Selective Perturbations as a Diagnostic for Benchmark-Based LLM Comparisons

Ivan Dubrovsky [1] **Nina Gubina** [2] **Anastasia Orlova** [2] **Illarion Iov** [1] **Irena Gureeva** [3] **Nikolay O. Nikitin** [1] **Alexey Zaytsev** [3]

Abstract

Benchmark accuracy is a useful summary of model performance, but it does not show how sensitive a model comparison is to question wording. We study this sensitivity with *selective perturbations*: small edits to multiple-choice questions that change the answer of one target model while preserving other models' answers. The question we ask is whether local edits can isolate the part of benchmark sensitivity that is specific to one model. We implement a reference-preserving search constraint and evaluate the resulting perturbations on both reference models used during search and unseen models held out from the search. On the full MMLU dev split, unconstrained perturbations often degrade several models at once. With the selectivity constraint, a large target-specific component remains: across Gemma-3-12B, Llama-3.1-8B, and Qwen3.5-9B, target accuracy drops by 0.38–0.44, while reference drops remain at most 0.04 and unseen-model drops at most 0.10. Smaller supporting experiments on GPQA Diamond, within the Gemma family, with Gemini-2.5-Flash as target, and with selective improvement show the same qualitative pattern. Manual inspection suggests that the target-specific component is structured: Qwen3.5-9B is more often affected by coarse substitutions that corrupt domain anchors, while Gemma-3-12B is affected by milder edits such as near-synonyms, register shifts, and casing changes. These results suggest that aggregate benchmark scores can hide not only how often models fail, but also which local changes expose their failures.

[1]NSS Lab, ITMO University, Saint Petersburg, Russia [2]Center for AI in Chemistry, ITMO University, Saint Petersburg, Russia [3]Applied AI Institute, Moscow, Russia. Correspondence to: Ivan Dubrovsky <idubrovsky@itmo.ru>, Alexey Zaytsev <Likzet@gmail.com>.

Accepted to the 1st Workshop on Combining Theory and Benchmarks, CTB@ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

1. Introduction

Benchmarks such as MMLU (Hendrycks et al., 2020), BIG-bench (Srivastava et al., 2023), GPQA (Rein et al., 2023), and evaluation suites such as HELM (Liang et al., 2022) are widely used to compare large language models. A single accuracy number is useful because it gives a compact summary of performance and makes models easy to rank. This convention continues the earlier leaderboard-based evaluation tradition of GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019). But the same compactness hides local structure: two models can obtain similar scores while relying on different parts of the question, different surface cues, or different interactions between the stem and the answer choices.

This matters most when benchmark scores are used comparatively. A small gap between two models may reflect a stable difference in performance, or it may depend on details of the particular item wording and evaluation protocol. Prior work has shown that evaluation results can shift under prompt-format changes (Sclar et al., 2024), answer-order permutations (Zheng et al., 2024), answer-extraction and implementation choices (Biderman et al., 2024), paraphrase-level edits (Alzahrani et al., 2024; Nalbandyan et al., 2025), dataset-quality issues (Gema et al., 2025), and benchmark contamination (Zhou et al., 2023; Xu et al., 2024; Zhao et al., 2025). These results suggest that an aggregate benchmark score is not only a property of a model and a dataset, but also of a particular rendering of the task.

We ask a more targeted question: can local edits isolate the part of benchmark sensitivity that is specific to one model rather than shared across models? This question differs from measuring whether a model can be attacked at all. A perturbation that lowers one model's accuracy may simply make the item harder for every model. For benchmark comparison, the more informative quantity is the gap between the target model's degradation and the degradation of the models against which it is being compared.

We study this question through *selective perturbations*. Given a target model and a set of non-target models, we search for edits to a multiple-choice question that change the

target's answer while preserving the others. This separates two effects that are mixed together in ordinary adversarial degradation. Some perturbations make an item broadly harder and transfer to many models. Others expose model-specific sensitivity: the target changes its answer, while other models remain stable. An overview of the protocol is shown in Figure 1.

To make this distinction meaningful, we separate non-target models into *reference* and *unseen* models. Reference models are used during perturbation search through a custom selectivity constraint. Unseen models are not queried during search and are evaluated only after perturbations are found. This prevents the main selectivity claim from relying only on the models whose stability was explicitly enforced, and turns selective perturbation search into a diagnostic for comparative robustness rather than a standard attack benchmark.

Overall, this paper makes the following contributions:

- We use selective perturbations as a diagnostic for decomposing benchmark sensitivity into shared and model-specific components. Non- selective attacks reveal broad sensitivity; selective attacks isolate cases where one model is affected while others remain stable.
- We implement this diagnostic with TextAttack (Morris et al., 2020) and a reference-preserving constraint, and evaluate the resulting perturbations on held-out unseen models. This reference/unseen split separates constraint satisfaction during search from post-hoc evidence of selectivity.
- On the full MMLU dev split, we find that selective perturbations preserve a large target-specific drop while sharply reducing collateral effects on reference and unseen models. The same pattern holds on GPQA Diamond as a second benchmark.
- We test the same diagnostic in supporting settings covering same-family models, an API-only frontier target, character-level attacks, and the reverse objective of selective improvement.
- We qualitatively inspect successful perturbations and find that the target-specific component is structured rather than uniform across models.

Conceptually, aggregate accuracy estimates capability well only if a model's correctness depends on the meaning of an item and not on its surface form. Selective perturbations test this directly: a meaning-level edit that changes one model's answer but not the others' shows that, for that model, correctness depends on surface form in a way the comparison set does not share.

2. Related Work

Benchmark sensitivity. Several studies quantify how LLM benchmark measurements change under evaluation choices that are not part of the nominal task. Prompt templates and formatting choices can substantially affect model behavior (Sclar et al., 2024), and multiple-choice models are sensitive to answer-order permutations and selection biases (Zheng et al., 2024). Other work studies benchmark perturbations and leaderboard stability, showing that rankings can shift under paraphrases, item variants, or alternative subsets (Alzahrani et al., 2024; Nalbandyan et al., 2025). Broader evaluation frameworks such as HELM (Liang et al., 2022) and reproducibility-focused work on language-model evaluation (Biderman et al., 2024) similarly emphasize that scores depend on implementation details, prompting conventions, and scoring choices. Benchmark refinements such as MMLU-Pro (Wang et al., 2024a) and audits such as MMLU-Redux (Gema et al., 2025) further reflect the need for evaluation protocols that expose more than a single aggregate score. Contamination studies (Zhou et al., 2023; Xu et al., 2024; Zhao et al., 2025) raise a related concern: static benchmark scores may partly reflect training-data exposure rather than stable task competence. These works motivate our setting, but most report aggregate changes in score. We instead condition on the stability of other models to isolate the part of the change that is specific to a target model.

