Explanation-Guided Metamorphic Testing of Specialized Language Models: An Empirical Study

Xingcheng Chen[1]
Technical University of Munich, fortiss, Germany

Mehmet Besenk
Technical University of Munich, Germany

Andrea Stocco
Technical University of Munich, fortiss, Germany

Abstract

Background. Task-specialized language models are increasingly integrated into software engineering workflows to support vertical-domain activities such as issue triaging, document classification, and automated analysis. Despite their growing adoption, there is limited empirical evidence on how to systematically test their robustness and detect brittle decision behaviors under semantics-preserving input transformations.

Aims. This paper investigates whether explainability-guided metamorphic testing can improve the effectiveness and validity of robustness testing for specialized language models compared to heuristic mutation strategies.

Method. We conduct a large-scale empirical study of explanation-guided metamorphic testing across three datasets, four model architectures, and 20 testing configurations derived from combinations of attribution methods and mutation strategies. The evaluated configurations combine attribution-based token prioritization, LLM-driven mutation, and automated semantic verification to generate linguistically valid test variants. We assess failure discovery capability, semantic validity, and testing efficiency against heuristic baselines.

Results. Explanation-guided metamorphic testing generates 2.30× more verified failure-inducing test cases than heuristic mutation strategies. Semantic verification substantially improves mutation validity and achieves high label-preservation precision among gate-accepted variants according to human annotation. The study further reveals systematic shortcut behaviors across models, including over-reliance on named entities and formatting cues.

Conclusions. The results provide evidence that explanation-guided metamorphic testing is an effective and practical approach for empirically evaluating the robustness of task-specialized language models used in vertical AI applications.

2012 ACM Subject Classification Software and its engineering → Empirical software validation

Keywords and phrases Metamorphic testing, explainable AI, specialized language models, vertical AI, attribution-guided testing, automated test generation, empirical software engineering.

Digital Object Identifier 10.4230/LIPIcs.ESEM.2026.23

1 Introduction

Large language models (LLMs) are increasingly integrated into software engineering (SE) workflows, supporting tasks such as issue triaging, document classification, and automated analysis [25]. Beyond large proprietary foundation models, there are many task-specialized language models adapted for a specific downstream task and deployed as components of vertical AI applications. These models may be obtained by distillation, fine-tuning,

[1] Corresponding author

20th International Symposium on Empirical Software Engineering and Measurement (ESEM 2026).
Editors: Robert Feldt and Maria Paasivaara; Article No. 23; pp. 23:1–23:21
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

or lightweight adaptation of general pretrained backbones, to achieve lower latency and deployment cost while maintaining strong task-specific performance.

Despite their growing adoption, specialized language models remain highly vulnerable to shortcut learning, spurious lexical correlations, and brittle decision boundaries inherited from fine-tuning datasets. Small semantics-preserving perturbations, such as modifying entity names, formatting cues, or sentiment-bearing expressions, can therefore trigger inconsistent predictions and unreliable downstream behavior. Existing robustness testing approaches rely on black-box perturbation strategies or manually designed transformations, which often generate unrealistic or semantically invalid test cases while providing limited insight into why failures occur [4]. We therefore study robustness in settings representative of practical vertical AI deployments, where compact fine-tuned models are commonly preferred over large proprietary foundation models due to latency, privacy, and deployment constraints. Moreover, specialized language models are often deployed with full white-box access, enabling the use of attribution signals and internal explanations to identify decision-critical input regions [52]. Despite this opportunity, there is still limited empirical evidence on whether explainability-guided testing can systematically improve the generation of valid failure-inducing test cases and expose shortcut-driven robustness weaknesses in specialized language models.

Prior work has shown that language models frequently rely on brittle decision boundaries and shortcut behaviors, including superficial lexical artifacts, entity biases, and formatting cues [13, 17, 34]. Traditional evaluation pipelines mainly rely on static benchmark datasets that act as fixed test suites and primarily measure average generalization performance [46]. Such evaluations provide limited support for systematically generating corner cases capable of exposing hidden robustness weaknesses. Manual construction of challenging test cases is costly and difficult to scale [20], while existing automated mutation strategies often generate semantically invalid or unrealistic perturbations.

In this paper, we present an empirical study of explainability-guided metamorphic testing for task-specialized language models. The study investigates whether attribution-based guidance can improve the effectiveness and validity of robustness testing compared to heuristic mutation strategies. To this end, we evaluate combinations of explainable AI (XAI) attribution, LLM-driven mutation, and semantic verification mechanisms that generate targeted metamorphic variants while preserving semantic consistency. Attribution analysis identifies decision-critical tokens, a locally deployed Judge LLM produces controlled ablation and adversarial injection variants, and a semantic validation stage based on bidirectional natural language inference (BNLI) and log-perplexity shift (LPS) filters invalid mutations.

We conduct an empirical evaluation across multiple model architectures, including DistilBERT, BART-large, and Qwen-0.5B, and across diverse classification tasks, including SST-2, AG News, and GitHub Issue classification [38, 51]. Our results show that explanation-guided metamorphic testing generates substantially more verified failure-inducing test cases than heuristic mutation strategies while reducing semantically invalid perturbations. Furthermore, our analysis reveals systematic shortcut behaviors, including entity over-reliance in discriminative models and sensitivity to formatting artifacts in generative models.

2 Background

2.1 Task-Specialized Language Models

LLMs are commonly built on Transformer architectures [44] and include encoder-only models (e.g., BERT [10]), encoder–decoder models (e.g., BART and T5 [21, 32]), and decoder-only models (e.g., GPT-style models and Qwen [41]). In practical software engineering workflows,

however, deploying large proprietary models is often infeasible due to latency, cost, and privacy constraints. As a result, many SE applications rely on *task-specialized language models* obtained through distillation or supervised fine-tuning of smaller architectures [35, 41].

While these models provide improved efficiency and deployability, prior work has shown that language models frequently rely on brittle decision boundaries and shortcut behaviors, including lexical artifacts, entity biases, and formatting cues [34, 13, 17]. Such weaknesses can compromise robustness in production SE pipelines, motivating the need for systematic robustness testing approaches.

2.2 Metamorphic Testing

Metamorphic Testing (MT) addresses the *oracle problem* by validating the consistency of a system's behavior across related inputs instead of requiring a ground-truth output for every test case [6, 37]. MT defines *metamorphic relations* (MRs), which specify how outputs should remain invariant, or change predictably, under controlled input transformations.

For language models, typical MRs involve semantics-preserving transformations such as paraphrasing, synonym substitution, or contextual edits that should not alter the expected label. By generating related inputs and comparing predictions, MT can expose robustness violations and shortcut behaviors.

Recent work has explored metamorphic testing for language models through manually defined or heuristic transformations [8]. In this paper, we empirically investigate whether explainability-guided mutation and semantic verification improve the effectiveness and validity of metamorphic testing for task-specialized language models.

2.3 Explainable AI for Language Models

Explainable AI (XAI) techniques estimate the contribution of individual input features to a model's prediction [52]. In language models, this typically corresponds to assigning importance scores to input tokens. Attribution methods generally fall into three categories: (1) gradient-based methods, such as Integrated Gradients [40, 29]; (2) perturbation-based methods, such as Occlusion and LIME [49, 23, 33]; and (3) architecture-aware methods based on Transformer attention patterns [1].

