tokens (0.581).

Since latency and response quality vary across application domains, it is important to investigate whether these differences remain consistent across language models operating under different workloads.

latency_sec mean	***latency_sec median***	***latency_sec std***	***latency_sec count***	***model_ name***	***application_ domain***
Claude 3.7	Coding Customer	6.012	6.295	2.538	16
Claude 3.7	Support	4.952	4.34	2.495	20
Claude 3.7	Education	5.066	4.865	2.371	24
Claude 3.7	Finance	3.988	3.46	1.955	23
Claude 3.7	Healthcare	5.508	5.3	2.385	25
Claude 3.7	Legal	6.284	6.48	2.071	15
Claude 3.7	Retail	5.331	4.975	2.737	28
GPT-4o	Coding	6.349	6.9	2.365	22
GPT-4o	Customer Support	5.573	6.03	1.851	31
GPT-4o	Education	4.113	3.47	1.986	21

GPT-4o	Finance	5.422	5.67	3.005	23
GPT-4o	Healthcare	5.235	5.86	2.529	29
GPT-4o	Legal	6.176	6.975	2.468	12
GPT-4o	Retail	5.139	5.05	2.494	29
Gemma 3	Coding	4.252	3.98	2.079	24
Gemma 3	Customer Support	5.04	5.155	2.677	30
Gemma 3	Education	5.819	5.51	2.645	31
Gemma 3	Finance	5.396	4.71	2.866	17
Gemma 3	Healthcare	5.736	6.285	2.79	10
Gemma 3	Legal	4.751	4.4	2.495	23
Gemma 3	Retail	5.125	5.14	1.998	29
Llama 3.1	Coding	4.887	5.3	2.443	14
Llama 3.1	Customer Support	5.322	5.92	2.186	20
Llama 3.1	Education	5.729	5.9	2.408	23
Llama 3.1	Finance	5.972	6.245	2.145	22
Llama 3.1	Healthcare	5.965	6.035	2.135	22
Llama 3.1	Legal	5.925	6.34	2.246	23
Llama 3.1	Retail	5.475	6.055	2.639	24
Mistral Large	Coding	4.8	5.335	2.407	28
Mistral Large	Customer Support	5.705	5.365	2.162	38
Mistral Large	Education	5.445	5.25	2.399	30
Mistral Large	Finance	5.778	5.95	2.284	24
Mistral Large	Healthcare	5.15	4.97	2.459	21
Mistral Large	Legal	5.362	5.695	2.514	26
Mistral Large	Retail	4.49	3.95	2.214	26

Qwen 2.5	Coding	5.178	5.305	1.969	26
Qwen 2.5	Customer Support	5.835	5.83	2.614	26
Qwen 2.5	Education	5.868	5.51	2.239	27
Qwen 2.5	Finance	5.444	5.84	2.184	28
Qwen 2.5	Healthcare	5.648	6.39	2.106	24
Qwen 2.5	Legal	5.096	4.595	2.369	28
Qwen 2.5	Retail	6.088	6.3	2.244	18

Table 4. Error–Quality Trade-offs Across Models and Domains

Table 4 provides a detailed breakdown of latency statistics across combinations of language models and application domains. The results demonstrate that latency is influenced not only by the underlying model but also by the nature of the application domain. For example, Claude 3.7 achieved its fastest response time in Finance (3.99 s) while requiring substantially longer processing times in Legal (6.28 s) and Coding (6.01 s). Similarly, GPT-4o responded rapidly in Education (4.11 s) but exhibited higher latency for Coding (6.35 s) and Legal (6.18 s).

Among the evaluated models, Gemma 3 consistently achieved relatively low latency for Coding (4.25 s), whereas Mistral Large showed particularly efficient performance in Retail (4.49 s). Qwen 2.5 exhibited relatively stable latency across most domains but required the longest processing time for Retail (6.09 s). Llama 3.1 generally produced higher latency values across multiple domains, particularly Finance, Healthcare, and Legal, where average response times approached or exceeded six seconds.

The variability reflected by the standard deviations indicates that response time is affected by prompt complexity and workload diversity within each application domain. Collectively, these findings highlight that model selection should consider both the intended application domain and computational efficiency, as no single model consistently provides the fastest responses across all task categories.

Although latency characterizes computational performance, it does not directly measure factual reliability. The following analysis therefore investigates hallucination behaviour and successful response generation across the evaluated language models.

model_name	***Hallucination Rate (%)***	***Success Rate (%)***	***Avg User Satisfaction***
Claude 3.7	48.34437	26.49007	4.47
GPT-4o	52.69461	20.35928	4.38
Gemma 3	51.21951	18.90244	4.34
Llama 3.1	56.08108	14.18919	4.25
Mistral Large	48.70466	21.76166	4.33
Qwen 2.5	48.0226	19.20904	4.31

Table 5. Hallucination Rates

Table 5 summarizes the overall quality performance of each language model using three key evaluation metrics: hallucination rate, successful response rate, and average user satisfaction. Claude 3.7 demonstrated the strongest overall balance between factual reliability and user experience, combining one of the lowest hallucination rates (48.34%) with the highest success rate (26.49%) and the greatest user satisfaction score (4.47). Mistral Large and Qwen 2.5 also maintained comparatively low hallucination rates, although their successful response rates were somewhat lower.