Adversarial text perturbations. Text attacks span character-level methods (Ebrahimi et al., 2018; Gao et al., 2018), word-level substitutions (Ren et al., 2019; Jin et al., 2020; Garg & Ramakrishnan, 2020; Li et al., 2020), and sentence-level paraphrasing, syntactic transformations, or style transfer (Iyyer et al., 2018; Ribeiro et al., 2018; Qi et al., 2021). Related robustness-testing frameworks, including CheckList (Ribeiro et al., 2020), Robustness Gym (Goel et al., 2021), TextFlint (Gui et al., 2021), and AdvGLUE (Wang et al., 2021), use transformations or challenge sets to probe failures that are hidden by held-out accuracy.Frameworks such as TextAttack (Morris et al., 2020), OpenAttack (Zeng et al., 2021), and PromptBench (Zhu et al., 2024) provide abstractions for transformations, constraints, and search. We use TextAttack as a scaffold, but the quantity of interest is not target degradation alone. It is the difference between target degradation and non-target stability under a matched perturbation recipe.

Transferability and model-specific failures. Adversarial perturbations can transfer between models, but transfer is incomplete. Prior work has studied this phenomenon in benchmark perturbations (Alzahrani et al., 2024), adversarial data collection and dynamic benchmarking (Nie et al., 2020; Kiela et al., 2021), universal triggers (Wallace et al., 2019), and prompt-level attacks (Zhu et al., 2024; Wang

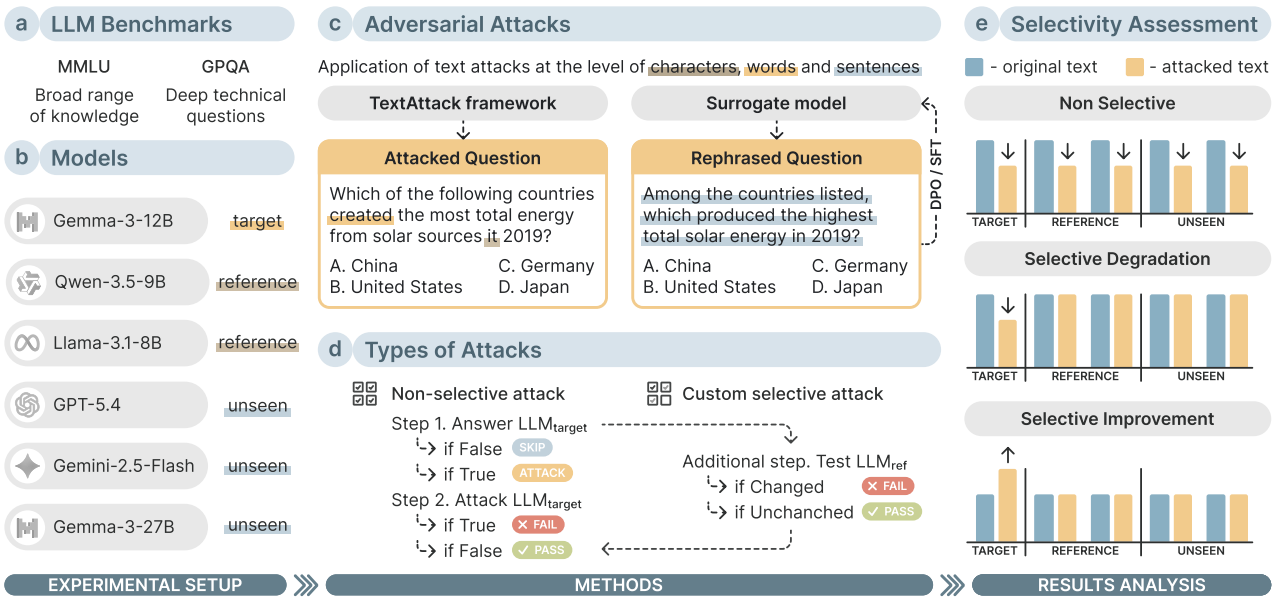

Figure 1. Selective-perturbation evaluation protocol. (a) We start from multiple-choice items from MMLU and GPQA. (b) Models are split into target, reference, and unseen roles. The target is the model whose answer we try to change; references constrain the search; unseen models are evaluated only after perturbations are found. (c) Candidate perturbations are produced with TextAttack transformations, and appendix experiments also consider a sentence-level surrogate rephrasing pipeline. In the main TextAttack experiments, only the question stem is modified. (d) Non-selective attacks require only a target change, while selective attacks additionally require reference predictions to remain unchanged. (e) We evaluate the resulting items by comparing original and perturbed accuracies for target, reference, and unseen models, separating broad degradation from target-specific degradation or improvement.

et al., 2024b; Biswas et al., 2025). In most of this work, transferability is treated as a property of attacks or a measure of attack strength. Our protocol uses incomplete transfer as a measurement signal. Reference models constrain search, while unseen models test whether perturbations remain selective beyond the models used to construct them. This makes the reference/unseen split central: reference stability alone shows constraint satisfaction, whereas unseen-model stability suggests the perturbation is not merely broad degradation that happens to satisfy the search constraint.

3. Method

Diagnostic setup. For each run, we choose a target model M_t, a set of reference models $\mathcal{M}_r$, and a set of unseen models $\mathcal{M}_u$. Reference models are used during perturbation search; unseen models are held out and evaluated only after perturbations are found.

For a model M, let $S_M(Q)$ be its accuracy on the original questions Q and $S_M(Q')$ its accuracy on the perturbed questions Q'. We report the accuracy drop

$$D_M = S_M(Q) - S_M(Q'), \tag{1}$$

so positive values indicate degradation. For plots, we also report the target-reference gap

$$\Delta_t = D_{M_t} - \frac{1}{|\mathcal{M}_r|} \sum_{M_i \in \mathcal{M}_r} D_{M_i}. \tag{2}$$

A larger Δ_t means that the perturbation affects the target more than the reference models.