Although XAI techniques are primarily used for model interpretation, attribution scores can also support robustness testing by identifying decision-critical tokens. Prior studies suggest that highly influential tokens often correspond to shortcut features or brittle lexical cues [34]. In this work, we empirically study whether attribution-guided mutations are more effective than unguided transformations at generating valid failure-inducing test cases.

3 Empirical Study

This study evaluates combinations of attribution methods, metamorphic mutation strategies, and semantic verification mechanisms for generating failure-inducing test cases.

The empirical evaluation investigates three methodological aspects of metamorphic robustness testing: (1) attribution-based mutation prioritization, (2) metamorphic mutation strategies, and (3) semantic verification, including human validation of the automatic gate.

3.1 Attribution-Based Mutation Prioritization

The study first investigates whether attribution scores can effectively guide mutation selection toward decision-critical input regions. Rather than relying on a single explanation technique,

Table 1 Illustrative examples of metamorphic mutations. Tokens with high attribution scores are underlined. MR-Ablation weakens influential lexical cues, while MR-Injection introduces adversarial context. Semantic equivalence is validated through the NLI score ($\mathcal{M}_{nli}$) and linguistic naturalness through the perplexity shift metric ($\mathcal{M}_{lps}$).

MR Type	Original Input (x)	Saliency	Mutated Variant (x')	$\mathcal{M}_{nli} \uparrow$	$\mathcal{M}_{lps} \downarrow$
Ablation	The acting was superb, but the plot was weak.	superb (0.82)	The acting was good, but the plot was weak.	0.74	0.05
Ablation	A boring and predictable movie.	boring (0.75)	A plain and predictable movie.	0.81	0.12
Injection	The movie is a masterpiece of cinema.	disaster (injected)	The movie is a masterpiece, despite being a disaster to some.	0.78	0.18
Injection	It provides a fresh perspective.	terrible (injected)	It provides a fresh, albeit terrible, perspective.	0.65	0.25

we evaluate multiple attribution strategies as interchangeable experimental factors.

Formally, given an input sequence $x = [t_1, \ldots, t_n]$, the attribution process computes a saliency vector $\mathbf{a} = [a(t_1), \ldots, a(t_n)]$, where $a(t_i)$ denotes the influence of token t_i on the model prediction.

Gradient-based Attribution. When gradient access is available, we evaluate Integrated Gradients (IG) [40], defined as:

$$a_{ig}(t_i) = (e_i - e_i^0) \int_0^1 \frac{\partial f(x^0 + \alpha(x - x^0))}{\partial e_i} d\alpha \tag{1}$$

where e_i and e_i^0 denote the embedding vectors of token t_i and its baseline representation.

Perturbation-based Attribution. We also evaluate Occlusion [49], which estimates token importance by measuring prediction changes when a token is masked:

$$a_{occ}(t_i) = f(x) - f(x \setminus \{t_i\}) \tag{2}$$

where $x \setminus \{t_i\}$ replaces token t_i with a baseline token such as `[PAD]`.

Architecture-aware Attribution. For Transformer-based models, we additionally evaluate attention-based attribution methods [1, 5], which estimate token importance using attention aggregation and gradient-weighted attention scores.

Since many language models operate on subword tokens, token-level attribution scores are aggregated at the word level, i.e., $S(w_j) = \sum_{t \in w_j} a(t)$. The resulting ranked word set $\mathcal{T}(x) = \{(w_j, S(w_j))\}$ defines the mutation priority used during metamorphic mutation.

3.2 Metamorphic Mutation Strategies

Given an input text x with label y and a specialized language model as the system under test (SUT) $f(\cdot)$, the evaluation seeks to generate metamorphic variants $\mathcal{X}' = \{x_i'\}$ such that: (i) x_i' remains semantically equivalent to x, and (ii) the model prediction changes ($f(x_i') \neq f(x)$). Such inconsistencies indicate brittle decision boundaries, shortcut behaviors, or spurious correlations. The study next evaluates different strategies for generating metamorphic variants while preserving semantic meaning. Compared to rule-based perturbations, LLM-driven mutations enable context-aware rewrites that remain linguistically natural.

Table 1 presents representative mutation examples used throughout the study. We evaluate two primary metamorphic relations (MRs).

MR-Ablation. This relation evaluates over-reliance on influential lexical features. Given an input x, the mutation targets the word $w \in \mathcal{T}(x)$ with the highest attribution score. The

selected token is either removed or replaced with a semantically weaker alternative while preserving grammaticality.

MR-Injection. This relation evaluates robustness against distracting lexical cues. Adversarial trigger words extracted from misclassified examples are inserted into the input while preserving semantic coherence and fluency.

Mutations are generated using a locally deployed Judge LLM and processed in batches to improve efficiency. Each generated variant is accompanied by a structured self-verification output indicating whether semantic and grammatical constraints are satisfied.

3.3 Semantic Verification

A major challenge in generating metamorphic variants is the risk of producing invalid test cases, i.e., mutations that alter the underlying task label and lead to false positives. To mitigate this risk and establish a reliable, yet approximated automated oracle, we implement a validation stage that filters variants based on semantic consistency and linguistic naturalness.

Bidirectional Semantic Entailment. To verify that the mutation preserves the semantic meaning of the original input x, we employ a cross-encoder Natural Language Inference (NLI) model [48, 26]. However, uni-directional entailment ($x \to x'$) is insufficient for testing: while it ensures the mutation x' does not introduce contradictory information, it does not penalize the removal of important context. Conversely, the reverse relation ($x' \to x$) verifies that the mutated input still entails the constraints expressed in the original sentence. We therefore enforce a *bidirectional* mutual entailment score:

$$\mathcal{M}_{nli}(x, x') = \min\left(P(E \mid x, x'), P(E \mid x', x)\right)$$

where $P(E)$ denotes the probability of entailment predicted by the NLI model. A mutation is accepted only if $\mathcal{M}_{nli}(x, x') > \tau_{nli}$. This bidirectional constraint reduces the likelihood that a mutation introduces semantic drift or removes essential contextual information.

Linguistic Naturalness Metric. While NLI captures semantic consistency, NLI models are often tolerant to grammatical errors or unnatural phrasing. As a result, they may assign high entailment scores to syntactically degraded inputs. If the SUT fails on such inputs, the failure may stem from out-of-distribution syntax rather than a genuine logical vulnerability.

To filter these artifacts, we evaluate linguistic fluency using perplexity [18], defined as the exponentiation of cross-entropy loss $\mathcal{L}(\cdot)$. Rather than using absolute perplexity values, we measure the *log-perplexity shift* between the original and mutated inputs:

$$\mathcal{M}_{lps}(x, x') = \log \frac{\mathrm{PPL}(x')}{\mathrm{PPL}(x)} = \mathcal{L}(x') - \mathcal{L}(x)$$

This metric isolates the fluency degradation introduced by the mutation itself. By subtracting the baseline loss of the original input, the measure reduces the influence of the intrinsic difficulty of the original sentence and focuses on the perturbation introduced by the metamorphic edit. A mutation x' is rejected if $\mathcal{M}_{lps}(x, x') > \tau_{lps}$, thereby reducing the likelihood that accepted variants are linguistically unnatural or far from the original data distribution.

