Evaluating Prompt Engineering Techniques for LLM-Based Test Case Generation in Automotive ADAS Development

Anastasios Zosimidis[1] [2] and Stefan-Alexander Schneider[1]

[1] Hochschule Kempten – University of Applied Sciences, Kempten, Germany

[2] BMW Group, Munich, Germany

anastasios.zosimidis@bmw.de, stefan-alexander.schneider@hs-kempten.de

Abstract: The growing complexity of Advanced Driver Assistance Systems (ADAS) demands efficient methods for generating test cases from natural language requirements. Systematic surveys of prompt engineering [12, 13] document a rapidly growing catalogue of techniques. This paper evaluates eight of them, applied to three LLMs (GPT-4o, Claude 3.5 Sonnet, Claude 3 Haiku), for automated test case generation from industrial ADAS requirements, and reports the reliability of its own evaluation panel alongside the comparative results. Using Latin Hypercube Sampling across five dimensions (prompt technique, requirement, model, temperature, and top_p), we generate 50 experimental configurations and evaluate them in a fully crossed design: four automotive domain specialists each rated all 50 test cases on five quality criteria, yielding 1,000 individual ratings with no missing values. All 50 configurations returned a complete response without a recorded execution failure. Technique means span a narrow band of 0.30 points (3.39 to 3.69) around an overall mean of 3.60, with Few-Shot obtaining the highest descriptive mean, and no comparison across techniques, models, or sampling parameters reached statistical significance. The ratings carry little resolvable signal at the level on which these comparisons operate: the between-test-case variance component is not distinguishable from the residual test-case-by-evaluator variation ($F = 0.98$, $p = 0.52$), and the observed spread between test cases ($SD = 0.22$) is fully accounted for by measurement noise alone (0.23). The evaluators additionally differed systematically in how leniently they used the scale, accounting for an estimated 71.8% of the variance-component total. The evaluators did not agree on which test cases were better than others ($ICC(2,4) = -0.01$, effectively zero agreement). We conclude that this study did not detect differences between prompt techniques, models, or sampling parameters under the evaluation design used, and that the uncalibrated four-rater panel used here could not resolve differences of the size observed. The study was not designed as an equivalence test, so the absence of detected differences should not be read as evidence that they do not exist.

Keywords: Prompt Engineering, Large Language Models, Test Case Generation, ADAS, Automotive Software Testing, Latin Hypercube Sampling, Expert Evaluation, Inter-Rater Reliability

1. INTRODUCTION

The development of Advanced Driver Assistance Systems (ADAS) and Automated Driving (AD) features in the automotive industry involves rigorous verification and validation processes. A critical step in this process is the generation of test cases from software requirements, which traditionally requires significant manual effort by domain experts. As requirements grow in volume and complexity, as observed in current automotive service pack development cycles, the need for automated and semi-automated test case generation becomes increasingly pressing.

Large Language Models (LLMs) have demonstrated remarkable capabilities in understanding and generating structured text from natural language specifications [19]. However, the quality of LLM outputs is highly dependent on the design of input prompts, a discipline known as prompt engineering [2, 12]. While prompt engineering has been studied extensively for general NLP tasks [12, 13], its systematic application to domain-specific software testing tasks in the automotive industry remains largely unexplored. To the best of our knowledge, no prior work provides a controlled, multi-dimensional comparison of prompt techniques, LLM models, and sampling parameters for automotive ADAS requirements with expert-based evaluation.

We adopt a prompt engineering approach, treating the LLM as an unmodified black box, rather than fine-tuning or Retrieval-Augmented Generation (RAG), as this preserves technique portability across models and avoids the need for domain-specific training data.

This paper addresses the following research questions:

RQ1: Which prompt engineering techniques yield the highest-quality test cases for automotive ADAS requirements?

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

RQ2: How do different LLM models (GPT-4o, Claude 3.5 Sonnet, Claude 3 Haiku) compare in test case generation quality?

RQ3: What is the effect of sampling parameters (temperature, top_p) on the quality and consistency of generated test cases?

In addition to these three pre-defined questions, the fully crossed rating design allowed a post-hoc assessment of the evaluation instrument itself. We report this analysis as exploratory in Section 4.6; it was not part of the original study design, but it proved essential for interpreting the results of RQ1 to RQ3.

To answer these questions, we design a comprehensive experiment using Latin Hypercube Sampling (LHS) to efficiently cover the multi-dimensional parameter space. We develop two purpose-built tools: the Prompt Engineering Tool (PET), a GUI application for authoring prompt templates with placeholder-based requirement injection, and PEBatch2, a batch execution engine that automates LLM API interactions and response collection at scale.

Our results run counter to the expectation implicit in much of this literature. Across all eight techniques and all three models, no group comparison reached statistical significance, and the ratings could not be distinguished by technique, model, or sampling parameter. The post-hoc analysis locates the reason. The between-test-case variance component, which aggregates every source that differs across test cases including technique and model, is not distinguishable from the residual test-case-by-evaluator variation, so the ratings carry little resolvable signal at the level on which the group comparisons operate. The evaluators additionally differed systematically in scale usage. This shifts the practical question from which prompt technique to choose towards how the quality of generated test cases can be measured reliably in the first place.

The main contributions of this paper are: (1) A systematic comparison of eight prompt engineering techniques for automotive test case generation, evaluated in a fully crossed design by four domain experts across five quality criteria, yielding 1,000 individual ratings with no missing values. (2) An LHS-based experiment design methodology that enables efficient exploration of the combined prompt technique, model, and parameter space with 50 configurations. (3) Two purpose-built tools, PET for prompt template authoring and PEBatch2 for batch LLM execution. (4) An exploratory post-hoc analysis of the expert-based evaluation procedure used here, including a variance decomposition of the rating matrix.

The remainder of this paper is organized as follows. Section 2 reviews related work on prompt engineering and LLMs in software testing. Section 3 describes our methodology, tooling, and experiment design. Section 4 presents experimental results, including the analysis of inter-rater reliability. Section 5 discusses findings, practical implications, and limitations. Section 6 concludes the paper and outlines future work.

2. RELATED WORK

2.1 Prompt Engineering Techniques

Prompt engineering has emerged as a critical discipline for maximizing LLM capabilities without modifying model parameters [13]. Sahoo et al. [12] provide a systematic survey categorizing prompting techniques by application area. Key techniques relevant to our work include:

Zero-Shot Prompting relies solely on the task description without examples. It serves as the baseline against which Brown et al. [1] introduce the few-shot setting.

Few-Shot Prompting includes exemplar input-output pairs to guide the model [1].

Zero-Shot Chain-of-Thought (Zero-Shot-CoT) prompting elicits step-by-step reasoning through an instruction such as "let us think step by step", without providing worked examples [5]. It is the example-free variant of Chain-of-Thought prompting, which originally relied on exemplars containing explicit reasoning chains [20].

Self-Consistency improves upon Chain-of-Thought by sampling multiple reasoning paths and selecting the most consistent answer [18].

Tree of Thoughts (ToT) enables deliberate problem-solving through branching exploration of reasoning paths [22, 10].

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

Thread of Thought (ThoT) systematically segments chaotic contexts for improved comprehension [23].

Role-Play Prompting assigns the LLM a specific expert persona to enhance domain-specific reasoning [6, 14, 21].

Emotional Prompting incorporates emotional stimuli into prompts to improve LLM performance [9].

2.2 LLMs in Software Testing and Requirements Engineering

The application of LLMs to software engineering tasks has gained significant attention in recent years. Wei [19] demonstrates the generation of code directly from natural language requirements, and the systematic surveys by Sahoo et al. [12] and Schulhoff et al. [13] document the rapid growth of prompting research across application areas.

However, the studies we are aware of evaluate a single prompting strategy or a single model, and typically use open-source benchmarks rather than industrial requirements. To the best of our knowledge, a systematic comparison of multiple prompt techniques for automotive domain test case generation, with industrial requirements and expert evaluation, has not been conducted.

2.3 Latin Hypercube Sampling

Latin Hypercube Sampling (LHS) is a statistical sampling method that ensures efficient coverage of a multi-dimensional parameter space [11]. Unlike full factorial designs, LHS provides near-optimal space-filling properties while keeping the number of experimental runs manageable. We adopt LHS to simultaneously explore prompt techniques, LLM models, and sampling parameters.