Selective perturbations. For selective degradation, we search for a perturbed question q' such that the target changes from correct to incorrect, while reference models preserve their original correct prediction:

$$M_t(q) = y, \quad M_t(q') \neq y, \tag{3}$$

$$M_i(q') = M_i(q) = y, \quad M_i \in \mathcal{M}_r. \tag{4}$$

The non-selective baseline uses the same attack recipe but does not enforce the reference-preservation constraint. This baseline measures shared sensitivity: perturbations that make the question broadly harder or less stable across models. The selective variant conditions on reference stability and therefore isolates a target-specific component. We always report the non-selective baseline alongside the selective variant, using the same attack recipe, query budget, and item ordering. The comparison between the two is the basis of the diagnostic: if the target/reference gap is small in

the non-selective setting and grows under the selective constraint, the additional gap can be attributed to the constraint rather than to the underlying recipe.

Attack success rate (ASR) is computed with respect to the attempted examples for the evaluated model. For target models, success corresponds to changing an originally correct prediction into an incorrect one in the degradation setting. For non-target models, the same statistic measures how often the perturbation changes their originally correct answer, so lower non-target ASR indicates better selectivity.

Implementation. The main experiments use TextAttack with the BAE word-substitution recipe (Garg & Ramakrishnan, 2020). We add a custom reference-preservation constraint, `SelectivityMin`, which rejects a candidate perturbation if any reference model's prediction differs from its prediction on the original question. The attack modifies only the question stem; the system prompt, answer choices, and answer marker are fixed.

Prompting and scoring details. All examples are rendered as four-choice questions with a fixed instruction: the model is asked to return only the letter of the correct answer. The same prompt template is used for MMLU and GPQA. Each input contains the system instruction, the question stem, the four answer choices, and the answer marker; during attack search, only the question-stem field is exposed to TextAttack's modification operator. This restriction is important for interpretation. It rules out attacks that merely permute answer choices, alter the label format, or change the output marker, so a successful perturbation reflects sensitivity to the local wording of the question rather than to a change in the evaluation interface. Model outputs are scored by extracting the first occurrence of a valid answer letter. Calls are deterministic where supported, with temperature zero, a single completion, and a fixed seed.

Attack recipe and query accounting. The BAE recipe proposes word substitutions with a masked-language-model proposal distribution and a greedy word-importance search. We use 60 candidate substitutions per step, a Universal Sentence Encoder similarity threshold of 0.88, and the standard repeat and stopword pre-transformation constraints. The query budget is 600 target-model calls per item and 250 reference-model calls per item. Reference predictions on the original input are cached once per item; candidate perturbations are then rejected whenever any reference prediction changes. If the reference budget is exhausted, subsequent candidates for that item are rejected rather than treated as successful. This makes the selective condition conservative: failure to verify reference stability cannot create an apparently selective example.

Model access and role assignment. All main TextAttack runs are executed through the same API interface with provider fallbacks disabled. The cross-family experiments cycle Gemma-3-12B, Llama-3.1-8B, and Qwen3.5-9B through the target role, while the other two models form the reference set. The unseen set is evaluated only after attack search and includes a frontier model, an API-only Gemini model, and, where applicable, a larger same-family model. Thus the reported selectivity is tested in two ways: directly, by enforcing reference stability during search, and indirectly, by checking whether models that were not available to the search remain more stable than the target.

The main experiments use the full MMLU dev split (Hendrycks et al., 2020) and the first 50 items of GPQA Diamond (Rein et al., 2023). The target is cycled across Gemma-3-12B, Llama-3.1-8B, and Qwen3.5-9B; the other two models serve as references. Unseen models include GPT-5.4, Gemini-2.5-Flash, and a larger same-family model when applicable. Supporting experiments use Gemma-3-4B/12B/27B in an intra-family setting, Gemini-2.5-Flash as an API-only target, and a selective-improvement objective. Full prompts, model identifiers, attack parameters, query budgets, and scoring details are listed in Appendix A. The 50-item pilot corresponding to the main MMLU experiment is reported separately in Appendix B. Each successful selective edit is a single word-level substitution under a Universal Sentence Encoder similarity threshold of 0.88; we report perturbation-magnitude statistics in Appendix A.

4. Results

4.1. Full MMLU dev split

The main experiment uses the full MMLU dev split and cycles Gemma-3-12B, Llama-3.1-8B, and Qwen3.5-9B as targets. For each target, the other two models serve as references during search, while GPT-5.4, Gemini-2.5-Flash, and a larger same-family model are evaluated only after perturbations are found.

Table 1 reports the full per-model breakdown. Non-selective perturbations produce large target drops, but they also degrade non-target models. This is the shared-sensitivity component: many local edits make a question broadly harder or less stable across models. The selective constraint changes this pattern. Across the three targets, the target still drops by 0.38–0.44, while the maximum reference drop is 0.04. Held-out unseen models also remain much more stable than the target, with drops of at most 0.10.

The gap between selective and non-selective attacks is central to the interpretation. The non-selective attack shows that the base perturbation mechanism can degrade many models at once. The selective attack shows that a sizeable part of the degradation remains after conditioning on reference-model

Table 1. Selectivity evaluation of the BAE attack on the full MMLU dev split (285 questions) for all three cross-family targets. Each block groups a distinct target. **Selective**: attack uses our custom selectivity constraint; **Non-selective**: standard BAE. Models marked with † are targets; models marked with ‡ are in the optimization reference set; all others are *unseen* (not used during adversarial optimization). **Calls**: number of model queries during optimization (unseen models receive one call per question). **ASR**: attack success rate. The selectivity gap observed on the 50-question pilot (Table 8) is preserved at full-split scale: target drops of 0.38 to 0.44 under the selective constraint with at most 0.04 drop on reference models and at most 0.10 drop on unseen models.