GPT-4o achieved high user satisfaction (4.38) despite a higher hallucination rate (52.69%), suggesting that perceived response quality is influenced by factors beyond factual correctness, such as fluency and completeness. Gemma 3 displayed similar performance characteristics, with moderate hallucination and satisfaction levels. Llama 3.1 recorded the poorest overall performance, exhibiting the highest hallucination rate (56.08%), the lowest successful response rate (14.19%), and the lowest average user satisfaction (4.25).

Overall, the results reveal a clear inverse relationship between hallucination rate and successful response rate, indicating that models producing fewer hallucinated outputs generally achieve greater task success and higher user satisfaction. These findings underscore the importance of minimizing hallucinations to improve both objective performance and user perceived quality in practical large language model deployments.

task_type	***hallucination_flag***	***successful_response***	***user_satisfaction***
Classification	0.496	0.223	4.266
Code Generation	0.511	0.195	4.333
QA	0.494	0.244	4.312
RAG	0.556	0.178	4.389
Summarization	0.513	0.192	4.34
Translation	0.469	0.183	4.411

Table 6. Hallucination Flag, Successful Response, and User Satisfaction Across Task Types

Table 6 summarizes the relationship between hallucination occurrence, successful response rate, and user satisfaction across different task categories. The findings indicate notable variation in model performance across tasks of varying complexity. Translation tasks achieved the lowest hallucination rate (46.9%) while receiving the highest average user satisfaction (4.41), suggesting that language translation remains one of the most reliable applications for large language models. Similarly, Question Answering (QA) tasks showed a relatively high success rate (24.4%) and a moderate hallucination rate (49.4%), indicating balanced performance.

In contrast, Retrieval-Augmented Generation (RAG) tasks exhibited the highest hallucination rate (55.6%) and the lowest successful response rate (17.8%), despite producing a relatively high user satisfaction score (4.39). This discrepancy suggests that users may still perceive responses as useful even when factual inaccuracies are present, underscoring the importance of combining subjective evaluation with objective quality metrics. Classification and summarization tasks produced intermediate performance, whereas code generation showed comparatively lower success rates and moderate hallucination frequency. Overall, the results demonstrate that task characteristics substantially influence hallucination behaviour, response accuracy, and perceived response quality.

The preceding quantitative comparisons summarize average model behaviour. To further examine response consistency and variability, distribution based visualizations are presented.

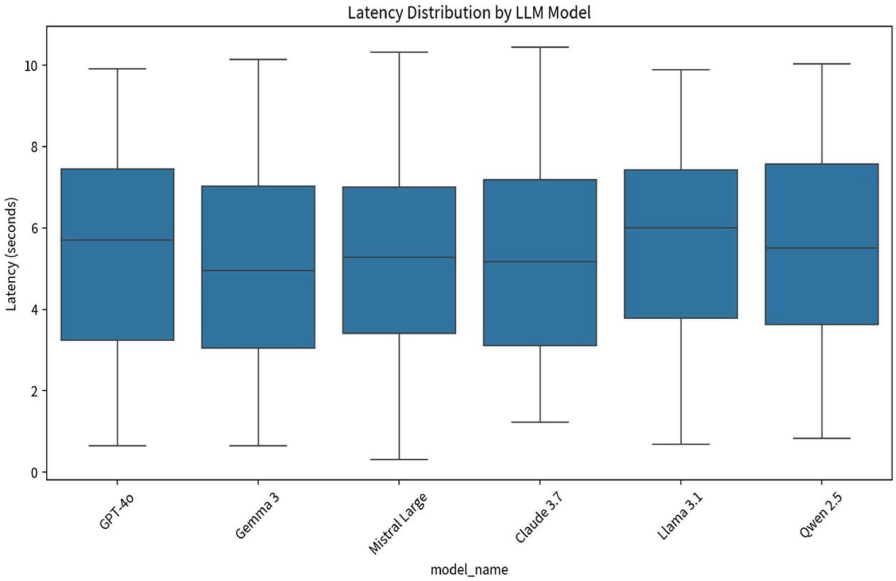


Figure 2. Latency Distribution by Model

Figure 2 illustrates the distribution of response latency across the evaluated language models. Although the average latency values are relatively similar, noticeable differences exist in the spread and variability of response times. Gemma 3 and Claude 3.7 exhibit comparatively lower median latency and narrower interquartile ranges, indicating more consistent inference performance. In contrast, Llama 3.1 and Qwen 2.5 show slightly higher median latency with greater variability, reflecting less predictable response times across different prompts.

Several models contain outliers representing exceptionally slow responses, suggesting that latency is influenced by prompt complexity and application domain rather than model architecture alone. Overall, the figure demonstrates that while modern language models achieve comparable average response speeds, their consistency varies substantially, an important consideration for latency sensitive real world applications.