3. PROPOSED METHOD

3.1 Overview

Our methodology comprises five phases: (1) prompt template design for eight engineering techniques, (2) experiment design using LHS, (3) automated prompt execution via custom tooling, (4) expert evaluation of generated test cases, and (5) statistical analysis of the resulting ratings.

3.2 Prompt Template Design

We design prompt templates for eight techniques, each structured around a common test case output format with three categories: Precondition (PRE), Action (ACTION), and Postcondition (POST) steps. This three-part structure aligns with the test management tool used by the participating automotive OEM, enabling direct integration of generated test cases into existing workflows. Each template uses five placeholders that are populated with requirement data: Summary, Description, supplementary system information, quality-related metadata, and traceability links.

Table 1 summarizes the distinguishing characteristics of each technique.

Technique	Key Characteristic
Zero-Shot	Direct task description; no examples or reasoning guidance
Zero-Shot-CoT	Requests step-by-step Chain-of-Thought reasoning without providing worked examples before presenting the final test case
Few-Shot	Includes two worked domain-specific test case examples from real ADAS requirements
Self-Consistency	Zero-Shot-CoT template executed 5×; responses consolidated via a 6th LLM call
ToT	Simulates three automotive domain experts who independently create, discuss, and consolidate test cases
ThoT	Instructs the LLM to walk through context in "manageable parts step by step"
Role-Play	Assigns the persona of "a Test Engineer with extensive experience in automotive ADAS testing"
Emotional	Adds urgency cues ("My job could be on the line") to boost performance

Table 1. Prompt engineering techniques and their distinguishing characteristics.

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

3.3 Tooling

Prompt Engineering Tool (PET): A Tkinter-based GUI application that allows engineers to: (a) load automotive requirements from Excel files, validated against six mandatory columns covering the requirement summary, its description, two sources of supplementary system information, the requirement identifier, and traceability links, with incomplete rows excluded before prompt generation, (b) author and manage prompt templates with placeholders via an integrated text editor, and (c) export filled prompts to Excel for batch processing, with automatic chunking for large requirement sets (100 requirements per file). Prompt templates persisted as a JSON dictionary in methods.json, keyed by technique name.

PEBatch2 — Batch LLM Execution Engine: A Python application that: (a) generates the experiment design matrix via LHS (testcase_sampling.py), (b) reads sampled configurations and iterates over each experimental run, (c) sends prompts to the LLM API via an AWS-authenticated REST interface with OAuth2 token management, (d) implements special handling for Self-Consistency: five independent queries followed by a consolidation query, (e) collects responses including token counts, finish reasons, and timing metadata, and (f) exports all results to timestamped Excel files.

3.4 LLM Models Under Study

The selection of LLMs was constrained by the enterprise usage policies of the participating automotive OEM, which restrict access to models from OpenAI and Anthropic, provisioned via an internal API gateway running on AWS and Azure infrastructure with Google Apigee-X. From the set of available models, we selected three that span both providers and, according to the vendors' own tier designations, different capability levels, so that the comparison is not confined to a single provider or a single tier. Table 2 summarizes the models.

Model	Provider	Context Window	API Identifier
GPT-4o	OpenAI	128k tokens	gpt-4o
Claude 3.5 Sonnet	Anthropic	200k tokens	anthropic.claude-3-5-sonnet-20240620-v1:0
Claude 3 Haiku	Anthropic	200k tokens	anthropic.claude-3-haiku-20240307-v1:0

Table 2. LLM models evaluated in this study.

GPT-4o (OpenAI, released May 2024), Claude 3.5 Sonnet (Anthropic, released June 2024), and Claude 3 Haiku (Anthropic, released March 2024) were accessed via the API identifiers listed in Table 2. Architecture details and parameter counts are not officially disclosed for these models and are therefore not characterized here. Note that the two Anthropic identifiers are version-pinned, whereas gpt-4o is an alias that resolves to whichever snapshot was current at execution time; the exact GPT-4o version used is therefore not recoverable.

All models were accessed through the same enterprise API with identical system prompts and MAX_COMPLETION_TOKENS = 4096. All 50 configurations were executed on 7 March 2025 within a single session, generating 85,464 tokens in total. Cost and latency were not measured in this study.

3.5 Experiment Design via Latin Hypercube Sampling

We use LHS to sample 50 experimental configurations across five dimensions. The design was generated with pyDOE, which produces stratified samples in the unit interval for each dimension. Continuous dimensions were scaled directly to their target range; discrete dimensions were obtained by multiplying the unit-interval draw by the number of levels and truncating to an integer, with clamping at the upper bound. Because 50 samples do not divide evenly into eight techniques, this yields slightly unequal group sizes of five to seven configurations per technique, and 16 to 17 per model. The requirement dimension is nested within the technique dimension: each of the eight techniques has its own pool of 53 pre-filled prompts covering the same set of requirements, and the requirement draw indexes into the pool of the technique selected for that configuration.

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

Dimension	Type	Values / Range
Prompt Technique	Discrete (8)	Zero-Shot, Zero-Shot-CoT, Few-Shot, Self-Consistency, ToT, ThoT, Role-Play, Emotional
Requirement	Discrete (53)	Per-technique pool of filled prompts, same requirement set across techniques
LLM Model	Discrete (3)	GPT-4o, Claude 3.5 Sonnet, Claude 3 Haiku
Temperature	Continuous	[0.0, 1.0]
top_p	Continuous	[0.0, 1.0]

Table 3. LHS parameter space (5 dimensions, 50 samples).

For Self-Consistency samples, each run requires 5+1 = 6 API calls (five independent generations plus one consolidation), while every other technique requires a single call. With six Self-Consistency runs among the 50 configurations, this amounts to 80 API calls in total. Drawing 50 configurations from a pool of 53 requirements necessarily produces repetitions: the executed design covers 43 distinct requirements, of which 36 were sampled once and seven twice. Because the same requirement is present in every technique pool, the repeated draws occur under different technique, model, or parameter settings, so no configuration is an exact repetition of another.

The executed configurations, comprising prompt technique, model, temperature, and top_p for every generated test case, were preserved together with the model responses, and all analyses in this paper are reproduced from these preserved executed configurations. Because the original sampling script did not fix a random seed, re-running it produces a different design; reproducibility therefore refers to the analysis of the preserved executed configurations rather than to regenerating an identical LHS design.

The sample size of 50 was chosen for space filling coverage of the five dimensional parameter space rather than for group wise hypothesis testing. This trade off is inherent to the design: an LHS that covers two continuous parameters well distributes only five to seven configurations per prompt technique, which is sufficient for descriptive comparison but not for detecting moderate group differences. Section 5.5 quantifies the resulting limitation.

The system prompt used for all experiments is: "You are an AI assistant tasked with generating detailed technical test cases to verify the requirements for ADAS/AD features in a current automotive service pack release. Use your expertise in testing and automotive domain knowledge to provide comprehensive and well-structured test case documentation."

3.6 Expert Evaluation

Generated test cases are evaluated by four automotive domain experts (from six invited, yielding a 67% response rate) through structured surveys implemented as VBA-enhanced Excel workbooks in German, the working language of the participating automotive OEM. The same set of 50 unique generated test cases was independently assessed by all four evaluators. Evaluators received only the generated test case and its requirement identifier; the prompt technique, LLM model, and sampling parameters were not disclosed, so all ratings were collected blind to the experimental condition. Each evaluator rated all 50 test cases on five quality criteria using a Likert scale from 1 (lowest) to 5 (highest). This fully-crossed design yields 200 evaluator–test-case pairs and 1,000 individual criterion ratings, with no missing values. The five criteria, in the order presented in the survey instrument, are:

1. Repeatability & Consistency: Whether the test case yields repeatable, consistent results under identical conditions.

2. Clear Preconditions: Whether modes, settings, and tokens are unambiguously and completely defined.

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

3. Domain Knowledge Required: How much domain expertise executing the test case demands. The item is worded so that 5 indicates that very little expert knowledge is needed and 1 that extensive expert knowledge is required, so that higher values denote the more desirable outcome, consistent with the other four criteria.

