When the Trigger Fails: Metamorphic Testing of Tool-Calling AI Agents

Wyllgner França de Amorim
Federal University of Paraná
Curitiba, Brazil
wyllgner.amorim@proton.me

Abstract

Software agents built on foundation models depend on the *prompt*, the *Trigger* component, as their primary instruction interface, yet this component appears in roughly 1% of the test cases across the agent ecosystem. Recent literature calls for metamorphic testing of prompt behavior, but that call remains unanswered for the agentic case. We propose and evaluate a catalog of metamorphic relations over the Trigger, under the premise that transformations preserving the meaning of the prompt (paraphrase, typing noise, and language switching) should not alter the agent's behavior, defined by the selected tool and its arguments; when they do, a silent behavioral regression is revealed. Across 1,600 controlled invocations of two open models run locally, over 40 *tool-calling* tasks with five runs each, violations prove prevalent and strongly concentrated in language switching (English → Portuguese), which alters behavior in up to 41% of the tasks against fewer than 11% for paraphrase and noise; the inconsistency falls chiefly on the *arguments* rather than on the selected tool; and the test discriminates between the two configurations evaluated (Wilcoxon, $p = 0.025$), even after discounting an intrinsic *flakiness* affecting about 30% of the tasks. We release a reproducible framework with the relations, the data, and the analysis scripts, along with guidelines for integrating prompt regression testing into the development workflow.

Keywords

Metamorphic Testing, AI Agents, Tool Calling, Prompt Robustness, Behavioral Regression, Software Testing

1 Introduction

Software agents built on foundation models (FMs) have moved beyond laboratory prototypes and now make up production systems, from the automation of enterprise workflows to assistants that invoke external tools, query APIs, and coordinate tasks on the user's behalf. In this architecture, the *prompt* is the primary interface for instruction and context, the component that Hasan et al. [12] call the *Trigger* in their agent taxonomy, alongside the *Plan Body*, *Resource Artifacts*, and *Coordination Artifacts*. It is through the Trigger that the developer specifies *what* the agent should do and *how* it should behave; a subtle change in that text can redirect the agent's entire execution trajectory.

Despite occupying a central position, the Trigger component is critically undertested. In the first large-scale empirical study of testing practices in the agent ecosystem, covering 39 open-source frameworks and 439 agentic applications, Hasan et al. [12] reveal an inversion of testing effort: deterministic components, such as tools and workflows, concentrate more than 70% of the tests, while the Trigger appears in about 1% of all test cases. In other words, precisely the component most exposed to the ambiguity of natural language and to FM nondeterminism is the one that receives the least verification coverage.

This gap has not gone unnoticed by the community. Chen et al. [6], in proposing *promptware engineering*, the treatment of prompts as first-class software artifacts, identify prompt testing as a neglected stage of the lifecycle and explicitly call for the development of metamorphic testing techniques aimed at prompt behavior. That call remains, to a large extent, unanswered for the agentic case: *tool-calling* reliability is acknowledged to be fragile, since models frequently select the wrong tool, supply incorrect arguments, or misinterpret the output [11], yet we are not aware of a systematic framework for testing this fragility at the prompt layer, without relying on an absolute oracle, is lacking.

Metamorphic testing (MT) [4] offers exactly such a framework. Rather than requiring the expected result for each input, an oracle that is often unavailable in FM-based systems, MT verifies relations that should hold between related inputs and their outputs. The intuition underpinning this work is straightforward: a transformation that preserves the semantics of the Trigger, whether a paraphrase, the introduction of surface-level typing noise, or the translation of the statement between languages, should not alter the agent's behavior, that is, the selected tool and its essential arguments. When it does, there is a silent behavioral regression: a defect that manifests without the output being, in isolation, "wrong," and that goes unnoticed by conventional tests. We distinguish from the outset the method (MT, an intra-version consistency check) from the phenomenon it reveals, the silent behavioral regression, to avoid confusion with classical version-to-version regression.

The present work differs from the closest line of MT applied to LLMs. Approaches such as LLMorph [7] and metamorphic-relation catalogs for NLP [8] apply MT to isolated LLMs, evaluating textual outputs. We instead apply MT to the Trigger component within an agent's execution context, observing the resulting behavior, the tool selection and its arguments, which connects the technique to the gap of Hasan et al. [12] and the call of Chen et al. [6], and yields an objective comparator that dispenses with a costly LLM-as-judge. To isolate genuine regressions from FM noise, we add a *baseline* relation that repeats the identical input and multiple runs per case, statistically discounting the *flakiness* observable without any transformation [3, 20]. Running open-source models locally further eliminates API costs and strengthens reproducibility against the instability of commercial models [2].

Against this backdrop, this paper offers four contributions. The first is a catalog of metamorphic relations targeting the Trigger component of AI agents, covering semantic equivalence by paraphrase, robustness to noise, and language invariance, anchored

in an objective comparator based on the *(tool, arguments)* pair. The second is an empirical study over two open-source models (`qwen3.5:4b` and `llama3.2:3b`), 40 *tool-calling* tasks, and five runs per case, 1,600 controlled invocations, that measures the prevalence and types of violation and separates genuine sensitivity to the transformation from intrinsic nondeterminism. The third is a set of actionable guidelines for integrating metamorphic testing of the Trigger into agent development and continuous verification. The fourth is a reproducible artifact, comprising the *harness*, the metamorphic relations, the data, and the analysis scripts.

To guide the investigation, we formulate four research questions on the prevalence of violations (**RQ1**), the transformations and inconsistency types that drive them (**RQ2**), the share attributable to genuine sensitivity rather than nondeterminism (**RQ3**), and the discriminative power of the test between two agent configurations (**RQ4**), detailed in Section 4.1.

The remainder of the paper is organized as follows. Section 2 reviews the Trigger component, metamorphic testing, and related work. Section 3 presents the approach, Section 4 the study design, and Section 5 the results. Section 6 discusses implications and guidelines, Section 7 addresses threats to validity, and Section 8 concludes.

2 Background and Related Work

This section reviews the layer under test, the *Trigger* component and the evidence of its fragility, then metamorphic testing as a pseudo-oracle technique and its recent use over LLMs and agents, and closes by positioning our contribution against the closest works.

2.1 The *Trigger* Component and Agent Testing

FM-based agents are decomposed by the taxonomy of Hasan et al. [12] into architectural components, four of which matter here: the *Trigger* (the initial user request or prompt that activates a plan), the *Plan Body* (the FM-generated sequence of reasoning and action steps), the *Resource Artifacts* (external tools and APIs), and the *Coordination Artifacts* (inter-agent communication mechanisms). Although the Trigger governs the entire execution trajectory, it is the component that their empirical study finds least covered by tests, the asymmetry that motivates the present work.