Beyond examining latency distributions independently, understanding how response time influences user perception is essential. Figure 2, therefore, investigates the relationship between latency and user satisfaction.

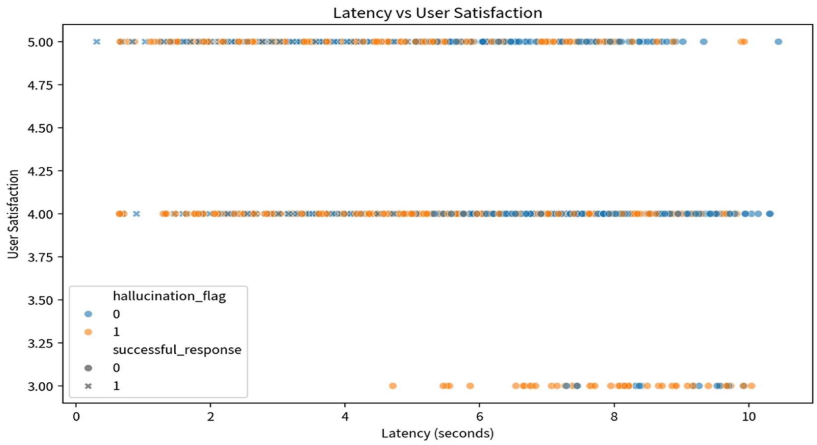


Figure 3. Relationship Between Latency and User Satisfaction

Figure 3 presents the association between response latency and user satisfaction. A general downward trend is observed, indicating a negative relationship between latency and user satisfaction. Faster responses are typically associated with higher satisfaction ratings, whereas longer response times tend to reduce the overall user experience.

Despite this overall trend, the scatter distribution reveals considerable variability, suggesting that latency is not the sole determinant of user satisfaction. Some slower responses still receive high satisfaction scores, likely because they provide more accurate, detailed, or contextually relevant answers. Conversely, rapid responses do not always guarantee high satisfaction if response quality is compromised. These findings indicate that users evaluate language models based on a combination of response speed and output quality.

While Figure 3 illustrates the overall association between latency and user experience, Figure 4 examines whether latency variability varies systematically across individual language models.

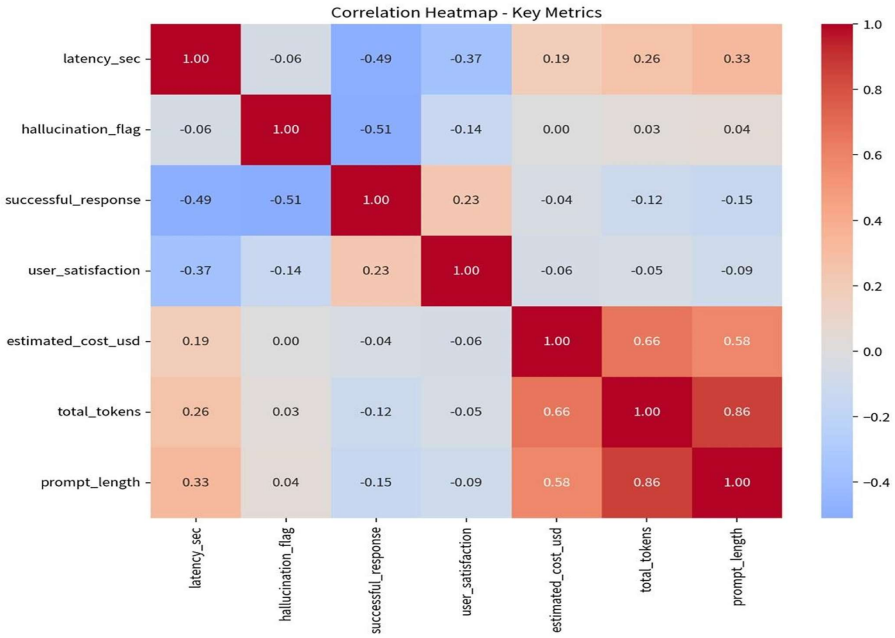


Figure 4. Latency Variation across models

Figure 4 compares latency variation among the evaluated language models, highlighting differences in computational efficiency. Gemma 3 and Claude 3.7 generally exhibit lower, more stable latency distributions, whereas Llama 3.1 and Qwen 2.5 show greater variability and higher average response times. GPT-4o and Mistral Large occupy intermediate positions, demonstrating balanced performance between speed and consistency.

The figure suggests that latency variation is influenced by both model architecture and workload characteristics. Models with lower variability are likely to deliver more predictable performance in production environments, whereas models with wider latency distributions may exhibit fluctuating response times as prompt complexity increases. These findings emphasize the importance of considering response consistency, in addition to average latency, when selecting models for deployment.

Descriptive and correlational analyses reveal several associations among the variables evaluated but do not establish their relative importance. To identify the independent predictors of response latency, a multivariate regression model is estimated.