3.4 Research Questions

$\mathbf{RQ_1}$ (configuration sensitivity). *How do different attribution, mutation, and semantic verification configurations affect the validity and failure discovery capability of metamorphic*

robustness testing?

The study evaluates multiple configurable components involved in the generation and verification of metamorphic variants, including attribution mechanisms, mutation strategies, and semantic filtering criteria.

RQ_2 (effectiveness and efficiency). *How effective and efficient are attribution-guided metamorphic testing configurations at discovering prediction inconsistencies across different models and datasets?*

To assess the practical applicability of our study, we evaluate the capability of attribution-guided metamorphic testing to expose robustness violations while maintaining reasonable computational overhead across diverse architectures and classification tasks.

RQ_3 (human validation). *To what extent does the automated semantic verification gate agree with human judgments of label preservation and fluency?*

This RQ assesses whether variants accepted by the semantic gate are judged valid by human annotators. It measures agreement between automatic gate decisions and human annotations for label preservation and fluency.

3.5 Metrics

For RQ_1 and RQ_2, we evaluate the effectiveness of the evaluated testing configurations using two metrics: (i) the *Attack Success Rate (ASR)*, defined as the proportion of generated metamorphic variants that successfully change the SUT prediction, and (ii) the number of *verified failures*, i.e., prediction-changing variants that satisfy the semantic verification criteria. To assess efficiency, we report the average time required to generate the first verified failure.

For RQ_3, we evaluate semantic-gate quality against human judgments. We report confusion statistics between automatic gate decisions and majority human labels, including accuracy and pass precision. Pass precision denotes the proportion of gate-accepted variants that are judged valid by humans.

3.6 Objects of Study

3.6.1 Datasets

To assess generalizability across diverse linguistic complexities and application domains, we adopted three text classification benchmarks, widely used in previous research [46, 8].

SST-2. The Stanford Sentiment Treebank (SST-2) [38] is a widely adopted GLUE benchmark consisting of 67k human-annotated movie reviews for binary sentiment analysis. We select SST-2 because it is notoriously rich in complex syntactic phenomena, such as contrastive conjunctions and nuanced negation. It serves as an ideal testbed to evaluate whether our metamorphic mutations effectively expose vulnerabilities in the model's compositional understanding.

AG News. For general topic classification, we employ the AG News dataset [51], containing 120k training samples categorized into four distinct news domains (World, Sports, Business, Sci/Tech). This dataset evaluates the model's reliance on broad vocabulary and named entities, acting as a standard control to verify that the Judge LLM can perform adversarial injections without drifting across strict topical boundaries.

GitHub Issue Classification. We utilize the large-scale GitHub Issue dataset originally curated for the NLBSE tool competition [19] and further refined for LLM fine-tuning tasks. This dataset consists of over 800k real-world issue titles and descriptions labeled into three

Table 2 Details of the SUTs and their original performance on the target datasets. Accuracies (%) are reported on the standard test/validation splits after supervised fine-tuning.

Model	Architecture	Params	Original Accuracy (%) ↑		
			SST-2	AG News	GitHub
DistilBERT-base-uncased [35]	Encoder-only	66M	91.17	92.22	85.63
Qwen/Qwen2.5 (Discriminative) [41]	Decoder-only	0.5B	92.20	93.93	86.28
Qwen/Qwen2.5-(Generative) [41]	Decoder-only	0.5B	93.69	91.95	83.39
facebook/bart-large [21]	Encoder-decoder	406M	95.76	95.09	86.76

categories: Bug, Enhancement, and Question. The complexity of this dataset stems from two primary factors: Multilingualism, as the issues are sourced from global open-source communities, containing technical discussions in multiple languages (e.g., English, Chinese, German, and Japanese); and Structural Noise, as the text heavily interweaves natural language with code snippets, URLs, and stack traces. Classifying these issues requires the SUT to perform deep logical reasoning and resist linguistic interference rather than relying on superficial keyword matching.

3.6.2 Models

To evaluate generalizability across model architectures and inference paradigms, we selected four representative Transformer-based models spanning Encoder-only, Encoder–Decoder, and Decoder-only architectures, as well as both discriminative and generative classification settings. Specifically, we evaluate `Qwen2.5-0.5B` under both discriminative and generative formulations to analyze how the inference paradigm influences susceptibility to metamorphic perturbations. Although the evaluated architectures originate from general-purpose pretrained language models, the systems under test are fine-tuned for downstream classification tasks and deployed as task-adapted inference pipelines. The detailed model specifications and supervised fine-tuning accuracies are reported in Table 2. All SUTs achieve strong baseline performance after fine-tuning, with accuracies ranging from 83.39% to 95.76% across the evaluated datasets. Among the evaluated models, `facebook/bart-large` achieves the highest overall accuracy.

3.6.3 Configurations and Baselines

To systematically analyze the impact of attribution and mutation choices, we define a configuration space of $4 \times 5 = 20$ setups obtained by combining four attribution mechanisms with five mutation strategies.

Attribution mechanisms (4 variants). We compare three attribution-based token prioritization methods—*Integrated Gradients* (IG) [40], *Occlusion* [49], and attention-based attribution [1, 5]—against a random-selection baseline without saliency guidance.

Mutation strategies (5 variants). We compare two LLM-based semantic mutation operators—*LLM-Ablate* and *LLM-Inject*—against three rule-based mutation strategies: *In-situ Ablation* (direct token deletion), *Prefix Injection* (trigger-token prepending), and *Random Injection*.

Several configurations correspond to established baselines commonly used in NLP robustness testing. The unguided setup without attribution prioritization or LLM-based rewriting resembles invariance-style perturbation testing as employed in CheckList [34], highlighting

the limitations of blind mutations and their tendency to introduce grammatical artifacts rather than semantically meaningful perturbations.

The combination of attribution-guided token selection with rule-based mutations mirrors the core intuition of TextBugger [22], where salient input regions are preferentially perturbed.

Similarly, prefix-based trigger insertion follows the attack-generation principle of XAI-Attack [3] and Universal Adversarial Triggers [45], where adversarial keywords are inserted at the beginning of the sequence to exploit positional sensitivity and shortcut behaviors.

Finally, combining attribution-guided token selection with direct in-situ deletion is inspired by the Representation Erasure paradigm [23]. This configuration isolates the effect of semantic rewriting and enables analysis of how semantic smoothing influences mutation validity and failure generation.