4. Creativity: Whether the test case is innovative and suitable for uncovering edge cases.

5. Acceptance Criteria: Whether expected results and outputs are clearly identifiable, enabling effective validation.

3.7 Statistical Analysis

The overall score of a test case for a given evaluator is defined as the unweighted mean of that evaluator's five criterion ratings; all five criteria are coded so that higher values denote the more desirable outcome. Because each criterion rating is an integer, an evaluator's overall score is a multiple of 0.2 and a per-test-case mean a multiple of 0.05. Floating-point representation perturbs these values by approximately 10^{-16}, which causes rank-based procedures to break genuine ties. All per-test-case means were therefore rounded to this exact grid before any rank-based statistic was computed. Test cases are the independent experimental units and the four evaluators are repeated raters; group comparisons therefore use the 50 per-test-case mean scores (the mean of the four evaluators' overall scores) as the unit of analysis. Because the per-test-case means derive from ordinal ratings and the groups are small and unbalanced, group differences (prompt technique, model, temperature group, top_p group) were assessed with the non-parametric Kruskal–Wallis test [8] and the effect size is reported as epsilon-squared, $\varepsilon^2 = H/(N-1)$ with $N = 50$, following Tomczak and Tomczak [16]. Holm-corrected Dunn post-hoc comparisons were planned only if the corresponding omnibus test was significant.

The sample size required to detect the observed effect was estimated in two ways. Analytically, the Kruskal–Wallis statistic was approximated by a non-central chi-square distribution with $df = k - 1$ and non-centrality $\lambda = \varepsilon^2(N - 1)$, solved for the N at which the power to exceed the critical value $\chi^2(0.95, df)$ reaches 0.80. As a check, the same quantity was obtained by simulation: datasets were resampled with replacement from the observed per-technique distributions at the observed group proportions, and the proportion of significant omnibus tests was recorded. The two approaches agree closely: at the analytically derived sample size of 109, the simulated power is 0.83. The simulation uses 20,000 replicates with a largest-remainder allocation of the group sizes and its own independently seeded generator, so the reported value does not depend on the order of preceding computations. Both approaches assume the observed effect size is the true one, which is optimistic for an estimate obtained from 50 observations.

Inter-rater reliability was assessed on the 50×4 matrix of overall scores following the conventions of Shrout and Fleiss [15], reporting a two-way random-effects, absolute-agreement, average-measures intraclass correlation, ICC(2,4), together with Kendall's coefficient of concordance (W) [4] and the mean pairwise Spearman correlation; systematic differences in rating level between evaluators were tested with the Friedman test [3]. To quantify the relative contribution of evaluators and test cases to the observed rating variation, variance components were estimated from a two-way ANOVA without replication on the same matrix, yielding separate estimates for the evaluator component, the between-test-case component, and the residual test-case-by-evaluator component. Bootstrap confidence intervals used 10,000 resamples with a fixed seed (20260815).

The intraclass correlation expresses the proportion of the total rating variance that is attributable to differences between test cases rather than to the evaluators. In the notation of Shrout and Fleiss [15], the first index denotes the model and the second the measurement form: model 2 applies because all four evaluators rated the same test cases and are treated as a sample from a wider population of domain experts, and form 4 because the analysis uses the mean of the four evaluators rather than a single rating. We report absolute agreement rather than consistency, since the Likert scale carries a fixed meaning and systematic leniency differences should therefore count against agreement. Values below 0.50 are conventionally regarded as poor [7]. Because the coefficient is estimated from variance components, it can take small negative values when the between-test-case component is indistinguishable from zero; such estimates are interpreted as zero agreement.

Analyses were implemented in Python using SciPy [17] and NumPy; the complete analysis script is provided as supplementary material and is described in Appendix A.

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

4. EXPERIMENTS AND RESULTS

Across all 200 evaluator–test-case pairs (4 evaluators × 50 unique test cases), the overall mean rating is $\bar{x} = 3.60$ (on a 1–5 Likert scale) with a pair-level standard deviation of $\sigma = 0.77$ and individual scores ranging from 2.0 to 5.0. All 50 configurations returned a complete response with finish reason STOP and no recorded execution failure, and every generated test case was rated by all four evaluators. When the four evaluator ratings are first averaged per test case, the between-test-case standard deviation is 0.22; the two values describe different aggregation levels and should not be interchanged. Figure 1 shows the distribution of the 50 per-test-case mean ratings.

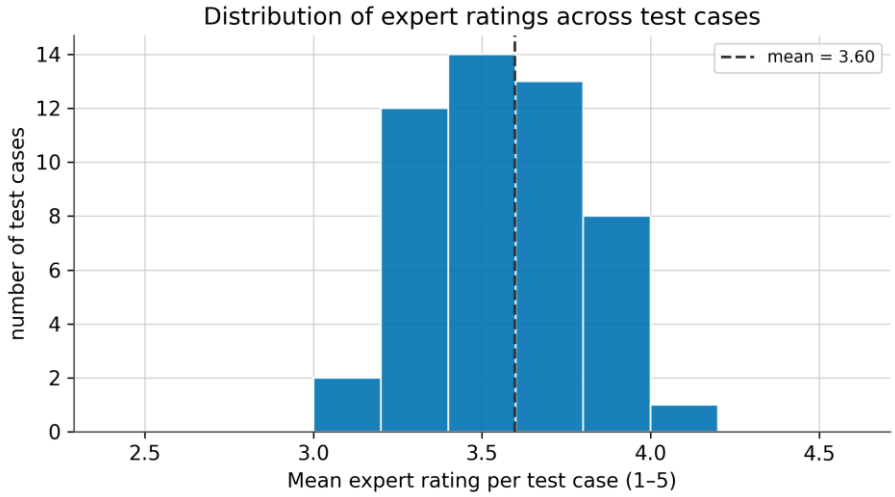

Figure 1. Distribution of per-test-case mean expert ratings across the 50 test cases (overall mean 3.60).

4.1 Overall Technique Comparison

Figure 2 shows the distribution of the 50 per-test-case mean ratings by prompt technique. Technique means lie within a narrow 0.30-point band (3.39–3.69) around the overall mean of 3.60, and the omnibus non-parametric comparison did not detect a statistically significant difference between the prompt-technique groups (see Table 5). Table 4 reports the corresponding descriptive statistics; all values are computed on the per-test-case mean rating, the unit of analysis used throughout Section 4.

Technique	n	Mean	Median	SD	IQR	95% CI
Few-Shot	**5**	**3.69**	**3.75**	**0.30**	**0.40**	**[3.47, 3.91]**
Role-Play	**6**	**3.65**	**3.62**	**0.17**	**0.29**	**[3.54, 3.77]**
Self-Consistency	**6**	**3.64**	**3.65**	**0.25**	**0.44**	**[3.48, 3.80]**
ToT	**5**	**3.64**	**3.55**	**0.23**	**0.15**	**[3.49, 3.82]**
Emotional	**7**	**3.64**	**3.60**	**0.17**	**0.23**	**[3.52, 3.76]**
Zero-Shot	**7**	**3.62**	**3.65**	**0.24**	**0.40**	**[3.46, 3.78]**
ThoT	**7**	**3.56**	**3.50**	**0.19**	**0.30**	**[3.44, 3.69]**
Zero-Shot-CoT	**7**	**3.39**	**3.30**	**0.23**	**0.38**	**[3.24, 3.54]**

Table 4. Descriptive statistics per prompt technique, computed on the per-test-case mean rating. CI = 95% bootstrap confidence interval of the mean (10,000 resamples).