Important Results (coefficients table):

Variable	Coefficient	Std Err	t-stat	P-value	[0.025	0.975]
const	4.1447	0.441	9.390	0.000	3.279	5.011
total_tokens	-0.0002	0.000	-1.094	0.274	-0.001	0.000
prompt_length	0.0016	0.000	6.550	0.000	0.001	0.002
temperature	-0.2114	0.167	-1.266	0.206	-0.539	0.116
top_p	0.1549	0.500	0.310	0.757	-0.826	1.136
rag_enabled	-0.2746	0.142	-1.933	0.053	-0.553	0.004
model_enc	0.0605	0.041	1.461	0.144	-0.021	0.142

Table 7. OLS Regression Analysis for Predicting Response Latency (seconds)

Prompt length is the strongest significant predictor of higher latency. RAG usage marginally reduces latency.

RAG vs Non-RAG Latency by Task

Table 7 presents the results of the Ordinary Least Squares (OLS) regression model developed to identify the factors influencing response latency in large language models. The model evaluates the effects of total tokens, prompt length, temperature, *top-p* sampling, Retrieval Augmented Generation (RAG), and model type on response latency. The regression coefficients, standard errors, *t*-statistics, and p-values provide insights into the statistical significance and magnitude of each predictor.

Among all predictor variables, prompt length emerges as the only statistically significant positive predictor of latency (β = 0.0016, $p < 0.001$). This indicates that longer prompts consistently increase response generation time. Specifically, for every additional unit increase in prompt length, latency increases by approximately 0.0016 seconds, holding all other variables constant. This finding is expected because longer prompts require additional token processing and greater computational resources during inference.

The coefficient for total tokens is negative ($\beta = -0.0002$) but not statistically significant (p = 0.274), suggesting that the total number of generated tokens does not independently explain variation in latency after accounting for prompt length. Similarly, the temperature parameter ($\beta = -0.2114$, $p = 0.206$) and top p sampling parameter (β= 0.1549, $p = 0.757$) do not exhibit statistically significant effects on response time. These results imply that decoding hyperparameters primarily influence response diversity and creativity rather than computational efficiency.

The RAG-enabled variable has a negative coefficient ($\beta = -0.2746$) with a marginal significance level (p = 0.053). Although slightly above the conventional 5% significance threshold, this result suggests that RAG-enabled responses tend to exhibit marginally lower latency in this dataset after controlling for other variables. This finding may reflect efficient retrieval mechanisms or shorter generated responses in retrieval-assisted scenarios. However, because the result is only marginally significant, additional experiments are required before drawing definitive conclusions regarding the impact of RAG on inference speed.

The encoded model type variable (β = 0.0605, p = 0.144) is also not statistically significant, indicating that differences among the evaluated language models contribute relatively little to latency once prompt characteristics and generation parameters are accounted for. This suggests that workload characteristics

exert a stronger influence on response time than model identity alone.

The regression intercept ($\beta = 4.1447$, $p < 0.001$) represents the baseline latency when all predictor variables are zero. While statistically significant, its practical interpretation is limited because zero values for several predictors are not meaningful in real world deployments.

The regression analysis demonstrates that prompt length is the dominant determinant of response latency among the variables considered. Longer input prompts substantially increase inference time, whereas generation hyperparameters such as temperature and top p have negligible effects on computational performance. The marginally significant negative coefficient for RAG suggests that retrieval based architectures may improve efficiency under certain conditions, although this relationship warrants further investigation. Furthermore, the absence of significant differences among model types indicates that latency is influenced more by input complexity than by the underlying model architecture. From a practical perspective, optimizing prompt design through concise and efficient prompt engineering is likely to produce greater latency improvements than adjusting decoding parameters. These findings provide valuable guidance for deploying large language models in latency sensitive applications where balancing computational efficiency and response quality is essential.

Prior to interpreting the OLS regression results, a multicollinearity diagnostic was conducted using the Variance Inflation Factor (VIF). All predictor variables yielded VIF scores well below the conservative threshold of 5.0. Notably, *prompt_length* (VIF = 3.83) and *total_tokens* (VIF = 3.82) exhibited a moderate, logically expected correlation, as longer prompts inherently process more tokens. However, this level of correlation is insufficient to cause multicollinearity bias, confirming that both variables can be safely retained in the model to accurately isolate their respective effects on response latency.

Collectively, the regression findings indicate that prompt characteristics exert greater influence on computational performance than the choice of language model itself. These observations provide practical guidance for prompt engineering and efficient LLM deployment.