Target	Evaluated model	Orig. Acc.	Selective			Non-selective		
			Calls	Acc.	ASR	Calls	Acc.	ASR
Gemma-3-12B	Gemma-3-12B †	0.73	33835	$0.29_{-0.44\downarrow}$	0.59	93489	$0.12_{-0.61\downarrow}$	0.84
	Qwen3.5-9B ‡	0.79	34564	$0.78_{-0.02\downarrow}$	0.04	460	$0.62_{-0.17\downarrow}$	0.22
	Llama-3.1-8B ‡	0.66	32076	$0.64_{-0.02\downarrow}$	0.05	460	$0.48_{-0.18\downarrow}$	0.31
	GPT-5.4	0.90	408	$0.86_{-0.05\downarrow}$	0.05	460	$0.71_{-0.19\downarrow}$	0.21
	Gemini-2.5-Flash	0.87	408	$0.82_{-0.05\downarrow}$	0.07	460	$0.68_{-0.19\downarrow}$	0.23
	Gemma-3-27B	0.74	408	$0.67_{-0.07\downarrow}$	0.10	460	$0.49_{-0.25\downarrow}$	0.35
Llama-3.1-8B	Llama-3.1-8B †	0.68	30256	$0.30_{-0.38\downarrow}$	0.56	69030	$0.13_{-0.55\downarrow}$	0.81
	Gemma-3-12B ‡	0.74	31060	$0.72_{-0.02\downarrow}$	0.04	444	$0.56_{-0.18\downarrow}$	0.26
	Qwen3.5-9B ‡	0.81	28201	$0.79_{-0.01\downarrow}$	0.03	444	$0.64_{-0.17\downarrow}$	0.21
	GPT-5.4	0.90	394	$0.87_{-0.03\downarrow}$	0.04	444	$0.76_{-0.14\downarrow}$	0.16
	Gemini-2.5-Flash	0.87	394	$0.85_{-0.02\downarrow}$	0.03	444	$0.73_{-0.14\downarrow}$	0.17
	Llama-3.1-70B	0.77	394	$0.73_{-0.04\downarrow}$	0.06	444	$0.63_{-0.14\downarrow}$	0.20
Qwen3.5-9B	Qwen3.5-9B †	0.79	38356	$0.41_{-0.39\downarrow}$	0.49	91781	$0.14_{-0.65\downarrow}$	0.82
	Gemma-3-12B ‡	0.73	40226	$0.69_{-0.04\downarrow}$	0.06	470	$0.46_{-0.27\downarrow}$	0.37
	Llama-3.1-8B ‡	0.67	36799	$0.65_{-0.01\downarrow}$	0.04	470	$0.49_{-0.17\downarrow}$	0.29
	GPT-5.4	0.90	395	$0.83_{-0.07\downarrow}$	0.09	470	$0.71_{-0.19\downarrow}$	0.21
	Gemini-2.5-Flash	0.88	395	$0.78_{-0.10\downarrow}$	0.12	470	$0.65_{-0.23\downarrow}$	0.27
	Qwen3.5-27B	0.87	395	$0.80_{-0.07\downarrow}$	0.09	470	$0.64_{-0.23\downarrow}$	0.27

stability. The unseen-model columns make the result less dependent on the two references used during search: in all three target settings, unseen drops are much smaller than target drops. The 50-item pilot version of the same experiment is reported in Appendix B.

4.2. GPQA Diamond

To check that the pattern is not specific to MMLU, we repeat the cross-family setup on the first 50 questions of GPQA Diamond (Rein et al., 2023), a harder benchmark whose baseline accuracies are substantially lower. Results are reported in Table 2.

The absolute target drops are smaller than on MMLU, which is expected given the lower original accuracies. The relevant pattern remains: under the selective constraint, the target drops by 0.16–0.20, while reference drops are at most 0.02 and unseen drops at most 0.05. This suggests that the shared-versus-target-specific decomposition is not specific to MMLU, although the absolute magnitudes depend on benchmark difficulty.

Because this experiment uses only the first 50 GPQA Diamond items, we treat it as a cross-benchmark check rather than a full-scale estimate of GPQA selectivity. Its role is to test whether the qualitative separation between shared and target-specific sensitivity persists when the baseline task is much harder and original accuracies are lower.

4.3. Supporting settings

Figure 2 summarizes three settings: the full MMLU cross-family experiment, the Gemma intra-family experiment, and GPQA Diamond. The detailed side-experiment tables are in Appendix C, Appendix D, Appendix E, and Appendix F.

Within the Gemma family, selectivity is weaker than in the cross-family setting, but target drops of 0.33–0.44 remain larger than same-family reference drops. With Gemini-2.5-Flash as an API-only target, the selective attack produces a 0.41 target drop while both reference models remain unchanged in accuracy. The selective-improvement setting shows the same idea in the opposite direction: local edits can selectively raise target accuracy while mostly preserving reference behavior. Character-level attacks produce weaker and less informative selectivity gaps than word-level substitutions.

The character-level comparison is useful as an ablation on the perturbation class. DeepWordBug changes the surface form of tokens through small character edits, whereas BAE replaces words with masked-language-model proposals subject to semantic and syntactic constraints. In our 50-item MMLU comparison, the character-level attacks still sometimes change the target answer, but the target drops are smaller and the gap between selective and non-selective conditions is less stable. This suggests that the strongest diagnostic signal in our setting comes from substitutions

Table 2. Selectivity evaluation of the BAE attack on the first 50 questions of GPQA Diamond for all three cross-family targets. Each block groups a distinct target. **Selective**: attack uses our custom selectivity constraint; **Non-selective**: standard BAE. Models marked with † are targets; models marked with ‡ are in the optimization reference set; all others are *unseen* (not used during adversarial optimization). **Calls**: number of model queries during optimization (unseen models receive one call per question). **ASR**: attack success rate. The selectivity gap is preserved on this harder benchmark, with reference and unseen models exhibiting near-zero drops under the selective constraint despite substantial drops on the targets.

Target	Evaluated model	Orig. Acc.	Selective			Non-selective		
			Calls	Acc.	ASR	Calls	Acc.	ASR
Gemma-3-12B	Gemma-3-12B †	0.27	2614	$0.10_{-0.17\downarrow}$	0.64	7911	$0.02_{-0.25\downarrow}$	0.92
	Qwen3.5-9B ‡	0.37	2044	$0.38_{+0.01\uparrow}$	0.00	62	$0.34_{-0.03\downarrow}$	0.06
	Llama-3.1-8B ‡	0.32	1942	$0.32_{+0.00}$	0.00	62	$0.30_{-0.02\downarrow}$	0.06
	GPT-5.4	0.53	59	$0.54_{+0.01\uparrow}$	0.00	62	$0.48_{-0.05\downarrow}$	0.08
	Gemini-2.5-Flash	0.35	59	$0.32_{-0.03\downarrow}$	0.00	62	$0.34_{-0.01\downarrow}$	0.11
	Gemma-3-27B	0.33	59	$0.34_{+0.01\uparrow}$	0.00	62	$0.30_{-0.03\downarrow}$	0.13
Llama-3.1-8B	Llama-3.1-8B †	0.30	3056	$0.14_{-0.16\downarrow}$	0.53	6375	$0.06_{-0.24\downarrow}$	0.80
	Gemma-3-12B ‡	0.24	2626	$0.24_{+0.00}$	0.00	62	$0.22_{-0.02\downarrow}$	0.08
	Qwen3.5-9B ‡	0.39	2369	$0.38_{-0.01\downarrow}$	0.00	62	$0.38_{-0.01\downarrow}$	0.05
	GPT-5.4	0.58	58	$0.56_{-0.02\downarrow}$	0.03	62	$0.56_{-0.02\downarrow}$	0.07
	Gemini-2.5-Flash	0.33	58	$0.36_{+0.03\uparrow}$	0.06	62	$0.30_{-0.03\downarrow}$	0.13
	Llama-3.1-70B	0.50	58	$0.48_{-0.02\downarrow}$	0.04	62	$0.48_{-0.02\downarrow}$	0.04
Qwen3.5-9B	Qwen3.5-9B †	0.38	3440	$0.18_{-0.20\downarrow}$	0.53	13529	$0.12_{-0.26\downarrow}$	0.68
	Gemma-3-12B ‡	0.28	3419	$0.26_{-0.02\downarrow}$	0.07	63	$0.26_{-0.02\downarrow}$	0.07
	Llama-3.1-8B ‡	0.32	2988	$0.32_{+0.00}$	0.00	63	$0.32_{+0.00}$	0.00
	GPT-5.4	0.58	60	$0.60_{+0.02\uparrow}$	0.00	63	$0.50_{-0.08\downarrow}$	0.11
	Gemini-2.5-Flash	0.33	60	$0.28_{-0.05\downarrow}$	0.13	63	$0.32_{-0.01\downarrow}$	0.06
	Qwen3.5-27B	0.49	60	$0.52_{+0.03\uparrow}$	0.00	63	$0.44_{-0.05\downarrow}$	0.13