The undertesting of the Trigger is particularly worrying because the sensitivity of LLM behavior to surface-level variations of the prompt is a well-documented phenomenon. Perturbations that preserve the intent (paraphrases or typing noise) can substantially alter a model's performance [17], and the observed inconsistency goes beyond generative randomness and paraphrasing, manifesting even under controlled conditions [1]. There is, however, an open debate about whether such sensitivity constitutes a defect of the model or an artifact of the evaluation protocol [13], a debate that directly informs our methodological decision to discount intrinsic nondeterminism (Section 4).

2.2 Metamorphic Testing

Metamorphic testing (MT), introduced by Chen et al. [4], attacks the *oracle problem*: the absence, in many systems, of a mechanism that decides whether the output of an arbitrary input is correct. Instead of verifying isolated outputs, MT specifies *metamorphic relations* (MRs), that is, properties that should be preserved between related inputs and their corresponding outputs. An MR is violated when a pair (source input, derived input) produces outputs that break the expected relation, signaling a defect without the absolute correct result being known. After two decades of maturation, consolidated in comprehensive *surveys* [5], MT has established itself as a pseudo-oracle technique applicable to domains in which exact oracles are infeasible, exactly the case of FM-based agents.

2.3 Metamorphic Testing Applied to LLMs and Agents

The application of MT to LLMs is an active and rapidly expanding line of research [21]. At the level of the isolated model, Cho et al. [8] catalog 191 MRs for natural language processing from a literature review, and automate 36 of them in the LLMorph tool [7]. These efforts, however, evaluate the *textual outputs* of an LLM treated as a black box, over standard NLP benchmarks, and not the behavior of an agent that selects and parameterizes tools.

At the systems level, MT has been combined with multi-agent flows: ARMeta [14] employs a *workflow* of multiple LLM agents to generate and execute metamorphic tests of REST APIs documented in OpenAPI. In that case, however, the agents are the testing *instrument*, and the system under test is the API; the present work inverts that relationship, taking the *tool-calling* agent itself as the system under test. Closest to us in spirit, de Zarzà et al. [9] subject reasoning agents to eight semantics-preserving edits of the statement; they observe, however, the *textual* response and judge it by semantic similarity, whereas we observe the invocation and decide equivalence exactly, over the (tool, arguments) pair. Applications of MT by semantic perturbation in related domains, such as fairness evaluation in LLMs [16] and hallucination detection in retrieval-augmented generation systems [18], reinforce the generality of the principle, albeit with targets distinct from ours.

2.4 *Tool-Calling*: Reliability and *Benchmarks*

The evaluation of *tool-calling* has matured into reference *benchmarks* such as the Berkeley Function Calling Leaderboard [15], which judges tool selection and argument correctness by syntactic analysis and has since grown from single calls to multi-turn and agentic categories, and τ-bench [19], aimed at multi-turn interactions among agent, user, and tools. Such *benchmarks* measure *accuracy* against a fixed ground truth, each statement being presented in a single canonical wording; in complement, a growing literature on *robustness* stresses the agent with perturbed inputs. ReliabilityBench [10] is the work closest to ours on this axis: it evaluates agent reliability along three dimensions, consistency under repeated execution, robustness to semantically equivalent task variations, and tolerance to tool failures. We share with it the concern for consistency and perturbation, but we distinguish ourselves from it in three points: (i) our *metamorphic relations* are decided over the invocation itself, the (tool, arguments) pair, rather than over the end state of an execution; (ii) we focus specifically on the Trigger component and introduce a *language-invariance* MR (English → Portuguese), absent from the predominantly Anglophone

Table 1: Positioning against the closest related work.

Work	System under test	Transformations	Observed output	Oracle
Catalog + LLMorph [7, 8]	isolated LLM	191 NLP MRs (36 automated)	free text	MR
ARMeta [14]	REST API	LLM-generated MRs	API response	MR
Semantic invariance [9]	reasoning agent	8 semantic edits	free text	semantic similarity
BFCL [15]	LLM / agent (tool use)	–	(tool, args)	fixed ground truth (AST)
ReliabilityBench [10]	*tool-calling* agent	task variations, failures	final state	action MRs (end state)
This work	***tool-calling* agent**	**MRs over the Trigger**	**(tool, args)**	**MR + *flakiness* disc.**

benchmarks; and (iii) we statistically discount, by means of a *baseline* MR, the intrinsic *flakiness* before attributing any violation to the transformation.

2.5 Positioning

Table 1 summarizes the positioning of this work against the closest lines. The gap we occupy is perceptible: the call of Chen et al. [6] for metamorphic testing of *promptware* remains unanswered for the agentic case; MT of LLMs concentrates on textual outputs of isolated models [7, 8]; when it reaches agents, the output observed is still free text judged by similarity [9]; and the robustness evaluation of agents, in the works we surveyed, does not isolate the Trigger layer, deciding equivalence over the end state of an execution [10]. To the best of our analysis of the literature, this work is the first to apply metamorphic testing to the Trigger component of *tool-calling* agents with an objective comparator based on the (tool, arguments) pair and explicit discounting of nondeterminism.

3 Approach

This section presents the conceptual contribution of the work: a formalization of the *tool-calling* agent as a function under test (Section 3.1), a catalog of metamorphic relations over the *Trigger* component (Section 3.2), and an objective comparator that defines, without an absolute oracle, when a relation is violated (Section 3.3). Figure 1 summarizes the resulting flow.

3.1 Formalization

Let A be an FM-based agent, equipped with a fixed inventory of tools $\mathcal{T} = \{\tau_1, \ldots, \tau_m\}$, each with a name and a parameter schema. Given a *Trigger* t, the natural-language statement that expresses the user's intent, the agent produces an invocation

$$A(t, \mathcal{T}) = (\tau, \mathbf{a}),$$

where $\tau \in \mathcal{T} \cup \{\perp\}$ is the selected tool ($\perp$ denotes a refusal or absence of a call) and $\mathbf{a}$ is the argument map assigned to the parameters of τ. Since A is nondeterministic even under a fixed configuration [3, 20], we treat $A(t, \mathcal{T})$ as a random variable and observe it through n independent runs.

A *metamorphic relation* (MR) is a pair $(f, \simeq)$ in which f is a Trigger transformation that *preserves the intent* and $\simeq$ is an equivalence relation over invocations. The MR stipulates the property

$$A(f(t), \mathcal{T}) \;\simeq\; A(t, \mathcal{T}),$$

that is, transforming the Trigger so as to preserve its meaning should not alter the agent's observable behavior. When the property does not hold, we record a *metamorphic violation*, interpreted as a *silent behavioral regression*: a defect that manifests without any output, in isolation, being demonstrably "wrong."