Implementation. To instantiate the Judge LLM ($\mathcal{J}$) for metamorphic generation, we deploy the *gpt-oss-20b* model locally using the Ollama framework. Using a locally hosted open-weights model ensures experimental reproducibility, avoids data privacy concerns and usage costs associated with cloud-based APIs, and prevents issues related to API version drift. The Judge LLM operates with a temperature of $T = 1.0$, enabling broader exploration of the mutation space. Potentially invalid variants are subsequently filtered by the two-stage semantic verification procedure. We compute the NLI scores using the *cross-encoder/nli-deberta-v3-small* model. Unlike standard bi-encoders that process sentences independently, cross-encoders perform full attention over the concatenated input pair, yielding higher accuracy and fine-grained sensitivity for textual entailment tasks [15, 27]. To calculate the perplexity, we utilize the standard *GPT-2* model [31]. As a foundational causal language model, GPT-2 serves as a highly efficient, lightweight, and universally accepted baseline for evaluating textual naturalness and perplexity in adversarial generation frameworks [14, 50].

All experiments, including supervised fine-tuning (SFT), XAI attribution computation, and metamorphic test generation, were conducted for three datasets and four model architectures on a single NVIDIA L40S GPU. The complete experimental pipeline required approximately 180 GPU hours.

3.7 Setup

3.7.1 Data Partitioning

To prevent data leakage and ensure a rigorous evaluation, we enforce a strict data partition. Specifically, we hold out a fixed subset of 2,000 instances from the original training data exclusively to compute XAI attributions and construct the adversarial candidate pools $\mathcal{T}(x)$. This holdout subset is strictly excluded from the SFT phase of the SUT. The remaining majority of the training data is utilized for standard SFT, while the original development and test splits are kept entirely untouched to evaluate model convergence and measure the final robust accuracy.

3.7.2 Supervised Fine-Tuning

To comprehensively evaluate the robustness of our target models, we adapt the fine-tuning pipeline to accommodate the fundamental architectural differences between discriminative and generative paradigms.

For discriminative settings, models are fine-tuned for standard sequence classification and appended with a linear classification head to output class probabilities. For generative settings, the classification task is reformulated as conditional text generation via instruction-based prompting. We wrap the inputs in a standardized template (e.g., *"Instruction: Classify*

the following GitHub issue... Category:"). To prevent the model from optimizing for input reconstruction, we apply target masking on prompts and inputs. This ensures the cross-entropy loss is computed exclusively on the generated target tokens (e.g., *"bug"*) and the sequence terminator.

To maintain experimental consistency and resource efficiency, all models undergo standard SFT. Following established best practices for ensuring fine-tuning stability in large language models [10, 28], we train the models using a learning rate of 2×10^{-5} with a warm-up ratio of 10%. To prevent overfitting, the training process is capped at 9 epochs (7 for generative architectures) and is strictly governed by an early stopping mechanism with a patience of 2 epochs.

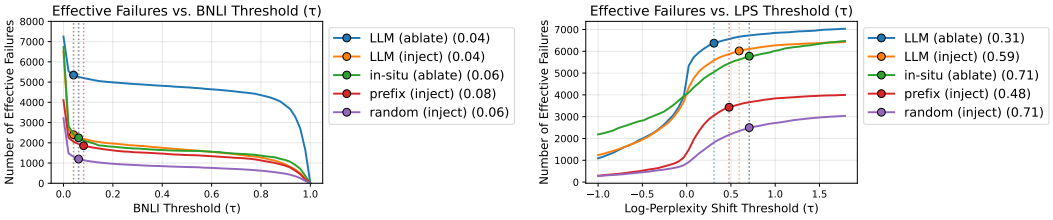
3.8 Evaluation Procedure

For RQ_1, we evaluate the effectiveness of different attribution methods for identifying failure-inducing mutation targets under the direct MR-Ablation setting, where attribution scores alone determine which token is perturbed. For each (Model, Explainer) pair, we measure the ASR under three ablation strategies: removing the token with the highest positive attribution score, removing the token with the strongest negative attribution score, and removing a randomly selected token as a baseline. Effective attribution methods are expected to produce substantially higher ASRs when perturbing positively attributed tokens compared to negatively attributed or randomly selected tokens, indicating that the explainer successfully identifies prediction-relevant regions.

To calibrate the semantic verification stage, we determine the thresholds of the entailment and perplexity filters using the Kneedle algorithm [36], which identifies inflection points in the metric distributions. For the entailment metric $\mathcal{M}_{nli}$, the threshold τ_{nli} is selected at the point preceding the sharp decline in retained failure-inducing variants, balancing semantic strictness and sample retention. For the linguistic naturalness metric $\mathcal{M}_{lps}$, the threshold τ_{lps} is selected at the elbow of the cumulative distribution to remove highly unnatural perturbations while retaining the majority of linguistically plausible variants.

For RQ_2, we instantiate the evaluation using the most effective attribution method identified in RQ_1 and compare attribution-guided mutation selection against random selection to isolate the contribution of explainability-guided targeting. We evaluate both the Attack Success Rate and the number of generated failure-inducing variants under two conditions: *pre-verification*, which includes all generated mutations, and *post-verification*, which includes only variants satisfying both semantic thresholds τ_{nli} and τ_{lps}. We further perform statistical analyses for the ASR difference between the studied configuration and its baseline. We report the mean paired difference, a non-parametric 95% bootstrap confidence interval over paired blocks [42], a one-sided Wilcoxon signed-rank test to estimate the stability of observed effects [47], and Cohen's d as effect size [9].

For RQ_3, we conduct a human evaluation study to assess whether the semantic verification stage aligns with human judgments of mutation validity. We sample mutations from all evaluated tasks, drawn throughout different attribution methods and mutation strategies. After removing empty outputs and non-English examples, the final evaluation set contains 383 mutation pairs. Annotators recruited through Amazon Mechanical Turk [39] are shown the task description, the original task label, the original input, and the mutated variant. They then evaluate whether the mutated input preserves the original label and whether it remains fluent and natural. Each annotation batch contains one attention-check question, and submissions failing the attention check are discarded. In total, we collect 92 submissions, of which 84 pass the quality filter, yielding 818 valid item-level annotations. A mutation is



(a) Retained effective failures across varying BNLI **(b)** Retained failures across LPS thresholds ↓ thresholds ↑

Figure 1 Sweet-spot analysis for determining automated verification thresholds using the Kneedle algorithm. Circular markers "○" denote the algorithmically identified optimal thresholds (τ) for each mutation strategy.

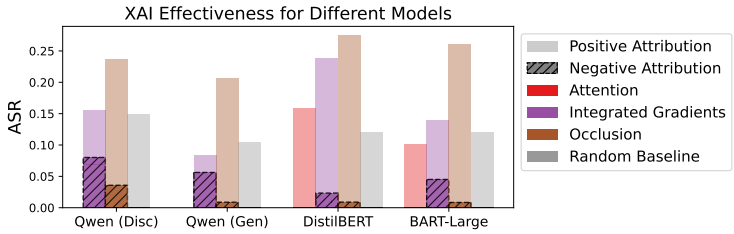


Figure 2 Effectiveness of different XAI explainers in guiding Metamorphic Ablation.

considered human-valid only if it is judged both label-preserving and fluent.

4 Results

4.1 Configuration sensitivity (RQ_1)

Figure 2 reports the Attack Success Rate obtained by different attribution methods under the direct MR-Ablation setting. Across all evaluated architectures, Occlusion consistently achieves the highest post-verification ASR values. In contrast, perturbing tokens with the strongest negative attribution scores (hatched bars) produces substantially lower ASRs. This clear separation provides an important sanity check, indicating that the attribution methods identify prediction-relevant regions rather than merely degrading sentence fluency through arbitrary token removal.