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

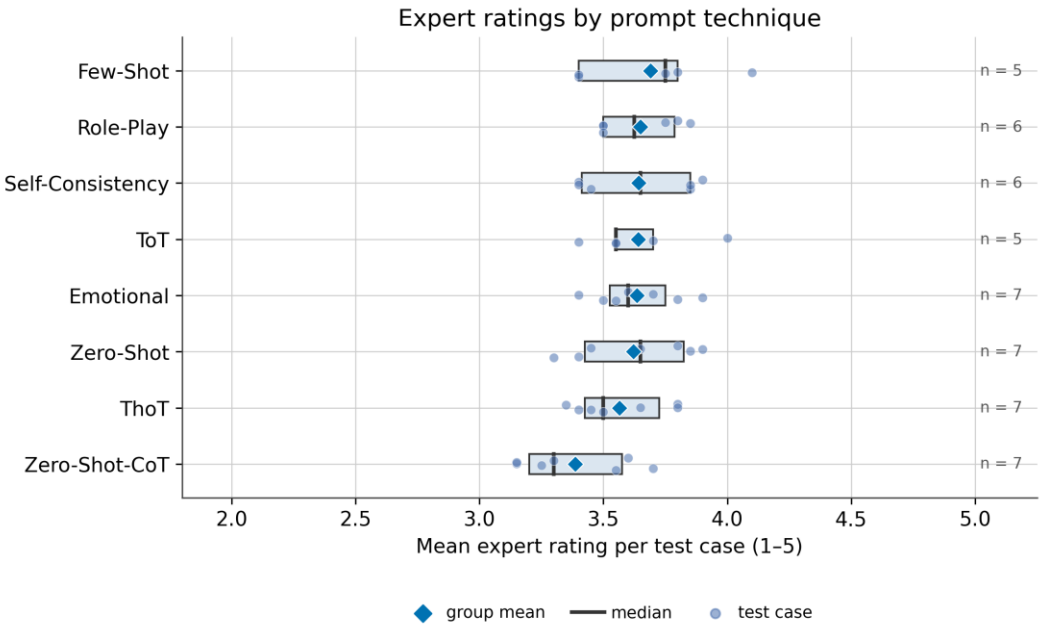

Figure 2. Per-test-case mean ratings by prompt technique. Each point is one test case, averaged over the four evaluators.

A detailed heatmap analysis of the five individual evaluation criteria (Figure 3) confirms that no single technique dominates across all dimensions. Few-Shot leads on Clear Preconditions (3.95); Emotional Prompting and Zero-Shot tie on Repeatability & Consistency (3.89); Self-Consistency leads on Creativity (3.79); Role-Play scores highest on Domain Knowledge (3.62), meaning its test cases require the least expert knowledge to execute; and Acceptance Criteria is a four-way tie among Emotional, Few-Shot, Role-Play, and ToT (3.75). Zero-Shot-CoT is lowest on four of five criteria.

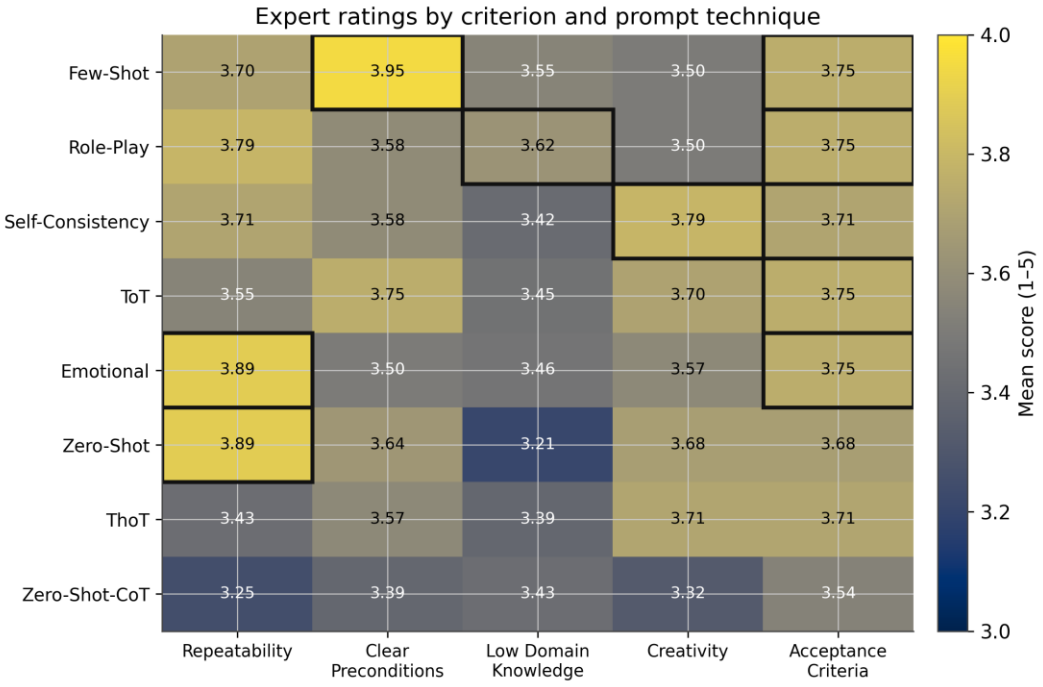

Figure 3. Mean expert scores per criterion and prompt technique. Domain Knowledge is reverse-coded: higher values indicate that less expert knowledge is required.

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

4.2 Model Comparison

All three evaluated LLMs received comparable mean expert ratings above 3.5 on the five-point scale, with only marginal descriptive differences (Figure 4). Claude 3.5 Sonnet obtained the highest descriptive mean ($\bar{x}$ = 3.65), followed by GPT-4o ($\bar{x}$ = 3.59) and Claude 3 Haiku ($\bar{x}$ = 3.55). The differences were small, the 95% bootstrap confidence intervals of the means overlap heavily, and the omnibus comparison was not statistically significant (see Table 5). The data therefore do not establish that one model consistently outperforms the others.

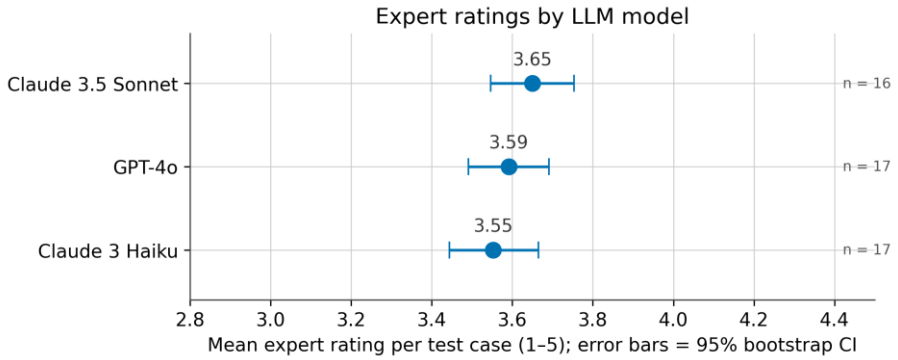

Figure 4. Expert ratings by LLM model. Markers show group means of the per-test-case ratings, error bars 95% bootstrap confidence intervals.

4.3 Effect of Sampling Parameters

Neither sampling parameter showed a meaningful association with rated quality: the rank correlation between temperature and the per-test-case mean rating is ρ = 0.148 (p = 0.305), and between top_p and the mean rating ρ = 0.088 (p = 0.543). Figure 5 shows the individual ratings against both parameters.

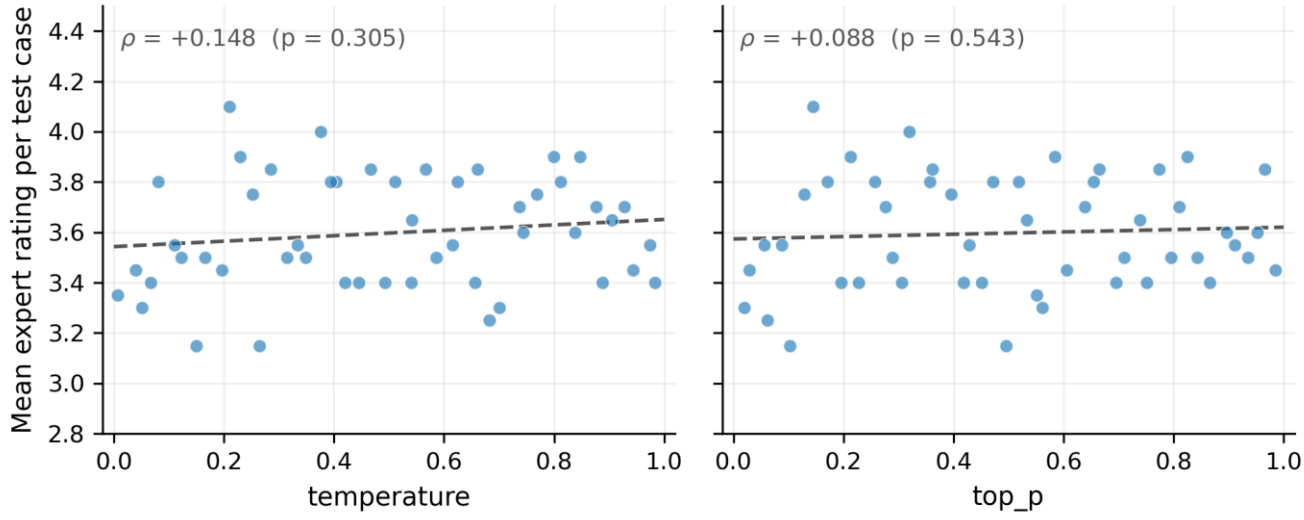

Figure 5. Per-test-case mean rating against the two sampling parameters. Dashed lines show linear fits; neither parameter is associated with rated quality.