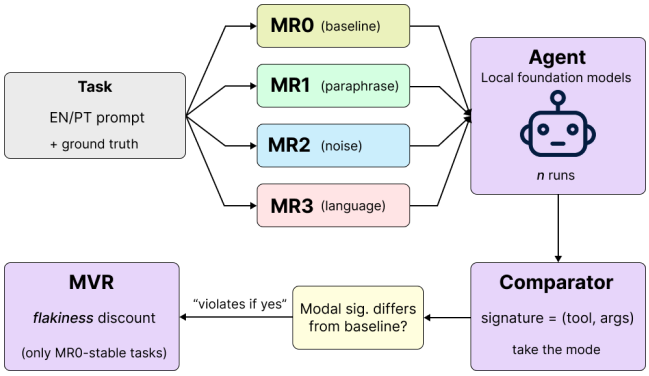

Figure 1: Methodological pipeline: from each task the four metamorphic variations are derived, run n times by the agent; the objective comparator over (tool, essential arguments) reduces each case to its modal signature $\hat{\sigma}$ and decides a violation *against the* baseline MR0, discounting the intrinsic *flakiness* in the MVR computation.

3.2 Catalog of Metamorphic Relations Over the Trigger

We instantiate four metamorphic relations that cover complementary and orthogonal dimensions of surface-level Trigger variation, all chosen because they preserve the intent and because they reflect perturbations that occur naturally in real use: the identity (MR0), which controls for intrinsic *flakiness*; manual paraphrase (MR1), for semantic equivalence; deterministic typing noise (MR2), for robustness to surface perturbation; and EN → PT translation (MR3), for language invariance. Figure 2 shows the four applied to a single Trigger.

MR0: baseline (control). The identity transformation, $f_0(t) = t$. By construction, $A(f_0(t), \mathcal{T}) \simeq A(t, \mathcal{T})$ should hold trivially; any divergence among the n runs of t measures the *intrinsic nondeterminism* (*flakiness*) of the agent, not an effect of a transformation. MR0 is the methodological instrument that separates genuine sensitivity from noise, a distinction central to the debate over whether prompt sensitivity is a defect or an artifact [13]. It answers the precise obstacle that Chen et al. [6] raise (Sec. 4.4) when calling for

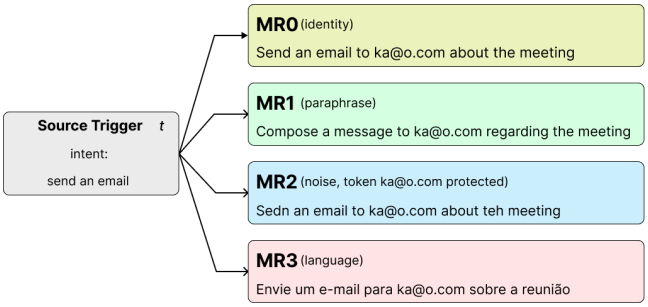

Figure 2: The four metamorphic relations applied to a single source Trigger. All preserve the intent (send an email to `ka@o.com`); MR2 injects typing noise into the natural language but *shields* the critical token that carries the essential argument, and MR3 translates the statement keeping the same recipient. A robust agent should produce the same invocation in all four.

metamorphic testing of prompts: since "LLMs often produce different outputs for the same prompt, it is difficult to assess whether the variation is due to prompt inconsistencies or model behavior."

MR1, paraphrase (semantic equivalence). $f_1(t)$ rewrites t preserving the intent, altering vocabulary and syntactic structure, for example:

"Send an email to ka@o.com"
↓
"Compose a message addressed to ka@o.com"

Since sensitivity to paraphrase is documented even in robust models [1, 17], we write and review the paraphrases *manually*, one per task: this way, an eventual change of behavior cannot be attributed to a drift of meaning introduced by the transformation.

MR2, surface noise (robustness). $f_2(t)$ injects realistic typing errors (duplication and transposition of adjacent letters, double spaces), without altering the language or applying global case changes, so as not to contaminate the effect of MR3. The transformation is *deterministic* (fixed seed), guaranteeing reproducibility, and *protects critical tokens*: any *token* containing an e-mail address or a digit (addresses, amounts, dates, times) is shielded against corruption. Without this protection, the task would become unfair: the failure would be that of the noise generator, not of the agent. Purely alphabetic arguments, such as a city or an event title, remain exposed, which is what MR2 is meant to stress.

MR3, language invariance. $f_3(t)$ replaces the Trigger with its equivalent translation into Portuguese, keeping the same intent and the same essential arguments. A robust agent should select the same tool with the same arguments regardless of the language of the statement. This relation is, at the same time, original relative to the predominantly Anglophone *benchmarks* [10, 15] and particularly pertinent to the Luso-Brazilian context of the event.

3.3 Comparator and Definition of Violation

The equivalence relation $\simeq$ needs an *objective and cheap* instantiation, one that dispenses with the costly and noisy use of an LLM as a judge. We define it over the *behavior signature* of an invocation, rather than over the free text eventually generated.

Behavior signature. For a task whose ground truth specifies the essential arguments K (the keys that matter for judging the intent; fields the statement leaves open, such as `subject` or `body`, are ignored), the signature of $(\tau, \mathbf{a})$ is

$$\sigma(\tau, \mathbf{a}) = \big(\,\mathrm{norm}(\tau),\ \{\, k \mapsto \mathrm{norm}(\mathbf{a}[k]) : k \in K \,\}\,\big),$$

where $\mathrm{norm}(\cdot)$ applies light normalization: *lowercasing*, whitespace removal, and numeric coercion, so that `"100"` and `100` are treated as equal and type differences across models do not generate false violations. Two invocations are equivalent, $(\tau, \mathbf{a}) \simeq (\tau', \mathbf{a}')$, if and only if $\sigma(\tau, \mathbf{a}) = \sigma(\tau', \mathbf{a}')$. Capturing the signature over *(tool, essential arguments)*, and not just over the tool, is what allows detecting subtle violations in which the agent gets the tool right but corrupts or alters the argument; Figure 3 illustrates a concrete case.

Modal signature and violation. Since each Trigger is run n times, we summarize the n signatures of a case (A, t, MR) by their *mode* $\hat{\sigma}$ (the most frequent signature). For an MR $X \in \{1, 2, 3\}$ and a task t, there is a violation when

$$\hat{\sigma}\big(A, f_X(t)\big) \ \neq\ \hat{\sigma}\big(A, f_0(t)\big).$$

Note that the comparison is made against the modal MR0 signature of the *model itself*, and not against the ground truth: we measure *intra-version consistency*, faithful to the spirit of metamorphic testing.

Flakiness discount and MVR.. So as not to confuse nondeterminism with the effect of the transformation, a task is considered *evaluable* only if it is stable under MR0, that is, if its n runs of $f_0(t)$ produce a single distinct signature. The *metamorphic violation rate* (MVR) of a model under an MR X is then the fraction of evaluable tasks in which X is violated:

$$\mathrm{MVR}_X = \frac{\big|\{\, t \in \mathcal{S} : \hat{\sigma}(A, f_X(t)) \neq \hat{\sigma}(A, f_0(t)) \,\}\big|}{|\mathcal{S}|},$$

$$\mathcal{S} = \{\, t : t \text{ stable under MR0} \,\}.$$

This discount materializes, at the level of the metric, the separation between genuine regression and *flakiness* that motivates MR0, and directly responds to the concern that prompt sensitivity might be an evaluation artifact [13].