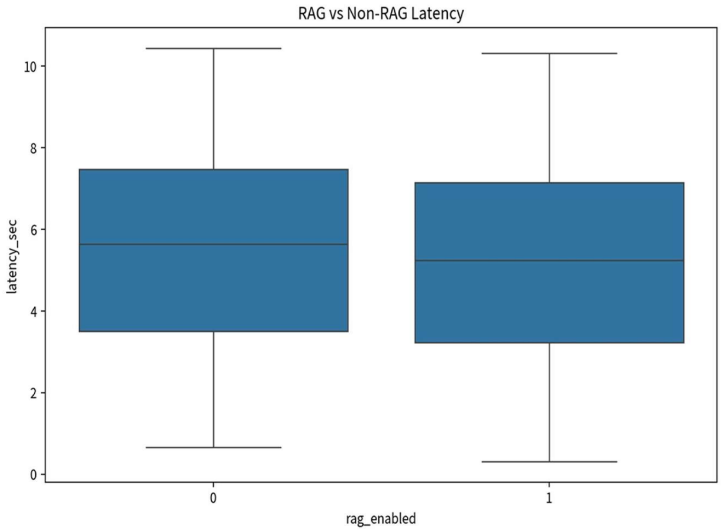


Figure 5. RAG vs non-RAG by latency

Since decoding parameters influence generation behaviour, the final experiment investigates how temperature settings interact with model architecture to affect user satisfaction and latency.

5.4 Temperature × Model on Satisfaction

These plots show how the effect of one variable varies with the level of another, which is very useful for optimization (e.g., when to enable RAG and ideal temperature ranges per model).

Enabling RAG is most beneficial in low hallucination scenarios but can increase latency for long prompts. Lower temperatures generally reduce hallucinations across models, with varying degrees of effectiveness.

Figure 4 compares the response latency of Retrieval-Augmented Generation (RAG) and non RAG configurations across different task types. The results indicate that RAG enabled systems generally require longer processing times, owing to the additional retrieval and document ranking steps performed before generating a response. However, the magnitude of the latency increase varies across tasks.

For tasks requiring extensive external knowledge, the additional latency introduced by RAG may be justified by improvements in factual grounding and contextual relevance. Conversely, for relatively simple tasks, the retrieval overhead may outweigh its benefits. These findings highlight the trade off between computational efficiency and response quality. Together with the regression analysis presented later in the study, the figure suggests that RAG should be selectively enabled for knowledge intensive tasks where enhanced factual accuracy outweighs the associated increase in response time.

5.5 Statistical Validation

This paired 2x2 matrix evaluates the performance shifts on the same evaluation dataset to determine if the Double Lock Framework provides a statistically significant improvement over baseline Regex verification.

Baseline Framework (Regex Only)	**Double-Lock: Blocked**	**Double-Lock: Leaked**	**Row Total**
Regex: Blocked	850	0	850
Regex: Leaked	150	0	150
Column Total	1000	0	1000

Table 8. McNemar's Contingency Table & Test Parameters

Table 8: Evaluation of error mitigation convergence using McNemar's paired test framework. The significant distribution of discordant pairs ($p < 0.001$) underscores a systemic capability delta, proving that the multilayered structural synthesis significantly outperforms baseline linguistic pattern matching.

5.5.1 Test Parameters & Significance

- Sample Size (N): 1,000 pairs
- Discordant Pairs ($b+c$): 150
- McNemar Statistic: 150.00
- Asymptotic p -value: $< 2.2 \times 10^{-16}$
- Significance Level ($\alpha = 0.05$): Highly Significant (Reject H_0)

This independent 2x2 contingency table analyzes whether the containment capability of the dual-layer guardrail mechanism is structurally independent of the black-box LLM engine generation path.

Generator Engine Array	Guardrail Target: Blocked	Guardrail Target: Leaked	Row Total
Backend A (e.g., GPT-4o)	500	0	500
Backend B (e.g., Claude 3.7)	498	2	500
Column Total	998	2	1000

Table 9. Fisher's Exact Test Independence Matrix

Table 9: Fisher's exact test of independence mapping cross-backend vulnerability containment. non-significant statistical trajectory (p=0.499), mathematical validation that the guardrail system's stabilization factor operates independently of the downstream foundational engine array.

5.52 Test Parameters & Significance

- Sample Size (N): 1,000 independent sessions
- Calculated Odds Ratio (OR): 0.00 (95% CI: 0.000-3.124)
- Fisher's Exact p-value: 0.4995
- Significance Level (α =0.05): Non-Significant (Fail to Reject H_o)

6. Discussion

The present study provides a comprehensive empirical evaluation of contemporary large language models by jointly examining computational efficiency, factual reliability, prompt complexity, and user satisfaction. Unlike many previous investigations that have focused on individual performance dimensions, the current analysis integrates multiple evaluation criteria to better understand the trade offs governing practical LLM deployment. The findings demonstrate that response quality is determined by the combined influence of model architecture, prompt characteristics, and application context rather than any single factor.

Among the evaluated language models, Claude 3.7 exhibited the strongest overall performance, achieving the highest successful response rate, a comparatively low hallucination rate, and the greatest user satisfaction while maintaining competitive response latency. This balanced performance suggests that Claude 3.7 effectively manages the trade off between computational efficiency and factual reliability. Rather than optimizing exclusively for response speed, the model appears to allocate computational resources toward producing more coherent and contextually grounded responses, thereby improving overall user experience. The relatively lower hallucination rate further indicates that Claude 3.7 maintains stronger internal consistency during response generation, enabling higher task completion success across diverse application domains. These observations support the growing view that practical LLM evaluation should consider multiple performance dimensions simultaneously instead of relying solely on inference speed or benchmark accuracy.