Grouping the parameters into thirds yields the same picture. For temperature (low: [0.00, 0.33), medium: [0.33, 0.66), high: [0.66, 1.00]), the group means differ only marginally (Figure 6): 3.54, 3.63, and 3.62 respectively. For top_p, grouped analogously, the three group means are practically identical (3.59, 3.60, 3.60; Figure 7). In both cases the 95% bootstrap confidence intervals overlap almost completely, and no statistically significant difference was detected between the sampled parameter groups (see Table 5). Neither sampling parameter therefore emerges as a lever for test case quality within the range examined here.

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

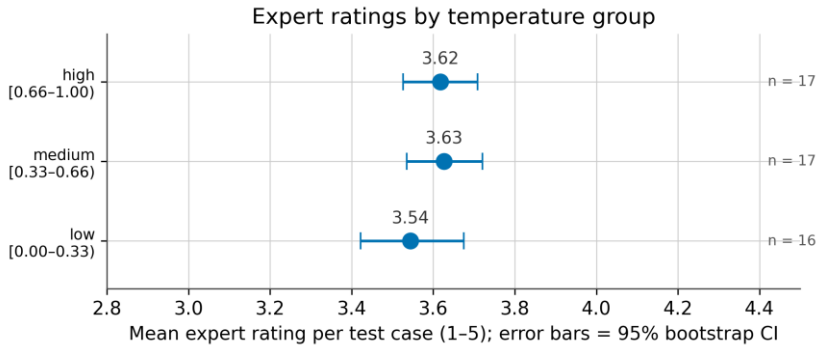

Figure 6. Expert ratings by temperature group. Markers show group means, error bars 95% bootstrap confidence intervals.

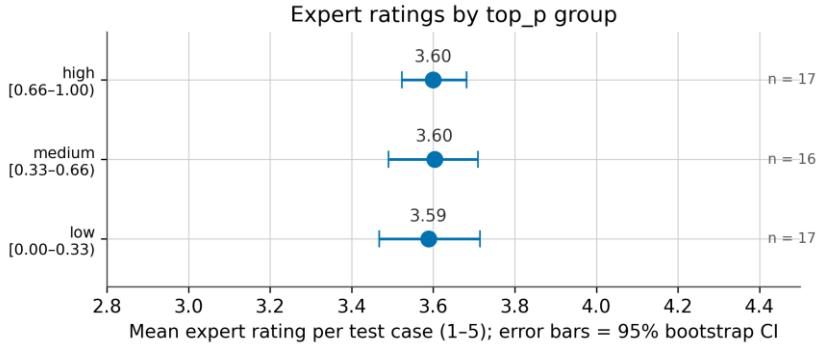

Figure 7. Expert ratings by top_p group. Markers show group means, error bars 95% bootstrap confidence intervals.

Comparison	Groups	Unit	H	df	p	ε^2
Prompt technique	**8**	**50 test cases**	**6.55**	**7**	**0.477**	**0.134**
LLM model	**3**	**50 test cases**	**1.79**	**2**	**0.409**	**0.037**
Temperature group	**3**	**50 test cases**	**1.27**	**2**	**0.529**	**0.026**
top_p group	**3**	**50 test cases**	**0.15**	**2**	**0.928**	**0.003**

Table 5. Omnibus non-parametric comparisons (Kruskal–Wallis) based on per-test-case mean ratings. Effect size ε^2 (epsilon-squared) = H/(N−1), N = 50. Temperature and top_p were grouped as low [0.00, 0.33), medium [0.33, 0.66), high [0.66, 1.00]. No comparison reached significance at $\alpha = 0.05$; no post-hoc tests were performed.

Total token volume per configuration, counting prompt and completion tokens together, was examined as a further descriptive covariate; because the count includes the prompt, techniques with long templates are not directly comparable to techniques with short ones. A scatter plot against overall rating (Figure 8) shows no association ($\rho = 0.031$, $p = 0.832$). The configurations above 3,000 tokens are exclusively Self-Consistency runs, whose token count aggregates six API calls rather than one generated response and is therefore not comparable to the other techniques. Among the single-call techniques, Few-Shot produces the longest outputs (median 1,931 tokens, reflecting the

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

included examples) and Zero-Shot, Emotional Prompting, and Role-Play the shortest (medians near 1,050). No monotonic association between output length and rated quality was detected.

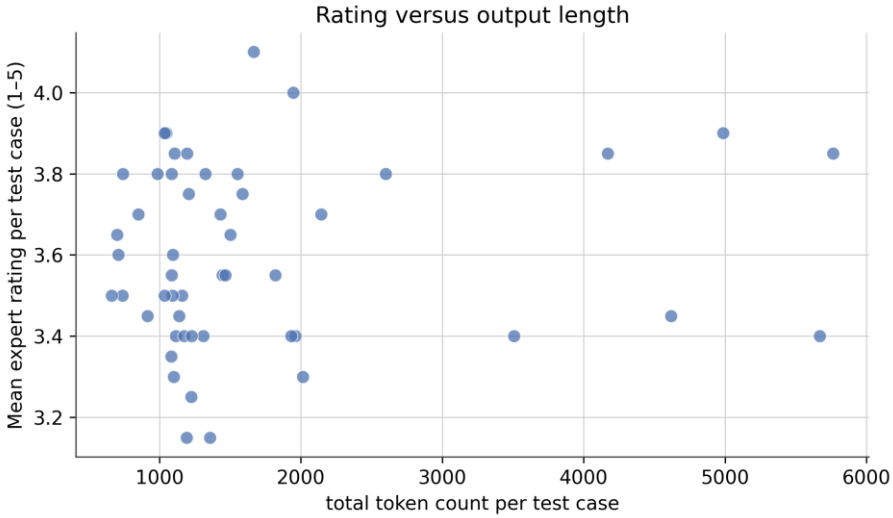

Figure 8. Per-test-case mean rating versus total token count (n = 50). Configurations above 3,000 tokens are Self-Consistency runs, which aggregate six API calls.

4.4 Self-Consistency Consolidation Analysis

Self-Consistency uses a 5+1 approach: five independent Chain-of-Thought generations are sent to the LLM, and a sixth consolidation call asks the model to identify the most consistent answer among the five responses. In this sample it reaches an upper-mid-range median ($\tilde{x}$ = 3.65) and the highest Creativity score of all techniques (3.79), but also the widest interquartile range (IQR = 0.44). These marginal descriptive comparisons provide no evidence that the consolidation step reduced rating variability in this sample.

The additional computational cost is substantial: six API calls per test case instead of one, and a median of 4,803 tokens against 1,036 to 1,931 for the single-call techniques. Within this sample that cost did not translate into a statistically distinguishable improvement in overall rating. The Creativity advantage aligns qualitatively with the theoretical motivation of Wang et al. [18], who argue that marginalizing over multiple reasoning paths improves LLM reliability; a formal cost–benefit assessment is left to future work.

4.5 Inter-Rater Reliability

Because the four evaluators independently rated the same 50 test cases, we assessed their agreement on the overall score. Agreement on the relative ranking of test cases was very low: the two-way random-effects, absolute-agreement, average-measures intraclass correlation was ICC(2,4) = −0.01, and the mean pairwise Spearman correlation was 0.002. Kendall's coefficient of concordance, corrected for ties, was W = 0.252, which did not reject the null hypothesis of no concordance (χ^2 = 49.37, df = 49, p = 0.459). Because the overall scores are means of five Likert items, they contain a substantial number of tied values within each evaluator, and the tie correction is therefore required. The three measures are mutually consistent: the approximate identity $\bar{\rho} = (kW - 1)/(k - 1)$ yields $\bar{\rho} = 0.003$ for k = 4 and W = 0.252, close to the observed mean Spearman correlation of 0.002.