Severity of a violation. Treating every divergence of an essential argument as a violation measures *consistency*, but does not decide whether the divergence is a *defect*: translating a search *query* preserves the function of the call, whereas translating a city name that the tool resolves against a geocoder may not. We therefore classify each argument, from the schema of its tool, as *free-text*, consumed as natural-language payload (`query`, `title`, `task`, `text`), or *resolved*, matched against an external vocabulary or identifier (`city`, `target_language`, `date`, `time`, `ticker`, currency codes, recipient). A violation is *acceptable* (S0) when the meaning is preserved in a *free-text* argument, *tool-dependent* (S1) when it is preserved but falls on a *resolved* one, and a *functional defect* (S2) when the argument is corrupted or semantically wrong, or the tool changes. Alongside the strict MVR we therefore report the *defect* MVR, which counts only S1 and S2.

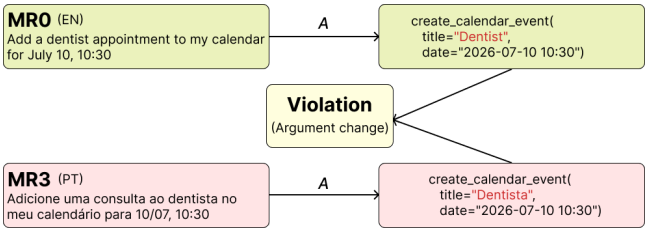

Figure 3: Example of a *silent behavioral regression*. Translating the Trigger from English (MR0) to Portuguese (MR3) preserves the intent and the selected tool, but alters the essential argument (`title`: `"Dentist"` → `"Dentista"`). The call remains syntactically valid and no error is raised; nonetheless, the modal signature diverges from that of the *baseline*, recording a violation.

Types of divergence. Beyond the aggregate rate, we classify each divergent call into three mutually exclusive types that support the qualitative analysis: *refusal* ($\tau = \perp$), *tool switch* (wrong tool, including any tool outside the inventory), and *argument change* (right tool, divergent essential arguments). This taxonomy connects the quantitative metric to the concrete causes observed (Section 5).

4 Study Design

To evaluate the approach of Section 3, we conducted a controlled empirical study over *tool-calling* agents run locally. This section describes the research questions that guide the investigation, the system under test, the models and the task set, the execution protocol, and the analysis metrics, closing with the measures adopted in favor of reproducibility.

4.1 Research Questions

The study is organized around four questions that progress from the prevalence of the phenomenon to its characterization and to the factors that modulate it:

- **RQ1 (prevalence).** How frequently do the agents violate the proposed metamorphic relations? This question establishes whether the fragility of the Trigger is a marginal or a systematic problem.
- **RQ2 (nature).** Which transformations most induce violation and what types of inconsistency (tool switch, argument change, refusal, or hallucination) predominate? Here we seek not only to quantify, but to *characterize* the failures.
- **RQ3 (nondeterminism).** What fraction of the violations corresponds to genuine sensitivity to the transformation, as opposed to the FM's intrinsic nondeterminism? This question methodologically validates the *flakiness* discount.
- **RQ4 (discrimination).** Does the violation rate differ between two distinct agent configurations, that is, does metamorphic testing serve as a discriminative instrument among agents? The question is exploratory, since the two configurations evaluated differ in more than one factor at a time.

4.2 System Under Test

The system under test is a single-turn *tool-calling* agent, in the configuration most widespread in production *frameworks*: the model receives a fixed *system prompt*, the user's Trigger, and the tool inventory $\mathcal{T}$ in the native *function calling* parameter, and must respond with exactly one invocation. To isolate the effect of the Trigger, the *system prompt* is kept identical across all conditions, reproduced here in full:

> *"You are a helpful assistant with access to tools. Use exactly one tool to fulfill the user's request."*

The tools are provided structurally (via a function schema), and not in the body of the text, so that the only variable manipulated across the experimental conditions is the metamorphic transformation applied to the Trigger. We opted for the single-turn scenario, in contrast to multi-turn *benchmarks* such as τ-bench [19], because it isolates the agent's elementary decision (selection and parameterization of a tool) without the confounding noise introduced by long trajectories, aligning the unit of measurement with the granularity of the metamorphic relation.

4.3 Models

We evaluate two open models of distinct sizes and families, both with native support for *tool-calling*: `qwen3.5:4b` (4 billion parameters) and `llama3.2:3b` (3 billion). These are two specific configurations, not representatives of their families: they differ at once in family, in number of parameters, and in inference characteristics, so any difference between them is the joint effect of these factors. The choice favors small models for two reasons. First, smaller models are more prone to violating metamorphic relations, which enriches the observable signal. Second, and more importantly, open models run locally eliminate the documented instability of commercial APIs and guarantee the study's reproducibility [2]. The models were served via Ollama on consumer *hardware* (a GPU with 4 GB of VRAM), demonstrating that the protocol is feasible without specialized infrastructure.

4.4 Task Set

We built our own controlled set of **40 tasks** over a fixed inventory of **8 plausible and mutually distinguishable tools** (Table 2): weather lookup, email sending, web search, calendar event creation, currency conversion, text translation, stock quote, and reminder setting. Each tool is exercised by five tasks, balancing coverage of the inventory. We opted for our own *dataset*, rather than reusing an existing *benchmark*, to precisely control the ground truth and the essential arguments of each task; even so, we follow the (tool, arguments) evaluation protocol established by the Berkeley Function Calling Leaderboard [15]. It is precisely these essential arguments, and not the optional fields, that the comparator of Section 3.3 inspects when deciding a violation.

Each task specifies a statement in English and its counterpart in Portuguese, both required for the language relation (MR3), the expected tool, and the set of essential arguments that constitute the ground truth. The metamorphic variations are generated *a single time* by a deterministic *script*, which applies the surface noise with a fixed seed and combines the manual paraphrases, and are then *frozen* into an input file (`variations.jsonl`). This freezing ensures

Table 2: Tool inventory and respective essential arguments (K), which make up the behavior signature evaluated by the comparator.

Tool	Essential arguments
Weather lookup	`city`
Email sending	`to`
Web search	`query`
Calendar event	`title`, `date`
Currency conversion	`amount`, `from_currency`, `to_currency`
Text translation	`text`, `target_language`
Stock quote	`ticker`
Reminder setting	`task`, `time`

that all models are confronted with exactly the same stimuli, and that the experiment is fully re-executable from the same input.

4.5 Execution Procedure

The experiment traverses the Cartesian product of models, tasks, metamorphic relations, and repetitions. Each Trigger is submitted to the agent with temperature 0.7 and a sampling seed equal to the repetition index, and each case is run $n = 5$ times to sample the model's nondeterminism. The complete configuration, 2 models $\times$ 40 tasks $\times$ 4 relations $\times$ 5 repetitions, totals 1,600 invocations. For each call, we record the raw output in full: the selected tool, the arguments, any free text, the latency, and any error, so that all subsequent analysis operates over the persisted *log*, and never over the volatile state of a run. The choice of a nonzero temperature is intentional: lowering it would not yield determinism [3, 20], only mask the *flakiness* we intend to measure.