Among the evaluated explainers, Occlusion emerges as the most stable and architecture-agnostic strategy for identifying vulnerability-prone tokens. Its effectiveness remains consistent across encoder-only, encoder–decoder, and decoder-only architectures. Based on these results, we select Occlusion as the default attribution method for the remaining experiments.

Figure 1 illustrates the number of retained failure-inducing variants under varying semantic verification thresholds. Across both semantic metrics, LLM-based semantic mutation strategies consistently produce substantially more semantically valid failures than rule-based baselines, indicating a higher ability to generate boundary-compliant perturbations while preserving linguistic plausibility.

To determine robust semantic verification thresholds, we apply the Kneedle algorithm to identify inflection points balancing semantic validity and sample retention. As shown in Figure 1(a), the number of retained failures decreases sharply as the bidirectional entailment threshold increases from 0.0 to 0.1, indicating that a large portion of raw perturbations

Table 3 Effectiveness of the evaluated mutation configurations across three datasets. We report the ASR results **Before/After** Semantic Verification Gate filtering. Best results after verification are marked in **bold**.

MR	Strategy	Qwen-0.5B (Disc)		Qwen-0.5B (Gen)		DistilBERT		BART-Large	
		Occlusion	Random	Occlusion	Random	Occlusion	Random	Occlusion	Random
SST-2									
ablate	lm	5.52/2.83	6.43/**3.35**	4.30/**1.81**	2.87/1.36	7.10/**2.54**	4.72/2.28	5.67/**2.28**	4.49/1.85
	in-situ	15.40/2.79	14.83/2.55	12.73/0.86	10.39/0.94	17.57/1.71	12.10/1.17	18.31/1.51	11.99/1.37
inject	lm	9.57/0.85	8.40/0.53	6.68/0.52	4.44/0.31	8.70/0.48	6.31/0.21	6.76/0.32	6.34/0.32
	prefix	5.85/0.53	5.79/0.74	7.67/0.94	6.32/0.63	8.22/1.11	5.04/1.43	6.08/0.63	6.50/0.58
	random	10.79/0.80	9.62/0.96	5.01/0.42	4.80/0.26	7.32/0.32	4.77/0.42	5.92/0.42	5.86/0.37
News									
ablate	lm	2.52/1.96	2.85/1.74	18.73/**14.42**	18.31/12.08	1.56/0.86	0.98/0.44	0.85/**0.43**	0.37/0.26
	in-situ	3.71/**2.21**	3.43/1.85	2.37/1.12	2.84/1.58	2.37/0.84	0.76/0.33	1.38/0.34	0.58/0.21
inject	lm	2.64/0.32	2.75/0.26	18.63/2.68	16.28/1.97	1.53/0.33	1.91/0.27	1.20/0.16	1.68/0.05
	prefix	1.06/0.37	1.06/0.21	9.89/4.10	10.22/3.83	2.56/**0.87**	1.36/0.44	1.36/0.21	2.25/0.37
	random	3.27/0.90	3.01/0.69	4.70/1.58	5.30/1.37	1.96/0.55	1.20/0.22	0.73/0.26	1.41/0.42
GitHub									
ablate	lm	8.70/6.62	8.34/6.08	8.31/5.89	6.73/4.93	11.08/**8.48**	10.88/6.98	10.49/**7.45**	7.89/5.63
	in-situ	2.78/1.85	2.55/1.56	4.62/2.35	3.13/1.20	4.03/2.42	1.80/1.11	3.79/2.37	1.74/1.22
inject	lm	13.01/7.08	13.41/**7.37**	12.45/5.76	12.91/6.62	16.01/8.32	13.20/7.09	15.11/6.34	13.61/7.09
	prefix	3.69/2.13	3.40/2.65	15.36/9.21	14.77/**10.46**	6.00/3.55	5.70/2.73	4.67/3.00	3.69/2.13
	random	2.53/1.44	1.96/1.44	7.22/4.30	8.74/5.56	2.50/1.05	2.38/1.05	2.54/1.38	2.36/1.33

exhibit semantic drift. Based on the detected inflection point, we select a unified threshold of $\tau_{nli} = 0.08$, which filters semantically inconsistent variants while preserving the majority of semantically valid failures.

Figure 1(b) shows the cumulative distribution of log-perplexity shift values. The curve follows an S-shaped pattern, with inflection points located approximately in the range $\tau_{lps} \in [0.0, 0.7]$. We select a unified threshold of $\tau_{lps} = 0.3$, which removes highly unnatural perturbations while retaining most linguistically plausible variants.

Configuration sensitivity (RQ_1): *Occlusion consistently provides the most effective attribution guidance across architectures. Kneedle-based calibration identifies stable thresholds ($\tau_{nli} = 0.08$ and $\tau_{lps} = 0.3$) that balance semantic preservation and retained failure coverage.*

4.2 Effectiveness and efficiency (RQ_2)

Table 3 summarizes the effectiveness of the evaluated mutation configurations across three datasets and four model architectures. We report the ASR both **before** and **after** semantic verification using the entailment and perplexity thresholds (τ_{nli} and τ_{lps}). This comparison provides insight into both the vulnerability of the evaluated models and the validity of the generated adversarial variants.

Across most evaluated settings, attribution-guided configurations combined with LLM-based semantic mutations generally increase post-verification ASR. The results additionally reveal notable differences across modeling paradigms. Generative settings exhibit substantially higher susceptibility to semantically valid adversarial insertions than discriminative counterparts. For example, on AG News, the generative Qwen configuration achieves a post-verification ASR of 16.28% under contextual injection, compared to 2.64% for its discriminative variant and 1.91% for DistilBERT.

Table 4 Pairwise comparisons using post-verification ASR. We report non-parametric 95% bootstrap confidence intervals (CI), one-sided Wilcoxon test, and paired Cohen's d_z as effect sizes.

Comparison	n	Method	Baseline	[95% CI]	Wilcoxon p	d_z
LLM-Ablate vs. In-situ Ablation	24	4.27	1.48	**[1.51, 4.31]**	**<0.001**	0.79^{M}
LLM-Inject vs. Random Injection	24	2.72	1.15	**[0.61, 2.66]**	**0.027**	0.60^{M}
LLM-Inject vs. Prefix Injection	24	2.72	2.20	[-0.45, 1.56]	0.634	0.20^{S}
Occlusion + LLM-Ablate vs. Random + LLM-Ablate	12	4.63	3.91	**[0.30, 1.17]**	**0.005**	0.89^{L}
Occlusion + LLM-Inject vs. Random + LLM-Inject	12	2.76	2.67	[-0.22, 0.40]	0.252	0.16^{N}

Notes. The **bold** values indicate $\mathrm{CI}_{95\%}(\Delta)$ is entirely above zero, or $p < 0.05$. Effect sizes use paired Cohen's d_z: ^{N} negligible ($d < 0.20$), ^{S} small ($0.20 \leq d < 0.50$), ^{M} medium ($0.50 \leq d < 0.80$), and ^{L} large ($d \geq 0.80$).