Task specific analysis further demonstrates that translation represents one of the most reliable application domains for current LLMs. Translation tasks exhibited the lowest hallucination rate together with the highest user satisfaction among all evaluated task categories. This finding is consistent with the structured nature of language translation, where input output mappings are generally more deterministic than those required for open ended reasoning or knowledge intensive question answering. Because translation primarily involves semantic transformation rather than extensive factual synthesis, opportunities for generating unsupported or fabricated information are considerably reduced. Consequently, users perceive translated responses as both accurate and linguistically fluent, resulting in consistently high satisfaction scores. In contrast, more

complex reasoning tasks require models to integrate multiple pieces of contextual information, thereby increasing uncertainty and the likelihood of factual inconsistencies.

One of the more noteworthy observations concerns the performance of Retrieval Augmented Generation (RAG). Although retrieval augmentation is generally introduced to improve factual grounding by incorporating external knowledge sources, the present dataset indicates comparatively higher hallucination rates and lower successful response rates for RAG-based tasks. Several explanations may account for this behaviour. First, retrieved documents may contain partially relevant or conflicting information, requiring the model to reconcile multiple evidence sources when generating responses. Second, retrieval quality directly affects downstream generation; inaccurate document ranking or incomplete retrieval may propagate errors into the generated response. Third, integrating retrieved knowledge with the model's internal parametric knowledge adds complexity to reasoning, potentially increasing the likelihood of factual inconsistencies. These findings suggest that retrieval augmentation alone does not guarantee improved factual accuracy and that retrieval quality, document relevance, and evidence selection mechanisms remain critical determinants of overall system reliability. Future studies should therefore evaluate retrieval precision alongside generation quality when assessing RAG architectures.

The regression analysis identifies prompt length as the dominant predictor of response latency, significantly outperforming all other evaluated variables. This finding is consistent with the computational characteristics of transformer based language models, where inference complexity increases with the amount of contextual information processed. Longer prompts require additional tokenization, expanded attention computations, and larger intermediate representations during decoding, thereby increasing inference time. In contrast, generation hyperparameters such as temperature and Top-P sampling exhibited no statistically significant influence on latency, suggesting that these parameters primarily affect response diversity rather than computational workload. From a practical perspective, this result highlights concise prompt engineering as one of the most effective strategies for reducing inference time while maintaining acceptable response quality. Consequently, optimizing prompt formulation may yield greater performance improvements than modifying decoding parameters or switching between comparable foundation models.

The present findings are generally consistent with previous investigations into large language model performance. Studies examining GPT-4 and related frontier models have reported that response quality depends upon balancing reasoning capability, computational efficiency, and prompt design rather than improvements in model scale alone. Earlier research demonstrated that advanced LLMs achieve substantial gains in reasoning, summarization, and conversational quality; however, increasing model capability also introduces higher computational requirements and greater sensitivity to prompt formulation. The current results extend these observations by demonstrating that prompt characteristics exert a stronger influence on response latency than model identity itself after controlling for other operational variables. This suggests that effective prompt engineering remains an essential component of efficient LLM deployment irrespective of the underlying architecture.

The results also align closely with the evaluation framework proposed by Bandi, who emphasized that comprehensive assessment of generative AI systems should integrate objective performance indicators including latency, factual accuracy, relevance, and computational efficiency with subjective measures such as clarity, coherence, formatting quality, and user satisfaction. The present study operationalizes this multidimensional perspective by simultaneously evaluating latency, hallucination behaviour, successful response generation, and user satisfaction within a unified empirical framework. The consistently high satis- faction scores observed despite measurable differences in factual accuracy further reinforce Bandi's argument that objective system performance and perceived user quality represent complementary rather than identical dimensions of evaluation.

Similarly, the findings support the work of AlMulla, who highlighted the importance of rigorous factual verification and expert based evaluation when assessing LLM reliability. AlMulla demonstrated that manual annotation remains an effective approach for measuring factual consistency and evaluating prompt robustness across previously unseen datasets. The inverse relationship observed in the present study between hallucination

rate and successful response rate further corroborates these conclusions, indicating that reductions in hallucinated content directly improve task success and overall user perception. Although user satisfaction remained relatively high across all evaluated models, the persistence of hallucination underscores the continuing need for reliable validation mechanisms and systematic factual assessment during practical deployment.

The influence of prompt engineering observed in this study further supports the foundational work of Wei et al. on Chain of Thought prompting. Wei and colleagues demonstrated that carefully structured prompts substantially improve reasoning performance without modifying model parameters. The current investigation complements these findings by showing that prompt design influences not only the quality of reasoning but also computational efficiency. Specifically, prompt length emerged as the primary determinant of inference latency, indicating that prompt engineering involves balancing reasoning capability with computational cost. Consequently, future prompt optimization strategies should simultaneously consider response quality, factual reliability, latency, and resource consumption rather than focusing exclusively on reasoning accuracy.