4.6 Metrics and Analysis

The primary metric is the *metamorphic violation rate* (MVR), defined in Section 3.3, reported per model and per relation to answer RQ1 and RQ2. In complement, we measure the *flakiness* under the *baseline* MR0, the average number of distinct signatures across the n repetitions, and the proportion of unstable tasks, as a direct answer to RQ3 and as the basis of the discount applied to the MVR. The nature of the failures (RQ2) is examined through the distribution of the four types of inconsistency and through a *qualitative inspection* of all violations, in which each case is manually labeled by its concrete cause. We also report, as secondary measures, the accuracy against the ground truth and the latency per model, which contextualize the *trade-off* between robustness and cost.

For RQ4, we compare the robustness of the two models by means of the Wilcoxon test paired by task, applied to the mean MVR of the tasks evaluable in both models, reporting the p-value. The paired and nonparametric test is appropriate because the observations are paired by task and the distribution of the violation rates does not satisfy normality assumptions. We adopt $\alpha = 0.05$ as the significance level.

4.7 Reproducibility

All controllable sources of variation were fixed (the sampling seeds, equal to the repetition indices, the seed of the noise generator, the frozen variation file, and the exact model versions, recorded in the artifact), and the use of local open models, instead of commercial APIs, removes one of the documented obstacles to replicating empirical studies with FMs [2], whose analysis also identifies incomplete and non-executable research artefacts as a dominant impediment. We release a complete artifact package (the tool inventory, the tasks and the ground truth, the paraphrases, the variation generator, the execution *harness*, the raw *log* of the 1,600 calls, and the analysis *scripts*), sufficient to fully reproduce the results of the next sections.

Table 3: MVR (%) per model and relation, over the MR0-stable tasks: strict (defect MVR in parentheses). Wilson 95% CIs for MR3: [24.5, 59.3] and [15.3, 47.1]; the artifact reports every interval.

Model	MR1	MR2	MR3
`llama3.2:3b`	7.4 (0.0)	7.4 (7.4)	**40.7** (18.5)
`qwen3.5:4b`	10.7 (0.0)	3.6 (3.6)	**28.6** (3.6)

5 Results

This section reports the findings of the 1,600 invocations, organized by the four research questions. After the *flakiness* discount (Section 3.3), 28 of the 40 tasks are evaluable for `qwen3.5:4b` and 27 of the 40 for `llama3.2:3b`; unless otherwise indicated, the violation rates are computed over this stable subset. All figures come from the raw *log* and are reproducible from the artifact package.

5.1 RQ1, Prevalence of Violations

Table 3 presents the metamorphic violation rate (MVR) per model and relation, and Figure 4 visualizes it. The result is unequivocal and symmetric across the two models: violations exist in all relations, but they are *strongly concentrated* in language invariance. MR3 reaches 40.7% of the tasks on `llama3.2:3b` and 28.6% on `qwen3.5:4b`, whereas paraphrase (MR1) and noise (MR2) remain below 11% on both models. The language of the Trigger alters the agent's behavior far more often than paraphrase or noise, and this *despite* the intent, the expected tool, and the essential arguments remaining identical.

Under the severity classification of Section 3.3, the defect MVR of MR1 is zero on both models, since every paraphrase violation is a semantically equivalent rewrite of a *free-text* argument, whereas MR2 is unchanged, as all three of its violations corrupt the argument and are therefore S2. The ordering among the relations thus survives the stricter criterion: the strict MVR is an upper bound, the defect MVR its conservative counterpart.

Answer to RQ1. Metamorphic violations are prevalent and systematic, not marginal: the Trigger component is fragile in a measurable way, and this fragility concentrates in the language dimension, which alone violates between 29% and 41% of the stable tasks.

5.2 RQ2, Nature and Types of Inconsistency

To characterize *how* the agents fail, Table 4 distributes the 600 invocations under transformation (MR1–MR3) of each model by how the call departs from the ground truth. The pattern is revealing: the dominant failure is not the tool switch, but the *argument change*. `llama3.2:3b` alters arguments in 200 calls and switches tools in only 5; `qwen3.5:4b` alters arguments in 176 calls and *never* switches

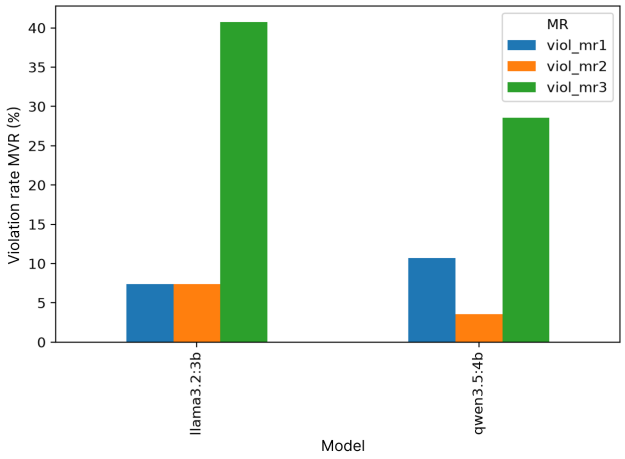

Figure 4: Metamorphic violation rate per model and relation. Language invariance (MR3) stands out in both models, well above paraphrase and noise.

Table 4: Departures from the ground truth in the calls under transformation (MR1–MR3).

Model	Arg. change	Tool switch	Refusal	OK
`llama3.2:3b`	200	5	28	367
`qwen3.5:4b`	176	0	18	406

tools. In other words, the agent almost always correctly identifies *which* tool to use, but fails at *how* to parameterize it, a shift of the problem from selection to argument fidelity that only becomes visible because the comparator observes the (tool, arguments) pair, and not just the tool.

The qualitative inspection of the 27 real violations deepens the diagnosis. Labeling each violation by its concrete cause (Table 5), a central finding emerges: *each metamorphic relation reveals a distinct failure mode*. Paraphrase (MR1) induces exclusively rewrites of the *query*, semantically equivalent but literally distinct:

`"tallest mountain in the world"`
↓
`"highest mountain on Earth"`

Noise (MR2) leaks from the surroundings into the essential argument itself, despite the protection of critical *tokens*:

`"São Paulo"`
↓
`"Soã Paulo"`

The language relation (MR3) translates the content of the argument:

`"Dentist"`
↓
`"Dentista"`

but in two cases it *corrupts* it during translation, and both fall on the `target_language` of the translation tool, which resolves the value against a fixed vocabulary:

`target_language="French"`
↓
`"francç"`

Table 5: Causes of the 27 real violations (qualitative inspection), with the severity of Section 3.3. Each cause concentrates in a single relation.