A central observation emerging from Table 3 is the substantial discrepancy between pre-verification and post-verification ASR. In many cases, unconstrained perturbation strategies initially appear highly successful, but most generated variants fail semantic validation. For instance, under direct lexical ablation on SST-2 (DistilBERT), the pre-verification ASR reaches 17.57%, but decreases to only 1.71% after semantic filtering. This large reduction indicates that many seemingly successful adversarial examples correspond to semantically invalid or label-altering perturbations rather than genuine robustness violations.

The semantic verification stage therefore plays a critical role in filtering invalid adversarial artifacts and ensuring that reported failures correspond to semantic-preserving inconsistencies. Under the default occlusion-guided configuration, LLM-based semantic mutations achieve a mean post-verification ASR of 3.70% across evaluated datasets and architectures, with semantic ablation reaching 4.63% and contextual injection 2.76%. In comparison, heuristic baselines achieve substantially lower verified ASR values, including In-situ Ablation (1.48%), Prefix Injection (2.20%), and Random Injection (1.15%).

Table 4 reports the pairwise tests using post-verification ASR. The most consistent gain comes from LLM-Ablate, which significantly improves over in-situ deletion and also benefits from Occlusion-guided target selection. For injection, LLM-Inject significantly improves over Random Injection, but not significantly over the stronger Prefix Injection baseline. Thus, the evidence supports a robust advantage for LLM-based ablation, while injection improvements are more setting-dependent.

To assess computational practicality, we additionally measured the average execution time required to generate and verify a single metamorphic variant. Rule-based perturbation strategies remain highly efficient, requiring approximately 0.22 seconds per generated attack on average. In contrast, LLM-based semantic mutations require approximately 4.71 seconds due to the overhead introduced by local LLM inference and semantic verification. Although computationally more expensive, these configurations produce substantially higher proportions of semantically valid robustness violations.

Effectiveness and efficiency ($\mathbf{RQ_2}$): *LLM-based ablation provides the most robust gain over hard deletion, while LLM-based injection improves over random injection but not reliably over the stronger prefix baseline. Semantic verification removes a large fraction of invalid adversarial artifacts, demonstrating that unconstrained perturbations substantially overestimate model brittleness.*

Table 5 Human validation of semantic verification and mutation configurations.

(a) Gate-level validation: confusion statistics between automatic gate decisions and human judgments. Pass precision is the human-positive rate among accepted variants.

Comparison	TN	FP	FN	TP	Accuracy [%]	Pass precision [%]
LPS gate vs. fluent	29	13	66	209	75.1	76.0
NLI gate vs. label-pres.	42	66	10	179	74.4	94.7

(b) Configuration-level validation. Human-valid variants are both label-preserving and fluent.

Configuration	**full gate [%]**	**human-valid [%]**	**label-pres. [%]**	**fluent [%]**	**gap [pp]**
LLM-Ablate	80.6	74.1	89.8	95.5	+6.5
LLM-Inject	37.3	41.3	78.1	77.0	-4.0
Prefix-Inject	47.8	23.9	81.1	42.1	+23.9
In-situ Ablate	37.5	19.3	73.7	43.5	+18.2
Random Inject	31.4	8.6	81.0	21.7	+22.9

4.3 Human validation (RQ_3)

Table 5 reports the human validation results. At the gate level, the NLI gate achieves a high pass precision of 94.7%. However, a non-trivial portion of rejected variants are still judged valid by humans, suggesting that the gate is conservative and prioritizes precision over recall. This behavior is appropriate for our setting, where reducing false positives is more important than maximizing mutation retention.

At the configuration level, LLM-Ablate achieves the highest human-valid rate and closely matches its automatic gate valid rate, suggesting that LLM-based ablation reliably preserves the intended metamorphic relation. For injection, LLM-Inject produces more human-valid and fluent variants than Prefix and Random Injection, even though its ASR advantage over Prefix is not statistically reliable in RQ2. This indicates that Prefix Injection is a strong failure-discovery baseline but often produces less natural variants. Overall, the semantic gate overestimates the quality of heuristic methods as the gaps are non-trivial, and LLM-based mutation produces more natural and usable test cases.

Human validation (RQ_3): *The semantic gate acts as a high-precision but conservative validity filter. Human validation supports the use of LLM-based semantic rewriting, especially for producing natural and label-preserving variants.*

5 Threats to Validity

5.1 Internal validity

Our semantic verification gate may still admit subtle meaning drift, particularly under lower threshold values. To mitigate this threat, we combine bidirectional entailment, fluency filtering, and manual inspection of a stratified sample of accepted and rejected mutations. Prior work has shown that bidirectional entailment provides a strong approximation of semantic preservation in adversarial NLP testing settings.

LLM-based mutation generation is inherently stochastic and may be sensitive to prompting and decoding choices. To reduce this threat, we use fixed prompts, decoding parameters, and identical computational budgets across all configurations. Moreover, all baselines are

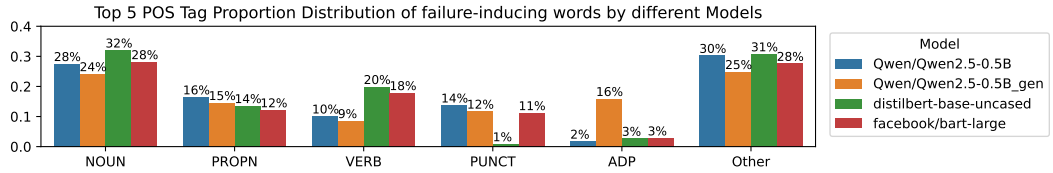


Figure 3 POS analysis of failure-inducing words.

implemented within the same framework and evaluated under identical settings, isolating the effect of the mutation strategy itself.

Finally, our approach relies on LLMs both for mutation generation and semantic verification, potentially introducing correlated biases. We mitigate this risk by separating generation and verification stages, combining multiple validation signals, and complementing automated filtering with manual inspection.

5.2 External validity

The generalizability of our findings may depend on the selected datasets, task semantics, mutation operators, and Judge LLM. To reduce overfitting conclusions to a specific setting, we evaluate multiple datasets and model families, including encoder-only, encoder–decoder, and decoder-only architectures.

Furthermore, our evaluation primarily relies on Attack Success Rate, which captures prediction inconsistencies but may not fully reflect the practical severity of failures. Finally, our mutation operators cover only a subset of possible semantic-preserving transformations; different perturbation spaces or future LLMs may lead to different behaviors.

6 Qualitative Analysis

Beyond detecting failures, it is important to understand the patterns that trigger them. To this end, we conduct a qualitative and statistical analysis of 2320 verified failures produced by the best RQ_2 configuration across all three datasets, focusing on the failure-inducing words isolated by XAI and categorizing them by their Part-of-Speech (POS) tags. As illustrated in Figure 3, the distribution reveals systemic biases and distinct architectural vulnerabilities.