At the same time, a Friedman test revealed strong and highly significant differences in rating level between the evaluators (χ^2 = 108.6, df = 3, $p \approx 2 \times 10^{-23}$). Per-evaluator means of the overall score were E1: 3.46, E2: 4.59, E3: 2.85, and E4: 3.49, a range of 1.74 points on a five-point scale. These analyses address different aspects of the rating process: Kendall's W and the ICC evaluate concordance in the relative ranking of test cases, whereas the Friedman test evaluates systematic differences in rating level. The evaluators thus applied the scale with different leniency while showing almost no agreement on the comparative ranking. Scale usage also differed in range: E2 produced only five distinct overall scores across all 50 test cases (SD = 0.22), while E4 produced fourteen (SD = 0.67).

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

This is not a judgement of the evaluators as individuals; the criteria are inherently interpretive and no calibration round preceded the survey. It does mean, however, that the averaged ratings can mask substantial disagreement, and that the small descriptive differences between techniques and models must be interpreted with caution rather than as robust evidence of superiority. Section 4.6 examines how much of the observed variation is attributable to the test cases at all.

4.6 Where the Variance Sits

The low inter-rater agreement raises a further question: how much of the observed variation in ratings is attributable to the test cases at all? A first indication comes from the raw score ranges. Across all 50 test cases, evaluator E2 never awarded an overall score below 4.2 and evaluator E3 never above 3.6. The two ranges do not overlap at any point, so no test case exists on which these two evaluators could have agreed in absolute level; agreement on the relative ranking is assessed separately below.

To quantify this, variance components were estimated from a two-way ANOVA without replication on the 50 × 4 matrix of overall scores. The analysis rests on three mean squares: MSR between test cases, MSC between evaluators, and MSE for the residual. Because each mean square already contains the residual variation, the residual is subtracted to obtain the underlying variance components. Table 6 reports both the mean squares and the resulting estimates, so that every value can be traced from the three inputs.

Source	Mean square	Component	Estimate	Share
Between test cases	**MSR = 0.2014**	**(MSR − MSE) / k**	**−0.0010**	**≈ 0%**
Between evaluators	**MSC = 26.2802**	**(MSC − MSE) / n**	**0.5215**	**71.8%**
Residual (test case × evaluator)	**MSE = 0.2055**	**MSE**	**0.2055**	**28.3%**

Table 6. Variance decomposition of the overall score across all 50 test cases and four evaluators (n = 50, k = 4). Shares are relative to the sum of the three components (0.7260).

Two features of Table 6 carry the result. First, MSR falls below MSE: the variation between test cases does not rise above the residual noise level, and the between-test-case component is therefore not distinguishable from the residual ($F = MSR/MSE = 0.98$, $df = 49$ and 147, $p = 0.52$). The component is consequently estimated as slightly negative, the conventional indication that the underlying variance is indistinguishable from zero; equivalently, the observed standard deviation between test-case means (0.22) matches the value expected from residual noise alone (0.23). Second, the mean square between evaluators exceeds the mean square between test cases by a factor of 130. Because all four evaluators rated all 50 test cases, however, this second feature enters every per-test-case mean as the same constant and cancels in the group comparisons: it describes how the panel used the scale, not why the comparisons could not resolve a difference. The same three mean squares yield the intraclass correlation reported in Section 4.5, $ICC(2,4) = (MSR − MSE) / (MSR + (MSC − MSE)/n) = −0.01$, so the two analyses are not independent findings but two readings of the same decomposition. This result constrains the interpretation of the omnibus tests reported above. The between-test-case component aggregates every source that differs across test cases, including requirement, technique, model and sampling parameters; that this component is indistinguishable from zero means the ratings carry little resolvable signal at the level on which the group comparisons operate. Together with the small and unbalanced groups, the residual test-case-by-evaluator variation and the low agreement materially limit the sensitivity and interpretation of all three marginal comparisons.

5. INTERPRETATION AND CONSIDERATIONS

5.1 Key Findings

RQ1 (Technique selection): Few-Shot obtained the highest descriptive mean (3.69) and Zero-Shot-CoT the lowest (3.39), but the omnibus non-parametric comparison did not detect a statistically significant difference between the prompt-technique groups (Table 5). No technique can be identified as superior from these data. Descriptively, Role-Play and Emotional Prompting produced the most stable ratings (SD = 0.17 each), ToT the narrowest interquartile range (IQR = 0.15), and Few-Shot the largest spread (SD = 0.30); Self-Consistency stood out only on the Creativity criterion.

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

Prompt technique showed a larger effect ($\varepsilon^2 = 0.134$) than model choice (0.037) or either sampling parameter (0.026 and 0.003), but at 50 test cases the design is underpowered for an effect of this size, so the comparison remains inconclusive rather than negative.

RQ2 (Model comparison): All three LLMs received comparable mean expert ratings above 3.5 on the five-point scale. Claude 3.5 Sonnet obtained the highest descriptive mean (3.65), followed by GPT-4o (3.59) and Claude 3 Haiku (3.55). The differences were small and the omnibus comparison was not statistically significant (Table 5). The ranking should therefore not be read as evidence of model superiority. In the absence of statistically significant quality differences, cost, latency, and availability may be considered as additional model-selection criteria; however, these factors were not evaluated in this study.

RQ3 (Parameter effects): Neither the temperature groups nor the top_p groups produced statistically significant differences within the available sample (Table 5). No universally optimal parameter configuration can be identified from these data; the observed between-group differences were small relative to the within-group variation.

Scope of the group comparisons: The LHS design varies prompt technique, model, temperature, top_p, and requirement simultaneously. The four omnibus tests are therefore marginal comparisons that do not hold the remaining factors constant, and with 50 unbalanced configurations they cannot isolate main effects. They should be read as exploratory group comparisons rather than as estimates of the causal effect of any single factor.

Evaluation reliability: The most consequential result of this study was not obtained from the three research questions but from the assessment of the evaluation instrument itself. Across the same 50 test cases, the four evaluators agreed almost not at all on which test cases were better ($ICC(2,4) = -0.01$) while differing systematically and strongly in how leniently they used the scale. A variance decomposition shows that the between-test-case component is not distinguishable from the residual test-case-by-evaluator variation ($F = 0.98$, $p = 0.52$), so the observed spread between test cases is fully accounted for by measurement noise.

5.2 Practical Implications

Technique recommendation: Given the low inter-rater agreement documented in Section 4.5, these data provide no basis for preferring one technique over another, and equally none for treating them as interchangeable (Limitation 10). The choice should be validated in the practitioner's own setting rather than imported from a ranking obtained elsewhere.

Parameter guidance: Within the sampled ranges, no statistically significant evidence was found that changing temperature or top_p systematically improved expert ratings; the observed between-group differences were small relative to the within-group variation.

Cost-benefit of Self-Consistency: Self-Consistency requires roughly six times the API calls of single-pass techniques and produced no statistically distinguishable improvement in rated quality. Its descriptive advantage on Creativity was not tested for significance and, given the reliability of the rating instrument, is not a basis for accepting the additional cost. A formal cost, latency, and functional-safety assessment remains future work; no claim about suitability for safety-critical deployment is made.

Evaluation practice: The most actionable finding of this study concerns the evaluation procedure rather than the generation techniques. Before comparing prompt techniques or models, practitioners should establish that their rating procedure can resolve the differences they intend to measure: a calibration round with anchor examples, an explicit rubric per criterion, and a reported agreement statistic. Without these, an averaged expert score may reflect the composition of the panel more than the quality of the artefacts under review.

5.3 Threats to Validity

Internal validity: The LHS design with 50 samples may not capture all parameter interactions, and the statistical power of the group comparisons is limited by the number of test cases, by the number of raters per test case, and above all by the reliability of the rating instrument itself; accordingly, none of the omnibus group comparisons reached significance. Together with the partial dependence between configurations, these constraints are quantified in Section 5.5.