Cause	n	Sev.	Relation
Argument translation, *free-text*	13	S0	MR3
Paraphrase altered the *query*	5	S0	MR1
Argument translation, *resolved*	4	S1	MR3
Corruption by noise	3	S2	MR2
Corruption during translation	2	S2	MR3

Answer to RQ2. The inconsistency manifests chiefly in the arguments passed to the tool, and not in the selected tool; the language relation is simultaneously the most frequent and the most severe, spanning from defensible argument translations to outright character corruption.

5.3 RQ3, Genuine Sensitivity *Versus* Nondeterminism

The intrinsic nondeterminism measured under the *baseline* MR0 is substantial: about **30%** of the tasks exhibit more than one behavior signature across the five repetitions of the *identical* input (32.5% for `llama3.2:3b`, 30.0% for `qwen3.5:4b`), with an average of 1.40 and 1.33 distinct signatures per task, respectively. This finding is doubly important. First, it empirically justifies the discount adopted: without excluding the MR0-unstable tasks, the MVR would incorporate variations that are noise of the model itself, overestimating the effect of the transformations, precisely the confusion between defect and artifact pointed out in the literature [13]. Second, since the two models exhibit similar *flakiness*, the difference in robustness observed in RQ1 and RQ4 is not an artifact of one model being intrinsically more unstable than the other, but a real effect of the metamorphic relations.

Answer to RQ3. There is expressive intrinsic nondeterminism (30% of the tasks), which makes the *flakiness* discount methodologically indispensable; once discounted, the remaining violations of MR1–MR3 constitute genuine sensitivity to the transformation, and not FM noise.

5.4 RQ4, Discrimination Between Configurations

The direct comparison between the two configurations shows that metamorphic testing has the power to discriminate between them. The Wilcoxon test paired by task, applied to the mean MVR of the 23 tasks evaluable in both models, is statistically significant ($p = 0.025$), with a mean MVR of 0.174 for `llama3.2:3b` against 0.101 for `qwen3.5:4b`, about 72% more violations. Because the two configurations differ at once in family, size, and inference characteristics, and because the test rests on 23 pairs, this comparison is exploratory: it establishes a difference between the pair evaluated, not the effect of family, generation, or scale, which would require a factorial design. The null test statistic indicates that all nonzero

paired differences point in the same direction, reinforcing the consistency of the effect. This robustness picture is coherent with the secondary measures of Table 6: `qwen3.5:4b` is more accurate across all transformations (overall accuracy of 71.4% against 66.6%) and never switches tools, at the cost of a median latency about eight times higher. It is also noteworthy that the accuracy of both models collapses under the language relation, from approximately 83% under the *baseline* to 45–53% under MR3, mirroring, on the plane of absolute correctness, the fragility that the MVR captures on the plane of consistency.

Table 6: Secondary measures: accuracy (%) per relation and median latency per call.

Model	MR0	MR1	MR2	MR3	Med. lat.
`llama3.2:3b`	83.0	68.5	70.0	45.0	1.1 s
`qwen3.5:4b`	82.5	73.0	77.0	53.0	8.7 s

Answer to RQ4. Metamorphic robustness differs significantly between the two configurations evaluated ($p = 0.025$): `qwen3.5:4b` is more consistent and more accurate than `llama3.2:3b`, although substantially slower. Metamorphic testing thus works as a discriminative instrument of quality among agents, with the caveat that this exploratory comparison does not isolate the contribution of any single factor.

6 Discussion and Guidelines

The results of Section 5 converge on a single thesis: the Trigger component of *tool-calling* agents, although it governs the entire execution trajectory, carries a measurable fragility that escapes accuracy-centered evaluation protocols. This section interprets these findings from the standpoint of those who build and operate agents, and derives actionable guidelines for incorporating metamorphic verification into the development cycle.

6.1 Silent Behavioral Regression Is a First-Order Risk

The most consequential finding is not that agents err, but that they err *silently*. In up to 41% of the stable tasks, a transformation that preserves the intent (translating the user's request from English to Portuguese) alters the agent's behavior without producing any failure signal: no exception, no *timeout*, no explicit refusal. The call is syntactically well-formed and the correct tool is selected; only the argument changes. For a production *pipeline* that monitors error and exception rates, this class of defect is invisible. This result materializes, on the agentic plane, the debate over whether prompt sensitivity is an evaluation artifact or a behavioral defect [1, 13], here decided by an exact comparator rather than by heuristic answer matching, and it extends the already acknowledged fragility of *tool-calling* [15] into a dimension, that of the language of the request, which the predominantly Anglophone *benchmarks* do not exercise.

The fact that the inconsistency concentrates in the *arguments*, and not in tool selection, reinforces this diagnosis. An agent that chose the wrong tool would fail in a detectable way downstream; one that chooses the right tool but parameterizes it inconsistently executes a plausible, and possibly incorrect, action without alarm, and it is this plausibility that makes the silent behavioral regression dangerous in systems that act upon the world.

6.2 Language as a Neglected Robustness Axis

The severity of the language relation (MR3) relative to paraphrase (MR1) and noise (MR2) suggests that the *multilingual* robustness of agents is a distinct quality axis, and not a corollary of robustness to surface perturbations. Small models seem to treat the translation of the Trigger as a genuine semantic change, propagating it to the arguments, sometimes defensibly, sometimes in a clearly defective way. For the Brazilian software-testing community, in which real applications operate mostly in Portuguese over models trained chiefly in English, this finding has a direct practical implication: evaluating an agent only in English overestimates its reliability in the language in which it will actually be used. The language MR offers a cheap and automatable instrument for exposing this gap.

6.3 Metamorphic Testing as a Selection Instrument and Regression *Gate*

The statistical significance of the difference between the two configurations (Section 5.4) demonstrates that the MVR is not merely descriptive, but *discriminative*: it orders agents by robustness in a way that accuracy alone does not capture, even though ordering two configurations does not establish which factor produces the difference. Notably, the most accurate model (`qwen3.5:4b`) is also the most consistent, but at the cost of a latency about eight times higher, a *trade-off* between robustness and cost that only becomes visible when consistency and accuracy are reported together. This positions metamorphic testing as a complement, and not a substitute, to accuracy *benchmarks* [15]: while the latter measure whether the agent matches the ground truth, metamorphic verification measures whether it behaves *stably* under variations that should be innocuous, an independent property, and indispensable for trusting an agent in production.

6.4 Actionable Guidelines

From the findings we derive four guidelines for builders of *frameworks* and agentic applications:

- **D1: Treat the Trigger as a testable artifact.** The inversion of testing effort documented by Hasan et al. [12], with the Trigger in ∼1% of the test cases, should be corrected by elevating the prompt to the condition of a first-class software artifact [6]. Metamorphic relations offer a concrete path: they generate test cases automatically from existing tasks, without requiring an absolute ground truth.
- **D2: Incorporate MRs as a regression *gate* in CI.** Being cheap and dispensing with an exact oracle, MR1 (paraphrase) and MR3 (language) can be run at every update of the model, *prompt*, or *framework*, flagging silent behavioral regressions before the *deploy*. The MVR works as a *gate* metric analogous to coverage in traditional testing.
- **D3: Report consistency alongside accuracy.** Accuracy against a fixed ground truth is insufficient to characterize an agent. We recommend reporting consistency under transformation

(MVR) and *baseline flakiness* as first-class metrics, complementing the accuracy *leaderboards*.