that preserve a readable question while changing the lexical cues available to the model. It also helps rule out a trivial explanation in which any perturbation budget would automatically yield the same selective effect.

The supporting experiments therefore play different roles. The intra-family study asks whether selectivity persists when target and references share more training and architectural structure; the frontier-target study asks whether the method requires open-weight access; the improvement study checks whether the same search logic can move accuracy upward rather than only downward; and the character-level study tests whether the effect depends on the word-level substitution space. Across these variants, the magnitude changes, but the diagnostic distinction between shared and target-specific sensitivity remains visible.

Taken together, these experiments show variation rather than a single universal effect size. The model-specific component is largest in the cross-family MMLU setting, smaller on GPQA where baseline accuracies leave less headroom, and weaker within one model family. This is consistent with the diagnostic view: the protocol measures how much of a model's sensitivity is shared with its references and how much is target-specific.

5. Qualitative Analysis

To connect the aggregate drops to concrete changes in the questions, we manually inspected successful selective perturbations for two targets in the cross-family experiment: Gemma-3-12B and Qwen3.5-9B. This is a descriptive analysis rather than a complete taxonomy of either model. Representative examples are shown in Table 3; extended notes are in Appendix G.

The inspected perturbations suggest different local failure profiles. Qwen3.5-9B is often affected by coarse substitutions that remove or corrupt a domain anchor, such as replacing a technical term with a generic or incompatible word. These edits can look semantically odd to a human reader, but the reference models often preserve the original answer, apparently using the remaining context and answer choices.

Gemma-3-12B is more often affected by milder edits: near-synonyms, register shifts, casing changes, or small changes in function words. These are closer to ordinary wording variation, although not all of them are strict paraphrases. The point is not that one target is globally more robust than the other, but that their target-specific components have different linguistic content.

These examples also clarify why we treat selective perturbations as a diagnostic rather than as a claim about semantic equivalence. Some successful edits are close to benign

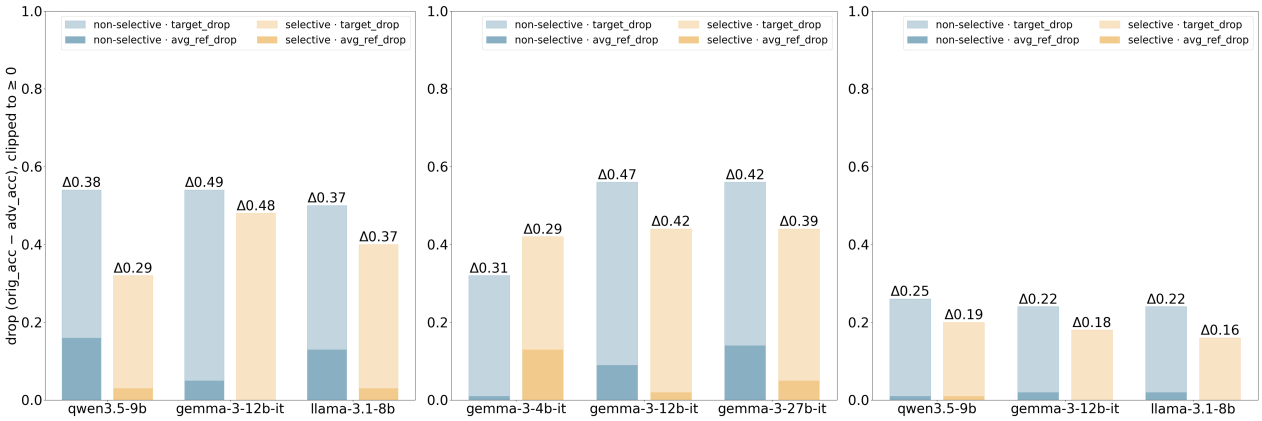

Figure 2. Target and reference drops across the main and supporting settings. Panels show the full MMLU cross-family experiment, the Gemma intra-family experiment, and GPQA Diamond. The plot legend distinguishes attack condition and evaluated group. Labels report the target-reference drop gap. Selective perturbations reduce collateral reference degradation while preserving a target-specific component.

benchmark variation: for example, replacing a technical noun with a near-synonym or changing the register of a short phrase. Other edits are less faithful because they damage a domain anchor while leaving enough context for other models to answer correctly. Both types are informative, but they answer different questions. The former indicates brittleness to wording that a human annotator might plausibly accept; the latter indicates which lexical anchors a model overuses or fails to recover from. Separating these cases is important for future use of the protocol, because the same aggregate target drop can be produced by qualitatively different perturbation mixtures.

This qualitative difference helps interpret the quantitative results. Selective perturbations do not merely identify that a model can be made wrong. They show what kinds of local changes are disproportionately associated with one model's failures while other models remain stable.

6. Discussion

Shared and model-specific sensitivity. The central observation is that local benchmark perturbations contain both shared and model-specific components. Non-selective attacks expose the shared component: many edits degrade several models at once. Selective attacks condition on reference-model stability and reveal a remaining target-specific component. This distinction is useful for interpreting benchmark comparisons, especially when two models have similar aggregate accuracy.