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

External validity: Results are specific to ADAS requirements from the participating automotive OEM and may not directly generalize to other manufacturers or software domains. The three LLM models evaluated represent a snapshot; newer models may yield different results.

Construct validity: Expert evaluation is inherently subjective. Structured criteria and multiple evaluators were intended to mitigate this, but the reliability analysis in Section 4.5 shows that they did not: agreement on the relative ranking of test cases was effectively zero, and averaging across four evaluators produced an apparent robustness the underlying ratings do not support. The overall score used throughout Section 4 should therefore be understood as a noisy proxy for test case quality rather than as a validated measurement.

5.4 Why Prompt Engineering over Fine-Tuning

A deliberate design decision in this work is the exclusive focus on prompt engineering rather than fine-tuning or other model adaptation techniques. Prompt engineering operates as a black-box approach—the pre-trained model remains unmodified, and the prompt itself serves as the sole variable. This avoids the need for domain-specific training data curation and ensures that the approach remains model-agnostic: if a newer or better-performing LLM becomes available, the same prompt templates can be reused without retraining. Furthermore, in enterprise environments where LLM access is governed by centralized API gateways (as in our industrial setting), fine-tuning may not even be technically feasible due to infrastructure constraints.

5.5 Limitations

We acknowledge several limitations that should be considered when interpreting the results:

1. **Statistical power.** The LHS design with 50 configurations limits statistical power; individual techniques are represented by only five to seven test cases each. No group comparison reached statistical significance, and the effect size for prompt technique was $\varepsilon^2 = 0.134$. At the observed effect size, the estimated power of the technique comparison at 50 test cases is 0.41 under the non-central chi-square approximation and 0.38 in simulation; reaching 0.80 would require roughly 109 test cases (Section 3.7), conditional on an instrument that can resolve an effect of this size; with the instrument used here, a larger sample would measure the noise more precisely rather than the effect. Because the effect size is itself estimated from 50 observations and small-sample effect sizes are biased upwards, this figure is an optimistic lower bound. The comparisons are inconclusive rather than negative, and all reported rankings are exploratory and descriptive.
2. **Evaluation reliability.** Four evaluators from six invited participated. Agreement among them on the relative ranking of test cases was effectively zero (ICC(2,4) = −0.01), while a Friedman test found strong differences in scale usage ($p \approx 2 \times 10^{-23}$; per-evaluator means 2.85–4.59). Averaged ratings may therefore mask substantial heterogeneity, and small descriptive differences are sensitive to evaluator-specific scoring behaviour. No calibration round preceded the survey.
3. **Non-independent observations.** The 50 configurations draw on 43 distinct requirements; seven requirements were sampled twice under different technique or model combinations. The observations are therefore not fully independent, which slightly narrows the effective sample size.
4. **Non-reproducible original sampling.** The original sampling script did not fix a random seed, so it cannot regenerate the executed design. Reproducibility here refers to re-analysing the preserved executed configurations, not to regenerating an identical LHS design.
5. **Subjective evaluation.** Test case quality was assessed exclusively through expert ratings on a 1–5 Likert scale. No automated quality metrics were applied in this study.
6. **No baseline comparison.** The study compares prompt techniques against each other, not against manually authored test cases or existing automated test generation tools.
7. **Scope of the sample.** All requirements originate from a single ADAS development project at the participating automotive OEM, and the three evaluated LLMs represent a snapshot accessed in March 2025. Newer models may yield substantially different results; the findings should be interpreted as technique-relative rather than model-absolute.
8. **Prompt engineering only.** Fine-tuning, Retrieval-Augmented Generation, and agentic multi-step workflows were not compared. These may yield superior results but require additional infrastructure, training data, and, in safety-critical

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

domains, regulatory re-validation. No formal cost, latency, or functional-safety analysis was performed, and no claim is made about suitability for safety-critical deployment.

9. Language and artefact availability. Prompts and generated test cases are in English, the original requirements contain both English and German elements, and the evaluation survey was administered in German. Language-mixing effects were not systematically investigated. The tools and prompt templates were developed as part of this study; their publication is subject to organizational approval.

10. Non-equivalence. The study tested for differences, not for equivalence. The absence of statistically significant differences does not establish that the techniques, models, or parameter settings are of comparable quality.

6. CONCLUSIONS

This paper presents a systematic evaluation of eight prompt engineering techniques for LLM-based test case generation in automotive ADAS development. Through an LHS-based experiment design spanning five dimensions and a fully crossed evaluation by four automotive domain experts, we examined whether the choice of prompt technique, model, or sampling parameter measurably affects the quality of generated test cases.

All eight techniques and all three models produced test cases that received moderate-to-high mean expert ratings. Few-Shot achieved the highest descriptive technique mean and Claude 3.5 Sonnet the highest descriptive model mean, but no comparison between techniques, models, or sampling parameters reached statistical significance. A post-hoc analysis of the evaluation instrument itself constrains how these results can be read: the between-test-case variance component is not distinguishable from the residual test-case-by-evaluator variation, so the ratings carry little resolvable signal at the level on which the comparisons operate, and the evaluators differed systematically in how leniently they used the scale. Together with the small and unbalanced groups and the simultaneous variation of all design factors, the low agreement and the residual test-case-by-evaluator variation materially limit the interpretability and sensitivity of the three marginal comparisons.

The practical consequence is that this study could not resolve differences between prompt techniques under the evaluation design used, and that an uncalibrated four-rater panel was not a sufficient instrument for the comparisons attempted here. Future evaluations should treat rater calibration as a design requirement rather than an afterthought: anchor examples, an explicit rubric per criterion, a reported agreement statistic, and a sample size derived from a power analysis. For the technique effect observed here, and under the observed distributions, this corresponds to an estimated sample size of roughly 109 test cases rather than 50 but only once the instrument itself can resolve differences of that size; the sample size is the second requirement, not the first.

Our purpose-built toolchain (PET for prompt authoring, PEBatch2 for batch execution) demonstrates a practical, reproducible workflow for large-scale prompt engineering experiments. The tools and prompt templates were developed as part of this study; their future publication is subject to organizational approval.

Future work includes: comparing prompt engineering with fine-tuning and Retrieval-Augmented Generation approaches; conducting a larger evaluation with a calibrated evaluator panel and a sample size derived from the effect sizes reported here; extending the evaluation to newer LLM models; and exploring automated evaluation metrics for test case quality to complement human assessment.

Author Contributions

A. Zosimidis conceptualized the research, designed the methodology, developed the tools, performed the experiments, analyzed the data, and wrote the paper. S.-A. Schneider supervised the research and reviewed the manuscript. All authors read and approved the final version.

Disclosure of AI Tool Usage

This research involves the use of LLMs (GPT-4o, Claude 3.5 Sonnet, Claude 3 Haiku) as the subject of study; the generated test cases are the research outputs being evaluated. AI assistants were additionally used during manuscript preparation for language editing, for implementing and reviewing the statistical analysis scripts, and for drafting and revising portions of the text. All statistics, tables, and figures were regenerated from the preserved raw data by a single

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

analysis script rather than transcribed from earlier drafts, and every reported value was verified against the raw data by the authors. No experimental result was substituted, interpolated, or fabricated. The authors are solely responsible for the accuracy and appropriateness of all content.