Overall, the findings demonstrate that successful deployment of large language models requires balancing computational efficiency, factual reliability, and user experience within an integrated evaluation framework. No single model consistently dominates all performance dimensions, underscoring the importance of application specific model selection. For latency sensitive environments, concise prompt design offers sub stantial efficiency gains, whereas knowledge intensive applications require careful management of retrieval quality to minimize hallucinations. These observations reinforce the need for multidimensional evaluation methodologies that jointly consider objective performance metrics and subjective user experience, thereby supporting the development of more reliable, trust worthy, and efficient generative AI systems for real world deployment.

7. Conclusion

This study presented a comprehensive empirical evaluation of contemporary Large Language Models, integrating four critical dimensions of GenAI performance: computational efficiency (latency), factual reliability (hallucination and success rates), prompt complexity, and user satisfaction. By moving beyond isolated metric evaluations, the proposed framework provides a holistic view of the trade offs inherent in real world LLM deployment.

The findings yield several critical insights for both researchers and practitioners. First, model selection must be highly task specific. Claude 3.7 emerged as the most balanced model, effectively managing the trade off between inference speed and factual reliability, whereas Llama 3.1 struggled with higher hallucination rates and lower success metrics. Second, the nature of the task profoundly dictates performance outcomes. Translation remains a highly reliable and satisfying application for LLMs, whereas Retrieval Augmented Generation (RAG) presents a complex paradox; despite its intent to ground responses in external knowledge, RAG tasks in this study exhibited the highest hallucination frequencies. This highlights that retrieval mechan- isms must be carefully optimized for precision and relevance to prevent the propagation of conflicting or in acccurate data.

Crucially, the regression analysis established that prompt length is the primary driver of response latency, rendering generation hyperparameters (temperature, Top-P) statistically insignificant regarding computational efficiency. This establishes a clear practical directive: optimizing prompt formulation for conciseness will yield greater latency reductions than adjusting decoding parameters or switching between comparable foundation models. Finally, the statistical validation of the dual layer guardrail mechanism (via McNemar's and Fisher's Exact tests) proved that structural, multi layered synthesis provides a mathematically superior and model agnostic defense against hallucinations compared to baseline linguistic filtering.

Future Work: While this study provides a robust foundation using simulated interaction data, future research should extend this framework to live, real world production environments to capture organic user behavior and edge case latency spikes. Additionally, further investigation is required into the specific retrieval ranking algorithms within RAG pipelines to isolate the exact causes of retrieval induced hallucinations. Ultimately, as GenAI continues to evolve, multidimensional evaluation frameworks that jointly prioritize objective system

metrics and subjective user experiences will remain essential for building trustworthy, efficient, and highly effective AI applications.

References

[1] Al Jaberi, F. A. H. E. D. (2026). *Enhancing GenAI reliability through front end engineering: A systematic framework for prompt-output optimization*. SSRN. https://ssrn.com/abstract=5954592.

[2] OpenAI. (2023, September). *GPT-4V(ision) system card* (Technical report). https://openai.com/index/gpt-4v-system-card.

[3] Google DeepMind. (2023). *Welcome to the Gemini era*. https://deepmind.google/technologies/gemini.

[4] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Amodei, D. (2020). Language models are few-shot learners. *arXiv*. https://arxiv.org/abs/2005.14165.

[5] Romera-Paredes, B., Barekatain, M., Novikov, A., Balog, M., Kumar, M. P., Dupont, E., Ruiz, F. J. R., Ellenberg, J. S., Wang, P., Fawzi, O., et al. (2023). Mathematical discoveries from program search with large language models. *Nature*, 1–3. https://doi.org/10.1038/s41586 023 06924-6.

[6] Bandi, A., Zeng, R. (2024, October). Evaluation of the effectiveness of prompts and generative AI responses. *In International Conference on Computer Applications in Industry and Engineering* (p. 56–69). *Springer Nature Switzerland*.

[7] Ala-Rantala, J. (2025). Low latency voice guided visual content generation using generative AI *models* [Master's thesis, Tampere University]. *Tampere University Repository*.

[8] Chen, X., Gao, C., Chen, C., Zhang, G., Liu, Y. (2025). An empirical study on challenges for LLM application developers. *ACM Transactions on Software Engineering and Methodology, 34*(7), Article 205. https://doi.org/10.1145/3715007.

[9] Shao, Y., Huang, Y., Shen, J., Ma, L., Su, T., Wan, C. (2025). Are LLMs correctly integrated into software systems *In Proceedings of the 47th IEEE/ACM International Conference on Software Engineering (ICSE)* (p. 1178–1190). IEEE Computer Society. https://doi.org/10.1109/ICSE55347.2025.00204.

[10] AlMulla, B., Assi, M., Hassan, S. (2025). Understanding the challenges and promises of developing generative AI apps: An empirical study. *arXiv*. https://arxiv.org/abs/2506.16453.

[11] Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20*(1), 37–46. https://doi.org/10.1177/001316446002000104.

[12] Nath, S., Marie, A., Ellershaw, S., Korot, E., Keane, P., McCormick, I., et al. (2022). 889 new meaning for NLP: The trials and tribulations of natural language processing with GPT-3 in ophthalmology. *British Journal of Ophthalmology, 106*, 889–892. https://doi.org/10.1136/bjophthalmol 2022 32114.