Unseen models are necessary. Reference models are part of the construction, so their stability alone is not enough to support a selectivity claim. The held-out unseen models provide the more informative test. In the full MMLU experiment, unseen-model drops remain much smaller than target drops across all three targets, suggesting that the perturbations are not simply broad degradations that happen to satisfy the reference constraint.

Ranking is not invariant to local wording. A concrete consequence of the main result is that the order of two similar-performing models on a benchmark can be reversed by local edits. On the full MMLU dev split, Qwen3.5-9B and Gemma-3-12B differ by 6 points in original accuracy (0.79 versus 0.73). Under selective edits with Qwen as target, Qwen drops to 0.41 while Gemma stays at 0.69. The two models swap places in the ranking, even though the answer choices, prompt format, and benchmark items are unchanged. This concerns the comparison itself, not only whether a model can be made wrong.

A practical implication is that benchmark reports should not treat a single accuracy gap as self-explanatory. When two models are close in aggregate score, selective perturbation slices can indicate whether the apparent gap is stable across local wording changes or depends on idiosyncratic sensitivity. This does not replace aggregate benchmarks, but adds a diagnostic layer for close comparisons.

Why the non-selective baseline matters. The selective result would be hard to interpret without the matched non-selective attack. A large target drop alone could mean that the target is fragile, but it could also mean that the perturbation recipe simply makes many benchmark items invalid or harder for all models. The non-selective baseline estimates this shared component under the same recipe and item ordering. The selective condition then asks how much degradation remains after explicitly conditioning on reference stability. In this sense, the method is not an attack leaderboard. It is a paired diagnostic: the difference between the two conditions is the evidence that the observed

Table 3. Representative successful selective attacks (BAE with custom constraint) against Gemma-3-12B (top) and Qwen3.5-9B (bottom) on the MMLU dev split. The substituted token is shown in **bold**. In all cases shown, the target answers correctly on the original and incorrectly on the perturbed input, while both reference models answer correctly on both versions.

Pattern	Original (excerpt)	Perturbed (excerpt)
Target: Gemma-3-12B — failures on near-paraphrases		
Near-synonym	… telescope with an *aperture* of 50 cm.	… telescope with an **opening** of 50 cm.
Near-synonym	… how many attempts …, *according to* the medical knowledge …	… how many attempts …, **pursuant to** the medical knowledge …
Casing only	*Consider* a computer design …	**consider** a computer design …
Polarity flip	The *presence* of homologous structures … indicates …	The **absence** of homologous structures … indicates …
Target: Qwen3.5-9B — failures on semantically incongruent substitutions		
Domain-anchor removal	What is the *embryological* origin of the hyoid bone?	What is the **actual** origin of the hyoid bone?
Domain-anchor removal	… accurate statement concerning *arthropods*?	… accurate statement concerning **diversity**?
Incongruent term	Why is Mars *red*?	Why is Mars **purple**?
Incongruent term	Every function from a *finite* set onto itself …	Every function from a **continuous** set onto itself …

failure mode is more target-specific than the perturbation recipe by itself.

Reference choice affects the question being asked. The reference set defines the comparison class. If the references are close variants of the target, as in the Gemma intra-family experiment, the constraint is stricter with respect to family-shared behavior and the selectivity gap becomes smaller. If the references come from different model families, the protocol is better suited for detecting model-specific sensitivities that do not transfer broadly. Neither choice is universally correct. For leaderboard interpretation, references should be the models whose comparison with the target is substantively important. For robustness analysis, references can be chosen to control for a hypothesized nuisance factor, such as prompt format sensitivity or shared benchmark familiarity.

Semantic validity is a spectrum. The perturbations found by the search are local edits, not a curated set of human-validated paraphrases. Some preserve meaning closely; others corrupt a domain term or make the question less coherent. We treat them as interventions on model behavior rather than as a pure paraphrase benchmark. This choice is appropriate for diagnosing which local changes expose model-specific failures, but it also limits how the results should be interpreted.

7. Limitations

Our perturbation class is limited: the main experiments use word-level TextAttack substitutions, with character-level and sentence-level variants reported only as supporting studies. The perturbations are not all meaning-preserving, so the protocol is different from a human-validated paraphrase robustness benchmark. The qualitative analysis covers two targets in detail and should be treated as descriptive, not as a mechanistic explanation of model behavior. The experiments use widely studied benchmarks, so pretraining contamination may affect both original accuracy and perturbation sensitivity. We do not run multi-seed reruns of the search procedure, so per-cell numbers are single-seed estimates. Finally, all evaluated frontier models are accessed through APIs and may change over time.

A second limitation is that the protocol measures selectivity relative to the chosen references, not an absolute property of a perturbation. A perturbation that is selective against two reference models may still transfer to a different unseen model with similar inductive biases. We partially address this by evaluating held-out models after search, but this is still a finite sample of possible non-target systems. The method should therefore be interpreted as evidence about a comparison set rather than as a guarantee that an edit is uniquely harmful to one model.

8. Conclusion

Selective perturbations can serve as a diagnostic for benchmark-based LLM comparisons. The main result is not only that such perturbations can be constructed, but that they separate two kinds of benchmark sensitivity. Non-selective perturbations often reveal shared sensitivity, degrading several models at once. With a reference-preserving constraint, a large target-specific component remains: on the full MMLU dev split, target models drop by 0.38–0.44 while reference and unseen models remain much more stable. The same qualitative pattern appears on GPQA Diamond and in

supporting settings, although the effect size changes. Manual inspection suggests that the target-specific component is structured: different models fail under different kinds of local edits.

Software and Data

We provide a repository with the code, attack configurations, model outputs, and scripts used to produce the results:

https://github.com/ITMO-NSS-team/llm-selective-perturbations

Acknowledgements

This work supported by the Ministry of Economic Development of the Russian Federation (IGK 000000C313925P4C0002), agreement No139-15-2025-010

Impact Statement

This paper presents a diagnostic method for understanding how benchmark-based comparisons between language models depend on question wording. The intended use is to make model comparisons more reliable, by separating broad benchmark sensitivity from model-specific sensitivity. The same technique could in principle be used to construct perturbations that bias a benchmark comparison against a specific model. We reduce this risk by framing the method as a paired diagnostic that always reports reference and unseen-model behavior alongside the target, and by releasing per-item outputs so that any reported comparison can be audited. We see no other societal consequences that require specific discussion beyond those already common to work on language-model evaluation.