References

[1] Brown, T.B. et al. (2020). "Language Models are Few-Shot Learners." Advances in Neural Information Processing Systems 33 (NeurIPS). arXiv:2005.14165. DOI: 10.48550/arXiv.2005.14165

[2] Chen, B. et al. (2024). "Unleashing the Potential of Prompt Engineering in Large Language Models: A Comprehensive Review." arXiv:2310.14735. DOI: 10.48550/arXiv.2310.14735

[3] Friedman, M. (1937). "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance." Journal of the American Statistical Association, 32(200), pp. 675–701. DOI: 10.1080/01621459.1937.10503522

[4] Kendall, M.G. & Babington Smith, B. (1939). "The Problem of m Rankings." The Annals of Mathematical Statistics, 10(3), pp. 275–287. DOI: 10.1214/aoms/1177732186

[5] Kojima, T. et al. (2023). "Large Language Models are Zero-Shot Reasoners." arXiv:2205.11916. DOI: 10.48550/arXiv.2205.11916

[6] Kong, A. et al. (2024). "Better Zero-Shot Reasoning with Role-Play Prompting." arXiv:2308.07702. DOI: 10.48550/arXiv.2308.07702

[7] Koo, T.K. & Li, M.Y. (2016). "A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research." Journal of Chiropractic Medicine, 15(2), pp. 155–163. DOI: 10.1016/j.jcm.2016.02.012

[8] Kruskal, W.H. & Wallis, W.A. (1952). "Use of Ranks in One-Criterion Variance Analysis." Journal of the American Statistical Association, 47(260), pp. 583–621. DOI: 10.1080/01621459.1952.10483441

[9] Li, C. et al. (2023). "Large Language Models Understand and Can be Enhanced by Emotional Stimuli." arXiv:2307.11760. DOI: 10.48550/arXiv.2307.11760

[10] Long, J. (2023). "Large Language Model Guided Tree-of-Thought." arXiv:2305.08291. DOI: 10.48550/arXiv.2305.08291

[11] McKay, M.D., Beckman, R.J. & Conover, W.J. (1979). "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code." Technometrics, 21(2), pp. 239–245. DOI: 10.1080/00401706.1979.10489755

[12] Sahoo, P. et al. (2024). "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications." arXiv:2402.07927. DOI: 10.48550/arXiv.2402.07927

[13] Schulhoff, S. et al. (2024). "The Prompt Report: A Systematic Survey of Prompting Techniques." arXiv:2406.06608. DOI: 10.48550/arXiv.2406.06608

[14] Shanahan, M. et al. (2023). "Role-Play with Large Language Models." arXiv:2305.16367. DOI: 10.48550/arXiv.2305.16367

[15] Shrout, P.E. & Fleiss, J.L. (1979). "Intraclass Correlations: Uses in Assessing Rater Reliability." Psychological Bulletin, 86(2), pp. 420–428. DOI: 10.1037/0033-2909.86.2.420

[16] Tomczak, M. & Tomczak, E. (2014). "The Need to Report Effect Size Estimates Revisited. An Overview of Some Recommended Measures of Effect Size." Trends in Sport Sciences, 1(21), pp. 19–25.

[17] Virtanen, P. et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." Nature Methods, 17, pp. 261–272. DOI: 10.1038/s41592-019-0686-2

[18] Wang, X. et al. (2023). "Self-Consistency Improves Chain of Thought Reasoning in Language Models." arXiv:2203.11171. DOI: 10.48550/arXiv.2203.11171

[19] Wei, B. (2024). "Requirements are All You Need: From Requirements to Code with LLMs." In Proc. IEEE 32nd International Requirements Engineering Conference (RE), pp. 416–422. DOI: 10.1109/RE59067.2024.00049

[20] Wei, J. et al. (2023). "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." arXiv:2201.11903. DOI: 10.48550/arXiv.2201.11903

[21] Xu, B. et al. (2023). "ExpertPrompting: Instructing Large Language Models to be Distinguished Experts." arXiv:2305.14688. DOI: 10.48550/arXiv.2305.14688

[22] Yao, S. et al. (2023). "Tree of Thoughts: Deliberate Problem Solving with Large Language Models." arXiv:2305.10601. DOI: 10.48550/arXiv.2305.10601

[23] Zhou, Y. et al. (2023). "Thread of Thought Unraveling Chaotic Contexts." arXiv:2311.08734. DOI: 10.48550/arXiv.2311.08734

Appendix

A. Reproducibility

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD

Both tools (PET and PEBatch2) are implemented in Python 3.x. PET uses Tkinter for the GUI, openpyxl/xlsxwriter for Excel I/O, and stores prompt templates in a JSON dictionary (methods.json). PEBatch2 uses pyDOE for LHS, boto3 for AWS authentication, and implements async LLM API polling with a 120-second timeout. The experiment design matrix is generated with 5 dimensions and 50 samples using pyDOE.lhs(); the original sampling script did not fix a random seed, so the executed configurations were preserved together with the model responses and serve as the authoritative record for all analyses. The descriptive statistics, non-parametric tests (Kruskal–Wallis with epsilon-squared effect sizes, Friedman rater-effect test, and inter-rater reliability) reported in this paper are computed from these preserved configurations using a fixed analysis seed (20260815); the figures are generated separately from the exported per-test-case data. PEBatch2 includes automated unit tests covering the sampling routine and the Self-Consistency consolidation path.

The complete analysis script, which reads the preserved execution record and the four evaluation workbooks and regenerates every statistic reported in Sections 4 and 5, is provided as supplementary material. The survey workbooks carry no configuration identifier, so ratings are aligned to executed configurations by position. The script asserts that alignment before any analysis rather than assuming it, verifying that all four workbooks contain the same 50 items in the same order, that the sequence of requirement identifiers matches the execution record, and that no rating is missing or out of range. Its reference output, which lists every statistic reported in Sections 4 and 5, is included so that each value in the paper can be traced to a single reproducible computation.

The underlying requirement texts and their identifiers originate from a proprietary requirements database of the participating automotive OEM and cannot be published; the generated test cases are derived from these requirements and are subject to the same restriction. The analysis script therefore cannot be executed without approved access to these files; the supplementary material documents the complete analysis pipeline and all intermediate statistics, and the included reference output is the verifiable record of every statistic reported here.

Discrete dimensions are indexed in the order in which the technique files are concatenated in the sampling script: Zero-Shot, ToT, Zero-Shot-CoT, Self-Consistency, ThoT, Few-Shot, Emotional, Role-Play; the model dimension is indexed as gpt-4o, Claude 3.5 Sonnet, Claude 3 Haiku. Each technique pool contains 53 filled prompts derived from the same set of ADAS requirements.

B. Environment

LLM models used: anthropic.claude-3-5-sonnet-20240620-v1:0, anthropic.claude-3-haiku-20240307-v1:0, gpt-4o. All 50 configurations were executed on 7 March 2025. API accessed via an internal REST interface with OAuth2 (M2M) authentication. Python packages: pyDOE, openpyxl, xlsxwriter, boto3, pandas, requests, numpy, scipy.

C. Variance Decomposition and Intraclass Correlation

The decomposition in Section 4.6 follows the standard two-way ANOVA without replication on the matrix of n = 50 test cases by k = 4 evaluators, with grand mean $\bar{x} = 3.597$. Writing $\bar{x}_{i\cdot}$ for the mean of test case i and $\bar{x}_{\cdot j}$ for the mean of evaluator j, the three mean squares are $MSR = k \cdot \Sigma_i (\bar{x}_{i\cdot} - \bar{x})^2 / (n - 1) = 0.2014$, $MSC = n \cdot \Sigma_j (\bar{x}_{\cdot j} - \bar{x})^2 / (k - 1) = 26.2802$, and $MSE = \Sigma_{ij} (x_{ij} - \bar{x}_{i\cdot} - \bar{x}_{\cdot j} + \bar{x})^2 / ((n - 1)(k - 1)) = 0.2055$. The variance components follow as $\sigma^2_{testcase} = (MSR - MSE)/k = -0.0010$, $\sigma^2_{evaluator} = (MSC - MSE)/n = 0.5215$, and $\sigma^2_{residual} = MSE = 0.2055$. The intraclass correlation in the average-measures, absolute-agreement form of Shrout and Fleiss [15] is $ICC(2,4) = (MSR - MSE) / (MSR + (MSC - MSE)/n) = -0.0041 / 0.7229 = -0.0057$, reported as −0.01. The corresponding single-measure coefficient is $ICC(2,1) = -0.001$. Per-evaluator means and standard deviations of the overall score are E1: 3.46 (0.44), E2: 4.59 (0.22), E3: 2.85 (0.37), and E4: 3.49 (0.67).

D. Sensitivity to Repeated Requirements

Seven requirements appear twice among the 50 configurations. Restricting the analysis to the first configuration drawn for each requirement (n = 43) leaves the technique comparison unchanged: $H = 6.09$, $df = 7$, $p = 0.529$, $\varepsilon^2 = 0.145$. This is a simple robustness check rather than a substitute for a mixed model that would account for the requirement as a random effect.

Recieved date: May 29 2026
Revised date: Aug.7, 18, Oct.20
Accepted date: July 7 2026

DOI: https://10.63211/j.p.26.204503
ISSN 2760-8646 (Online) ICAITD