Across all four model architectures, Nouns (`NOUN`) consistently constitute the largest vulnerability surface. Furthermore, the nominal features, combining Nouns (`NOUN`) and Proper Nouns (`PROPN`), contribute nearly half of the critical vulnerabilities. This strongly indicates that the SFT LMs frequently overfit to specific topical keywords or entities and bypass robust semantic reasoning.

To corroborate our statistical findings, we extract and inspect the specific failure-inducing tokens and their original contexts. This qualitative analysis unveils some distinct patterns of shortcut learning and semantic overfitting across the target models. On the GitHub issue classification task, models like BART-Large exhibit a severe reliance on explicit, human-injected meta-words rather than comprehending the issue's technical description. For instance, ablating the literal word `"bug"` or `"question"` from the issue body consistently causes the model to misclassify the artifact, even when the surrounding text clearly describes an error trace (e.g., *RuntimeError: source object number out of range'*) or a user query (e.g., *May I ask if I can get a pull request...?'*). This reveals that the model acts largely as a shallow keyword-matcher, failing to generalize to issues where developers omit explicit category

tags. In news categorization, discriminative models (e.g., DistilBERT) frequently exploit Named Entities (`ENT_TYPE=ORG`) as predictive shortcuts. We observe that models heavily rely on publisher acronyms (e.g., `"AFP"`, `"AP"`, `"Reuters"`) or dominant corporate entities (e.g., `"Apple"`, `"Pepsi"`) to determine the news category. For example, ablating `"(AFP)"` from a headline about European stocks causes the model's prediction to flip. This indicates a vulnerability where the SUT memorizes the data source distribution rather than reasoning about the semantic content of the headline.

7 Discussion

Effectiveness of Attribution-Guided Mutation Selection. The results of our study indicate that attribution-guided mutation selection substantially improves the discovery of verified robustness violations compared to unguided or purely heuristic perturbation strategies. Across datasets and architectures, configurations based on attribution signals generally achieve higher verified attack success rates and expose more semantically valid failures than random mutation baselines. These findings suggest that attribution methods provide an effective heuristic for identifying decision-relevant regions of the input space that are particularly sensitive to localized perturbations.

At the same time, the results also reveal differences across attribution techniques. Gradient-based and perturbation-based methods exhibit varying effectiveness depending on the model architecture and task, indicating that no single attribution strategy universally captures the most vulnerability-prone input regions. This variability further highlights the importance of systematically evaluating attribution methods within robustness testing settings rather than assuming uniform effectiveness across models.

Impact of Semantic Verification. A central finding of the study is that unconstrained adversarial generation substantially overestimates model brittleness. Many perturbations that successfully alter model predictions fail to preserve the original semantics or introduce severe linguistic degradation. Without semantic verification, these invalid mutations inflate attack success rates and provide misleading estimates of robustness weaknesses.

The human results also show why ASR alone is insufficient for comparing test-generation strategies. A configuration may be effective at inducing prediction flips while still producing variants that are unnatural, awkwardly inserted, or less useful for diagnosis. The results therefore demonstrate that semantic validation is essential for obtaining reliable robustness measurements in metamorphic testing settings. Human validity further adds a complementary quality dimension, which is particularly important when comparing context-aware LLM mutations with stronger but more mechanical heuristics such as prefix-trigger insertion. The latter can be competitive in ASR, but human validation reveals that context-aware rewriting produces more natural and actionable variants.

Evidence of Shortcut Reliance. The discovered failures provide further evidence that task-specialized language models frequently rely on brittle lexical patterns and shortcut correlations. In many cases, small semantic-preserving edits affecting highly attributed tokens are sufficient to trigger prediction inconsistencies. This behavior is particularly pronounced for entity names, sentiment-bearing adjectives, and dataset-specific lexical indicators that strongly correlate with training labels.

These observations align with prior work on spurious correlations and annotation artifacts in NLP datasets, suggesting that strong in-distribution accuracy does not necessarily imply robust linguistic understanding. The results additionally indicate that attribution-guided perturbations can serve as a practical mechanism for systematically exposing such shortcut-

driven behaviors.

Limitations. The evaluated metamorphic relations focus primarily on lexical-level transformations. Although these perturbations already expose substantial robustness weaknesses, they do not capture higher-level linguistic phenomena such as discourse structure, reasoning consistency, or conversational context. Second, the semantic verification mechanisms rely on pretrained NLI and language models, which may themselves exhibit biases or verification errors. Third, the evaluation is limited to text classification tasks and moderate-sized task-specialized language models; additional studies are required to assess whether the observed findings generalize to larger generative models and more complex NLP tasks such as summarization, dialogue generation, or question answering.

Despite these limitations, the study demonstrates that combining attribution-guided mutation selection with semantic verification provides reliable and scalable evidence for robustness evaluation in specialized language models.

8 Related Work

8.1 Test Generation for Deep Learning and Language Models

Testing techniques for deep learning systems primarily focus on generating inputs that expose unstable, incorrect, or brittle model behaviors. Early work on adversarial testing demonstrated that small perturbations to the input can cause incorrect predictions while preserving human-perceived semantics. In NLP, character-level and word-level attacks such as HotFlip and DeepWordBug generate adversarial examples through gradient-based or heuristic perturbation strategies [11, 12]. Subsequent studies shifted toward semantically constrained attacks that aim to preserve fluency and meaning. For example, Alzantot et al. used genetic search over embedding neighbors to generate natural adversarial examples [2], while TextFooler prioritizes important tokens and replaces them with semantically similar alternatives [17]. More recent approaches employ masked language models to generate context-aware substitutions and insertions that produce more fluent perturbations [24, 14].

Beyond adversarial attacks, behavioral testing approaches assess model robustness through systematic input transformations designed to verify specific linguistic capabilities. CheckList introduced a taxonomy of behavioral tests based on invariance, directional expectation, and minimal functionality properties [34]. Other works construct contrast sets and adversarial benchmarks to reveal weaknesses that are not captured by standard evaluation datasets [13, 30, 20]. Collectively, these studies show that language models often fail under relatively simple linguistic variations despite strong benchmark performance.

However, many existing techniques rely on manually designed templates, heuristic perturbations, or unconstrained search procedures. Prior evaluations also report that adversarial generation methods frequently produce invalid examples that unintentionally alter semantics or degrade linguistic quality [27]. To address this problem, prior work has proposed semantic and naturalness checks for generated NLP test cases. AEON, for example, evaluates whether generated NLP test cases preserve semantic similarity and natural language quality [16]. Metamorphic Testing (MT) has also been adapted to NLP through predefined metamorphic relations that verify prediction consistency under semantic-preserving transformations [6, 37, 8]. Nevertheless, many existing MT techniques still rely on random or weakly guided perturbations and provide limited guarantees regarding semantic validity.

This study is motivated by the observation that these components are often evaluated in isolation. We extend prior work by empirically investigating the combination of attribution-guided mutation selection, LLM-based semantic transformations, and automated semantic

verification for robustness testing of task-specialized language models.

8.2 Detecting Spurious Correlations in Language Models

A growing body of research investigates how language models exploit superficial patterns and spurious correlations present in training data. Adversarial evaluation studies have shown that models frequently rely on shallow lexical cues instead of robust linguistic reasoning. Universal adversarial triggers, for instance, demonstrate that short token sequences can systematically manipulate model predictions across many inputs [45]. Similarly, adversarial NLI benchmarks reveal that models often exploit annotation artifacts and dataset-specific heuristics rather than genuine semantic understanding [30].