References

Alzahrani, N., Alyahya, H., Alnumay, Y., Alrashed, S., Alsubaie, S., Almushayqih, Y., Mirza, F., Alotaibi, N., Al-Twairesh, N., Alowisheq, A., et al. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13787–13805, 2024.

Biderman, S., Schoelkopf, H., Sutawika, L., Gao, L., Tow, J., Abbasi, B., Aji, A. F., Ammanamanchi, P. S., Black, S., Clive, J., et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024.

Biswas, S., Nishino, M., Chacko, S. J., and Liu, X. Universal and transferable adversarial attack on large language models using exponentiated gradient descent. *arXiv preprint arXiv:2508.14853*, 2025.

Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. 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)*, pp. 31–36, 2018.

Gao, J., Lanchantin, J., Soffa, M. L., and Qi, Y. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE security and privacy workshops (SPW)*, pp. 50–56. IEEE, 2018.

Garg, S. and Ramakrishnan, G. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6174–6181, 2020.

Gema, A. P., Leang, J. O. J., Hong, G., Devoto, A., Mancino, A. C. M., Saxena, R., He, X., Zhao, Y., Du, X., Madani, M. R. G., et al. Are we done with mmlu? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5069–5096, 2025.

Goel, K., Rajani, N. F., Vig, J., Taschdjian, Z., Bansal, M., and Ré, C. Robustness gym: Unifying the nlp evaluation landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pp. 42–55, 2021.

Gui, T., Wang, X., Zhang, Q., Liu, Q., Zou, Y., Zhou, X., Zheng, R., Zhang, C., Wu, Q., Ye, J., et al. Textflint: Unified multilingual robustness evaluation toolkit for natural language processing. *arXiv preprint arXiv:2103.11441*, 2021.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Iyyer, M., Wieting, J., Gimpel, K., and Zettlemoyer, L. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1875–1885, 2018.

Jin, D., Jin, Z., Zhou, J. T., and Szolovits, P. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8018–8025, 2020.

Kiela, D., Bartolo, M., Nie, Y., Kaushik, D., Geiger, A., Wu, Z., Vidgen, B., Prasad, G., Singh, A., Ringshia, P., 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*, pp. 4110–4124, 2021.

Li, L., Ma, R., Guo, Q., Xue, X., and Qiu, X. Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6193–6202, 2020.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Morris, J., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., and Qi, Y. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations*, pp. 119–126, 2020.

Nalbandyan, G., Shahbazyan, R., and Bakhturina, E. Score: Systematic consistency and robustness evaluation for large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pp. 470–484, 2025.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4885–4901, 2020.

Qi, F., Chen, Y., Zhang, X., Li, M., Liu, Z., and Sun, M. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 4569–4580, 2021.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Ren, S., Deng, Y., He, K., and Che, W. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 1085–1097, 2019.

Ribeiro, M. T., Singh, S., and Guestrin, C. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (volume 1: long papers)*, pp. 856–865, 2018.

Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4902–4912, 2020.

Sclar, M., Choi, Y., Tsvetkov, Y., and Suhr, A. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, volume 2024, pp. 25055–25083, 2024.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on machine learning research*, 2023.

Wallace, E., Feng, S., Kandpal, N., Gardner, M., and Singh, S. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2153–2162, 2019.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. 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*, pp. 353–355, 2018.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Wang, B., Xu, C., Wang, S., Gan, Z., Cheng, Y., Gao, J., Awadallah, A. H., and Li, B. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. *arXiv preprint arXiv:2111.02840*, 2021.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024a.

Wang, Z., Tu, H., Mei, J., Zhao, B., Wang, Y., and Xie, C. Attngcg: Enhancing jailbreaking attacks on llms with

attention manipulation. *arXiv preprint arXiv:2410.09040*, 2024b.

Xu, C., Guan, S., Greene, D., Kechadi, M., et al. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244*, 2024.

Zeng, G., Qi, F., Zhou, Q., Zhang, T., Ma, Z., Hou, B., Zang, Y., Liu, Z., and Sun, M. 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*, pp. 363–371, 2021.

Zhao, Q., Huang, Y., Lv, T., Cui, L., Sun, Q., Mao, S., Zhang, X., Xin, Y., Yin, Q., Li, S., et al. Mmlu-cf: A contamination-free multi-task language understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13371–13391, 2025.

Zheng, C., Zhou, H., Meng, F., Zhou, J., and Huang, M. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations*, volume 2024, pp. 19426–19454, 2024.

Zhou, K., Zhu, Y., Chen, Z., Chen, W., Zhao, W. X., Chen, X., Lin, Y., Wen, J.-R., and Han, J. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*, 2023.

Zhu, K., Zhao, Q., Chen, H., Wang, J., and Xie, X. Promptbench: A unified library for evaluation of large language models. *Journal of Machine Learning Research*, 25(254): 1–22, 2024.

A. Reproducibility Details

This appendix lists the experimental details needed to interpret the results in the main paper. The released artifact contains per-example outputs, successful perturbations, aggregate selectivity tables, and scripts for reproducing the plots. All main TextAttack experiments are run through the OpenRouter API.

A.1. Datasets and prompts

Table 4 summarizes the dataset splits, item counts, ordering, and choice formatting used in the MMLU and GPQA experiments.

Table 4. Dataset settings. MMLU is used for the main full-split experiment and for 50-item supporting studies. GPQA Diamond is used in the cross-benchmark experiment.

Field	MMLU	GPQA Diamond
HuggingFace ID	`cais/mmlu`	`Idavidrein/gpqa`
Configuration	all subjects	`gpqa_diamond`
Split	dev	train
Items in split	285	198
Items used in 50-item runs	first 50	first 50
Question ordering	as returned by HF	as returned by HF
Choices shuffled	no	yes, seed 42
Choice format	A/B/C/D	A/B/C/D

The prompt template is identical across datasets:

```
You are a helpful assistant.  Respond with only the letter of the correct answer
(A, B, C, or D).
Question:  <q>
Choices:
A. <c0>
B. <c1>
C. <c2>
D. <c3>
Answer:
```

The input columns visible to TextAttack are `system_prompt`, `question`, `choices`, and `answer_word`. We use `InputColumnModification` to restrict attacks to the `question` column. The system prompt, answer choices, and answer marker are fixed in the TextAttack experiments.

A.2. Attack settings

Table 5 lists the TextAttack recipes, constraints, and query budgets used for the word-level and character-level experiments.