D4: Evaluate in the production language. Given the severity of MR3, agents intended to operate in Portuguese (or in any language distinct from the predominant training one) must be explicitly tested in that language. Language invariance cannot be presumed from performance in English.

7 Threats to Validity

We discuss the threats to validity according to the four usual categories (construct, internal, external, and conclusion), with particular attention to the most delicate one of our design: the interpretation of the violations of the language relation.

7.1 Construct Validity

The most important construct threat concerns what a violation of MR3 effectively means. The qualitative inspection (Section 5.2) revealed that 17 of the 27 real violations correspond to *argument translation*: upon receiving the request in Portuguese, the agent also translates the *value* passed to the tool, for example, from `"Dentist"` to `"Dentista"`. Strictly, this violates the language invariance that MR3 postulates; but for the 13 that fall on *free-text* arguments (S0) it is a *defensible*, and possibly desirable, behavior for a request formulated in Portuguese, and only the remaining 4, on arguments the tool resolves externally (S1), may break the call. We therefore acknowledge that not every MR3 violation is a defect: the method captures the *behavioral inconsistency* under a transformation that preserves the intent, the phenomenon of interest, and it is up to the evaluator to qualify which inconsistencies constitute a genuine regression.

So that this nuance does not inflate our conclusions, we treat it as a measurement decision, reporting every rate in the strict and the defect readings. Of the 19 violations attributed to MR3, 2 are unequivocally defective, both of *corruption during translation* of a *resolved* argument:

```
target_language="French"
        ↓
    "francç"
```

The classification also disciplines the qualitative inspection: a divergence in a field outside the essential arguments, or in one that already varies under MR0, must not be attributed to the transformation, exactly the confusion that the *baseline* exists to prevent. Even discarding the defensible translations entirely, the language relation remains the most severe of the catalog, with unequivocal defects absent from paraphrase and noise; and the central qualitative finding, *each metamorphic relation reveals a distinct failure mode*, is robust to this reclassification.

A second construct threat is the comparator itself, which defines equivalence by the (tool, normalized essential arguments) pair and may, in principle, ignore textual nuances that a human judge would capture. We deliberately opted for this objective oracle, in preference to an *LLM-as-judge*, for three reasons: it is deterministic and reproducible, it is computationally cheap, and it does not introduce the judge model's own inconsistency into the measurement, a problem recognized in the LLM-evaluation literature [1]. The normalization of essential arguments, aligned with the BFCL protocol [15], mitigates false positives from cosmetic formatting differences.

7.2 Internal Validity

The main internal threat is confusing the model's intrinsic nondeterminism with a violation induced by the transformation. Even under a nominally fixed configuration, foundation models exhibit inference nondeterminism due to numerical and *batching* causes [3, 20]. We mitigate this threat by design: the *baseline* relation MR0 measures the *flakiness* of each task under repetition of the identical input, and we discount from the MVR all MR0-unstable tasks, approximately 30% of them (Section 5.3). The remaining violations, computed over the stable subset, reflect genuine sensitivity to the transformation. The execution with $n = 5$ repetitions and the aggregation by modal signature further reduce the influence of stochastic *outliers*.

7.3 External Validity

Our findings derive from two small open models, 40 tasks, and our own single-turn *dataset*, which limits generalization. Larger, closed, or multi-turn models may exhibit distinct robustness patterns. Three considerations attenuate this limitation. First, the choice of small models is methodologically intentional: they maximize the observable signal and establish a *lower bound* of robustness that larger models tend to exceed, without invalidating the existence of the phenomenon. Second, the controlled *dataset*, with an explicit ground truth in Portuguese and English, was necessary to precisely isolate the essential arguments and the language relation, absent from Anglophone *benchmarks* [15, 19]. Third, and decisive for replicability, the use of open models run locally removes the documented instability of commercial APIs in empirical studies [2]. The metamorphic approach itself is scale-independent and directly applicable to larger models and to multi-turn scenarios, which we leave as future work.

7.4 Conclusion Validity

The statistical inferences rest on the Wilcoxon paired test, chosen for being nonparametric and appropriate for observations paired by task whose distribution does not satisfy normality assumptions. We fixed all controllable sources of variation, the model's sampling seeds and the seed of the noise generator, and we froze the file of metamorphic variations, so that the experiment is fully reexecutable. We acknowledge that, in evaluating multiple metamorphic relations, successive comparisons could require correction for multiple testing; we restrict our significance claim to the main comparison between models (RQ4), reporting the exact p-value ($p = 0.025$) so that the result can be evaluated in its own context. The sample size per pair (23 tasks evaluable in both models) is modest, which recommends caution in extrapolating the magnitude of the effect, even though its direction is consistent.

8 Conclusion and Future Work

This work started from an asymmetry documented in the testing practice of agents based on foundation models: the Trigger component, which governs the entire execution trajectory, figures in about 1% of the test cases [12]. To attack this gap, we proposed

an approach for metamorphic testing of the Trigger of *tool-calling* agents, articulating a catalog of metamorphic relations (*baseline*, paraphrase, noise, and language) with an objective comparator based on the (tool, essential arguments) pair and with an explicit discount of nondeterminism. The approach offers a pseudo-oracle for a scenario in which the absolute oracle is infeasible, without requiring a ground truth beyond the tasks themselves.

The empirical study, with 1,600 invocations over two open models, supports four conclusions: violations are prevalent and concentrate in the language relation (up to 41% of the stable tasks, against under 11% in the others); the inconsistency falls chiefly on the *arguments*, constituting a *silent behavioral regression* invisible to conventional error monitors; a substantial intrinsic *flakiness* (∼30%) makes the nondeterminism discount indispensable, and under it the violations persist as a genuine effect; and the test discriminates models significantly ($p = 0.025$), complementing accuracy *benchmarks*.

These findings position metamorphic verification as a cheap regression *gate* for the agent development cycle, and highlight multilingual robustness as a neglected quality axis for applications running in Portuguese over models trained mostly in English.

Several directions extend this work. On the plane of the catalog, new metamorphic relations (instruction reordering, distractor insertion, and formality variations) would broaden the coverage of the Trigger's failure modes. On the plane of the system under test, the approach is directly applicable to larger and closed models, and to multi-turn scenarios such as τ-bench [19], in which the fragility of the Trigger may propagate along entire trajectories. On the plane of the comparator, investigating hybrid oracles that combine the objective equivalence by (tool, arguments) with a controlled semantic judgment would allow automatically qualifying which violations constitute a defect, mitigating the ambiguity we identified in argument translations. We release the complete artifact package, including the tasks, the variation generator, the *harness*, and the analysis *scripts*, so as to support the replication and extension of all the results reported here.