Explainable AI techniques have also been widely used to diagnose such behaviors by identifying which input regions contribute most strongly to model predictions [52]. Attribution methods such as Integrated Gradients [40], attention-based analysis [1], and relevance propagation approaches [5] provide insight into the evidence used by neural models during inference [7]. These attribution signals have subsequently been incorporated into adversarial example generation and robustness analysis pipelines [17, 3].

Despite these advances, explainability methods are predominantly used as post-hoc diagnostic tools rather than as systematic mechanisms for generating semantically controlled robustness tests. This study builds on prior findings by empirically evaluating whether attribution signals can effectively guide metamorphic mutation selection and improve the generation of valid failure-inducing test cases. We also analyze the resulting failures to characterize recurring shortcut patterns, such as over-reliance on named entities and topical keywords. In doing so, the study connects attribution-based diagnosis with metamorphic robustness testing. The results provide further evidence on the relationship between attribution-guided perturbations, semantic validity, and the exposure of brittle decision patterns in specialized language models.

9 Conclusions

In this paper, we presented an empirical study on XAI-guided metamorphic robustness testing for task-specialized language models. The study investigated how attribution-guided mutation prioritization, LLM-based semantic perturbations, and semantic verification mechanisms influence the generation of valid failure-inducing test variants without requiring explicit test oracles.

Our evaluation across multiple datasets, architectures, and classification paradigms shows that combining explanation-guided selection with LLM-based mutation consistently exposes more robustness violations than unguided and rule-based baselines. Semantic verification filters a substantial portion of invalid adversarial artifacts. The human study further shows that the semantic gate is high-precision but conservative, making it suitable for robustness testing settings where false positives are costly. The verified failures also reveal shortcut-driven behaviors, including sensitivity to named entities, lexical indicators, and formatting cues.

Future work will investigate richer metamorphic transformations beyond lexical edits, including structural, discourse-level, and conversational perturbations, as well as adaptive robustness testing settings in which discovered failures iteratively guide subsequent model refinement and evaluation.

10 Data Availability Statement

We provide an anonymous replication package with our experimental pipeline and processed datasets [43]. A public DOI is not released at submission time to avoid premature dissemination; the repository will be archived with a DOI upon acceptance.

References

1. Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, 2020. doi:10.18653/v1/2020.acl-main.385.
2. Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.
3. Markus Bayer, Markus Neiczer, Maximilian Samsinger, Björn Buchhold, and Christian Reuter. XAI-attack: Utilizing explainable AI to find incorrectly learned patterns for black-box adversarial example creation. In *LREC-COLING*, Torino, Italia, 2024. ELRA and ICCL.
4. Yupeng Chang et al. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3), March 2024. doi:10.1145/3641289.
5. Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791, 2021. doi:10.1109/CVPR46437.2021.00084.
6. Tsong Yueh Chen, Siu Cheung, and Siu-Ming Yiu. Metamorphic testing: A new approach for generating next test cases. *Technical Report HKUST-CS98-01*, 1998.
7. Xingcheng Chen, Oliver Weissl, and Andrea Stocco. Feature-aware test generation for deep learning models, 2026. URL: https://arxiv.org/abs/2601.14081.
8. Steven Cho, Stefano Ruberto, and Valerio Terragni. Metamorphic testing of large language models for natural language processing. In *2025 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 174–186, 2025.
9. Jacob Cohen. *Statistical power analysis for the behavioral sciences*. L. Erlbaum Associates, Hillsdale, N.J, 1988.
10. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi:10.18653/v1/N19-1423.
11. Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi:10.18653/v1/P18-2006.
12. Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56, 2018. doi:10.1109/SPW.2018.00016.
13. Matt Gardner et al. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323. Association for Computational Linguistics, November 2020.
14. Siddhant Garg and Goutham Ramakrishnan. BAE: BERT-based adversarial examples for text classification. In *EMNLP*, pages 6174–6181. Association for Computational Linguistics, November 2020.
15. Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021. arXiv:2111.09543.

16 Jen-tse Huang, Jianping Zhang, Wenxuan Wang, Pinjia He, Yuxin Su, and Michael R. Lyu. Aeon: a method for automatic evaluation of nlp test cases. ISSTA 2022, page 202–214, New York, NY, USA, 2022. Association for Computing Machinery.

17 Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *AAAI Conference on Artificial Intelligence*, 2019.

18 Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

19 Rafael Kallis, Oscar Chaparro, Andrea Di Sorbo, and Sebastiano Panichella. Nlbse'22 tool competition. In *2022 IEEE/ACM 1st International Workshop on Natural Language-Based Software Engineering (NLBSE)*, pages 25–28. IEEE, 2022. doi:10.1145/3528588.3528664.

20 Douwe Kiela et al. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, June 2021.

21 Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. Association for Computational Linguistics, July 2020. doi:10.18653/v1/2020.acl-main.703.

22 Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. In *Proceedings of NDSS*, 2019.

23 Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. In *Proceedings of arXiv:1612.08220*, 2016.

24 Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202. Association for Computational Linguistics, November 2020.

25 Percy Liang et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. Featured Certification, Expert Certification.

26 Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

27 John X Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Conference on Empirical Methods in Natural Language Processing*, 2020.

28 Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations (ICLR)*, 2021.

29 Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906, Melbourne, Australia, July 2018. Association for Computational Linguistics.

30 Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901. Association for Computational Linguistics, July 2020.

31 Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

32 Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

33 Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California, June 2016. Association for Computational Linguistics.

34 Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912. Association for Computational Linguistics, July 2020.

35 Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

36 Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171, 2011.

37 Sergio Segura, Gordon Fraser, Ana B. Sanchez, and Antonio Ruiz-Cortes. A survey on metamorphic testing. *IEEE Transactions on Software Engineering*, 42(9):805–824, 2016.

38 Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*. Association for Computational Linguistics, 2013.

39 Alexander Sorokin and David Forsyth. Utility data annotation with amazon mechanical turk. In *2008 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 1–8. IEEE, 2008.

40 Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017.

41 Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL: `https://qwenlm.github.io/blog/qwen2.5/`.

42 Robert J Tibshirani and Bradley Efron. An introduction to the bootstrap. *Monographs on statistics and applied probability*, 57(1):1–436, 1993.

43 Replication package, 2026. URL: `https://anonymous.4open.science/r/lexcheck-7B5B/README.md`.

44 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

45 Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of EMNLP-IJCNLP*, 2019.

46 Alex Wang et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. `doi:10.18653/v1/W18-5446`.

47 Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80, December 1945. `doi:10.2307/3001968`.

48 Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

49 Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.

50 Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. Openattack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 363–371, 2021.

790 **51** Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text
791 classification. In *Proceedings of the 29th International Conference on Neural Information*
792 *Processing Systems - Volume 1*, pages 649–657, 2015.
793 **52** Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang,
794 Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM Trans.*
795 *Intell. Syst. Technol.*, 15(2), February 2024. URL: `https://doi.org/10.1145/3639372`.