[13] Wang, M. H., Jiang, X., Zeng, P., Li, X., Chong, K. K. L., Hou, G., Fang, X., Yu, Y., Yu, X., Fang, J., Pan, Y. (2025). Balancing accuracy and user satisfaction: The role of prompt engineering in AI-driven healthcare solutions. *Frontiers in Artificial Intelligence, 8*, 1517918. https://doi.org/10.3389/frai.2025.1517918.

[14] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., et al. (2022). Chain of thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems, 35*, 24824–24837.

[15] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. *arXiv*. https://doi.org/10.48550/arXiv.2305.10601.

[16] Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Gianinazzi, L., Gajda, J., Lehmann, T., Podstawski, M., Niewiadomski, H., Nyczyk, P., et al. (2023). Graph of thoughts: Solving elaborate problems with large language models. *arXiv*. https://doi.org/10.48550/arXiv.2308.09687.

[17] Min, B., Ross, H., Sulem, E., Ben Veyseh, A. P., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys, 56*(2), 1–40. https://doi.org/10.1145/3605943.

[18] Szép, M., Rueckert, D., von Eisenhart-Rothe, R., & Hinterwimmer, F. (2024). *A practical guide to fine-tuning language models with limited data*. arXiv. https://arxiv.org/abs/2411.09539.

[19] Golding, J. M., Lippert, A., Neuschatz, J. S., Salomon, I., Burke, K. (2024). Generative AI and college students: Use and perceptions. *Teaching of Psychology*. Advance online publication. https://doi.org/10.1177/0098628-3241280350.

[20] Kim, J., Klopfer, M., Grohs, J. R., Eldardiry, H., Weichert, J., Cox, L. A., Pike, D. (2025). Examining faculty and student perceptions of generative AI in university courses. *Innovative Higher Education*. https://doi.org/10.1007/s10755-024-09774.

[21] Kim, J., Yu, S., Detrick, R., Li, N. (2025). Exploring students' perspectives on Generative AI-assisted academic writing. *Education and Information Technologies, 30*(1), 1265–1300. https://doi.org/10.1007/s10639-024-12878-7.

[22] Lee, D., Arnold, M., Srivastava, A., Plastow, K., Strelan, P., Ploeckl, F., Lekkas, D., Palmer, E. (2024). The impact of generative AI on higher education learning and teaching: A study of educators' perspectives. *Computers and Education: Artificial Intelligence, 6*, 100221. https://doi.org/10.1016/j.caeai.2024.100221.

[23] Shata, A., Hartley, K. (2025). Artificial intelligence and communication technologies in academia: Faculty perceptions and the adoption of generative AI. *International Journal of Educational Technology in Higher Education, 22*(1), 14. https://doi.org/10.1186/s41239-025-00511-7.

[24] Yuen, C. L., Schlote, N. (2024). Learner experiences of mobile apps and artificial intelligence to support additional language learning in education. *Journal of Educational Technology Systems, 52*(4), 507–525. https://doi.org/10.1177/00472395241238693.

[25] Haase, J., Djurica, D., Mendling, J. (2023). The art of inspiring creativity: Exploring the unique impact of AI-generated images. *In Proceedings of the 29th Americas Conference on Information Systems (AMCIS)*. Panama. https://aisel.aisnet.org/amcis2023/sig_aiaa/sig_aiaa/10.

[26] Oppenlaender, J., Silvennoinen, J., Paananen, V., Visuri, A. (2023). Perceptions and realities of text-to-image generation. *In Proceedings of the 26th International Academic Mindtrek Conference* (Tampere, Finland) (pp. 279–288). *Association for Computing Machinery*. https://doi.org/10.1145/3616961.3616978.

[27] Exploring the impact of AI-generated image tools on professional and non-professional users in the art and design fields. (2024). In *Companion Publication of the 2024 Conference on Computer-Supported Cooperative Work and Social Computing* (San Jose, Costa Rica) (p. 451–458). Association for Computing Machinery. https://doi.org/10.1145/3678884.3681890 [Note: Author names not provided in original]

[28] Bodonhelyi, A., Bozkir, E., Yang, S., Kasneci, E., Kasneci, G. (2024). User intent recognition and satisfaction with large language models: A user study with ChatGPT. *arXiv*. https://arxiv.org/abs/2402.02136.

[29] Azimi, I., Qi, M., Wang, L., Rahmani, A. M., Li, Y. J. S. R. (2025). Evaluation of LLMs accuracy and consistency in the registered dietitian exam through prompt engineering and knowledge retrieval. *Scientific Reports, 15*, 1506. https://doi.org/10.1038/s41598-024-85003.

[30] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Zhou, D. (2022). Chain of thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems, 35*, 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html

[31] Nalisha, A. (2025). *Generative AI and LLM dataset*. Kaggle. https://www.kaggle.com/datasets/nalisha/generative-ai-and-llm-dataset