Artifact Availability

To ensure reproducibility, we release a complete artifact package containing the tool inventory, the 40 tasks with ground truth in Portuguese and English, the manual paraphrases, the deterministic generator of metamorphic variations, the execution *harness*, the raw *log* of the 1,600 invocations, and the analysis *scripts* that produce all the tables and figures of this paper, together with a `README` documenting the requirements and the installation and execution steps.

Version 1.1.0 of the artifact is permanently archived on Zenodo at https://doi.org/10.5281/zenodo.22130934. The development repository, which may evolve, is mirrored at https://github.com/Wyllgner/When-the-Trigger-Fails.

The artifact is released under the MIT License, which permits use, modification, and redistribution, including for commercial purposes, provided the copyright notice is retained. The two models exercised in the study are publicly available open-weight models distributed through Ollama and remain subject to their own respective licences. The artifact contains no personal, sensitive, or human-subject data.

References

[1] Jihyun Janice Ahn and Wenpeng Yin. 2025. Prompt-Reverse Inconsistency: LLM Self-Inconsistency Beyond Generative Randomness and Prompt Paraphrasing. arXiv:2504.01282 [cs.CL] https://arxiv.org/abs/2504.01282

[2] Florian Angermeir, Maximilian Amougou, Mark Kreitz, Andreas Bauer, Matthias Linhuber, Davide Fucci, Fabiola Moyón C., Daniel Mendez, and Tony Gorschek. 2025. Reflections on the Reproducibility of Commercial LLM Performance in Empirical Software Engineering Studies. arXiv:2510.25506 [cs.SE] https://arxiv.org/abs/2510.25506

[3] Berk Atil, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. 2025. Non-Determinism of "Deterministic" LLM Settings. arXiv:2408.04667 [cs.CL] https://arxiv.org/abs/2408.04667

[4] T. Y. Chen, S. C. Cheung, and S. M. Yiu. 2020. Metamorphic Testing: A New Approach for Generating Next Test Cases. arXiv:2002.12543 [cs.SE] https://arxiv.org/abs/2002.12543

[5] Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, T. H. Tse, and Zhi Quan Zhou. 2018. Metamorphic Testing: A Review of Challenges and Opportunities. *ACM Comput. Surv.* 51, 1, Article 4 (Jan. 2018), 27 pages. doi:10.1145/3143561

[6] Zhenpeng Chen, Chong Wang, Weisong Sun, Xuanzhe Liu, Jie M. Zhang, and Yang Liu. 2026. Promptware Engineering: Software Engineering for Prompt-Enabled Systems. arXiv:2503.02400 [cs.SE] https://arxiv.org/abs/2503.02400

[7] Steven Cho, Stefano Ruberto, and Valerio Terragni. 2025. LLMorph: Automated Metamorphic Testing of Large Language Models. In *2025 40th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 4102–4105. doi:10.1109/ase63991.2025.00385

[8] Steven Cho, Stefano Ruberto, and Valerio Terragni. 2025. Metamorphic Testing of Large Language Models for Natural Language Processing. In *2025 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 174–186. doi:10.1109/icsme64153.2025.00025

[9] I. de Zarzà, J. de Curtò, Jordi Cabot, Pietro Manzoni, and Carlos T. Calafate. 2026. Semantic Invariance in Agentic AI. arXiv:2603.13173 [cs.AI] https://arxiv.org/abs/2603.13173

[10] Aayush Gupta. 2026. ReliabilityBench: Evaluating LLM Agent Reliability Under Production-Like Stress Conditions. arXiv:2601.06112 [cs.AI] https://arxiv.org/abs/2601.06112

[11] Hassan Hamad, Yingru Xu, Liang Zhao, Wenbo Yan, and Narendra Gyanchandani. 2025. ToolCritic: Detecting and Correcting Tool-Use Errors in Dialogue Systems. arXiv:2510.17052 [cs.AI] https://arxiv.org/abs/2510.17052

[12] Mohammed Mehedi Hasan, Hao Li, Emad Fallahzadeh, Gopi Krishnan Rajbahadur, Bram Adams, and Ahmed E. Hassan. 2026. An Empirical Study of Testing Practices in Open Source AI Agent Frameworks and Agentic Applications. arXiv:2509.19185 [cs.SE] https://arxiv.org/abs/2509.19185

[13] Andong Hua, Kenan Tang, Chenhe Gu, Jindong Gu, Eric Wong, and Yao Qin. 2025. Flaw or Artifact? Rethinking Prompt Sensitivity in Evaluating LLMs. arXiv:2509.01790 [cs.CL] https://arxiv.org/abs/2509.01790

[14] Shehroz Khan, Abdullah Mughees, Gaadha Sudheerbabu, Tanwir Ahmad, and Dragos Truscan. 2026. Multi-Agent LLM-based Metamorphic Testing for REST APIs. arXiv:2605.28321 [cs.SE] https://arxiv.org/abs/2605.28321

[15] Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. The Berkeley Function Calling Leaderboard (BFCL): From Tool Use to Agentic Evaluation of Large Language Models. In *Forty-second International Conference on Machine Learning*. https://openreview.net/forum?id=2GmDdhBdDk

[16] Harishwar Reddy, Madhusudan Srinivasan, and Upulee Kanewala. 2025. Metamorphic Testing for Fairness Evaluation in Large Language Models: Identifying Intersectional Bias in LLaMA and GPT. arXiv:2504.07982 [cs.CL] https://arxiv.org/abs/2504.07982

[17] Angelika Romanou, Mark Ibrahim, Candace Ross, Chantal Shaib, Kerem Oktar, Samuel J. Bell, Anaelia Ovalle, Jesse Dodge, Antoine Bosselut, Koustuv Sinha, and Adina Williams. 2026. Brittlebench: Quantifying LLM robustness via prompt sensitivity. arXiv:2603.13285 [cs.LG] https://arxiv.org/abs/2603.13285

[18] Channdeth Sok, David Luz, and Yacine Haddam. 2025. MetaRAG: Metamorphic Testing for Hallucination Detection in RAG Systems. arXiv:2509.09360 [cs.CL] https://arxiv.org/abs/2509.09360

[19] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045 [cs.AI] https://arxiv.org/abs/2406.12045

[20] Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. 2025. Understanding and Mitigating Numerical Sources of Nondeterminism in LLM Inference. arXiv:2506.09501 [cs.CL] https://arxiv.org/abs/2506.09501

[21] Zheng Zheng, Zenghui Zhou, Yinwang Xu, Daixu Ren, and Tsong Yueh Chen. 2026. Bidirectional Empowerment of Metamorphic Testing and Large Language Models: A Systematic Survey. arXiv:2605.13898 [cs.SE] https://arxiv.org/abs/2605.13898