The selective constraint rejects a candidate if any reference model's prediction differs from its prediction on the original question. Reference predictions on original questions are cached once per item. When the per-question reference budget is exhausted, all further candidates for that question are rejected.

The successful selective perturbations are small under standard text-edit measures. The BAE recipe applies word-level substitutions under a Universal Sentence Encoder similarity threshold of 0.88. As a representative example, across the 24 successful selective perturbations against Gemma-3-12B in the cross-family MMLU experiment, the question stem averages 24 words, almost all edits change a single word, and the mean character-level Levenshtein distance is 6.5 (range 1 to 11). This keeps the effective perturbation budget well inside the range used for semantics-preserving word-level attacks, indicating that the target-specific drops in the main paper come from minimal edits rather than large rewrites.

A.3. Inference settings and model identifiers

Table 6 reports the shared inference settings for OpenRouter calls, and Table 7 lists the model identifiers and roles used across experiments.

Table 5. Attack settings used in the main and appendix experiments. BAE is the main word-level recipe. DeepWordBug is included as a character-level comparison in Appendix F.

Field	BAE	DeepWordBug
Transformation	`WordSwapMaskedLM(method="bae")`	character edits
Candidate count	60	not applicable
Masked LM	`bert-base-uncased`	not applicable
Search method	`GreedyWordSwapWIR("delete")`	`GreedyWordSwapWIR()`
Goal function	untargeted classification	untargeted classification
Improvement goal	untargeted improvement variant	same
Pre-transformation constraints	repeat, stopword	repeat, stopword
Semantic similarity	USE threshold 0.88	none
POS constraint	verb/noun swaps allowed	none
Edit-distance constraint	none	Levenshtein $\epsilon = 50$
Target query budget	600 per item	600 per item
Reference query budget	250 per item	250 per item

Table 6. Inference settings for OpenRouter calls.

Field	Value
Temperature	0
Max tokens	16
Number of completions	1
Seed	42
Reasoning	disabled where supported
Provider fallbacks	disabled
Timeout	60 seconds
Retry policy	up to 15 attempts for transient API errors
Letter extraction	first match of `[ABCD]`
No-letter output	uniform over A/B/C/D, argmax resolves to A
Observed frequency of no-letter outputs	less than 1% of calls

B. Pilot 50-Item Cross-Family Study

Before running the full MMLU dev split, we ran the same cross-family setup on the first 50 MMLU dev questions. Table 8 reports the pilot results. The pilot shows the same qualitative structure as the full-split result, although individual effect sizes differ.

C. Gemma Intra-Family Study

The intra-family experiment tests whether selectivity remains visible when the target and reference models come from the same model family. Table 9 reports the Gemma-3-4B/12B/27B results.

Selectivity is weaker within the Gemma family than in the main cross-family MMLU setting, but it does not disappear. The selective target drops remain substantial, while same-family reference drops are smaller.

D. Frontier Target Study

This experiment uses Gemini-2.5-Flash as the target and Qwen3.5-9B and Llama-3.1-8B as references. Table 10 reports the frontier-target results.

The selective constraint produces a large drop on Gemini-2.5-Flash while leaving both reference models unchanged in accuracy. This suggests that the protocol does not rely on model internals or open-weight access.

E. Selective Improvement

The improvement setting reverses the direction of the objective: perturbations are selected to improve the target while preserving non-target behavior. Table 11 reports the selective-improvement results.

Table 7. OpenRouter model identifiers.

Display name	OpenRouter identifier	Role
Qwen3.5-9B	`qwen/qwen3.5-9b`	target/reference
Gemma-3-12B	`google/gemma-3-12b-it`	target/reference
Llama-3.1-8B	`meta-llama/llama-3.1-8b-instruct`	target/reference
Gemma-3-4B	`google/gemma-3-4b-it`	intra-family target/reference
Gemma-3-27B	`google/gemma-3-27b-it`	unseen / intra-family target/reference
Gemini-2.5-Flash	`google/gemini-2.5-flash`	target/unseen
GPT-5.4	`openai/gpt-5.4`	unseen
Llama-3.1-70B	`meta-llama/llama-3.1-70b-instruct`	unseen
Qwen3.5-27B	`qwen/qwen3.5-27b`	unseen

Table 8. Selectivity evaluation of the BAE attack on 50 MMLU dev-split questions. Each block groups a distinct target. **Selective**: attack uses our custom selectivity constraint; **Non-selective**: standard BAE. Models marked with † are targets; models marked with ‡ are in the optimization reference set; all others are *unseen* (not used during adversarial optimization). **Calls**: number of model queries during optimization (unseen models receive one call per question). **ASR**: attack success rate. For targets, lower Acc.$_\Delta$ and higher ASR indicate a stronger attack; for non-targets, Δ close to zero indicates better selectivity. Stronger selectivity manifests as a large target/non-target gap in Acc.$_\Delta$ and ASR.

Target	Evaluated model	Orig. Acc.	Selective			Non-selective		
			Calls	Acc.	ASR	Calls	Acc.	ASR
Gemma-3-12B	Gemma-3-12B †	0.65	4548	$0.18_{-0.47\downarrow}$	0.73	5490	$0.10_{-0.55\downarrow}$	0.84
	Qwen3.5-9B ‡	0.73	4482	$0.76_{+0.03\uparrow}$	0.00	77	$0.74_{+0.01\uparrow}$	0.05
	Llama-3.1-8B ‡	0.61	4343	$0.60_{-0.01\downarrow}$	0.10	77	$0.50_{-0.11\downarrow}$	0.27
	GPT-5.4	0.89	74	$0.82_{-0.07\downarrow}$	0.07	77	$0.86_{-0.03\downarrow}$	0.04
	Gemini-2.5-Flash	0.88	74	$0.78_{-0.10\downarrow}$	0.11	77	$0.80_{-0.08\downarrow}$	0.09
	Gemma-3-27B	0.62	74	$0.54_{-0.08\downarrow}$	0.13	77	$0.50_{-0.12\downarrow}$	0.19
Llama-3.1-8B	Llama-3.1-8B †	0.58	3553	$0.16_{-0.42\downarrow}$	0.71	5930	$0.10_{-0.50\downarrow}$	0.83
	Gemma-3-12B ‡	0.70	3814	$0.66_{-0.04\downarrow}$	0.06	75	$0.60_{-0.10\downarrow}$	0.17
	Qwen3.5-9B ‡	0.73	3226	$0.70_{-0.03\downarrow}$	0.03	75	$0.58_{-0.16\downarrow}$	0.22
	GPT-5.4	0.86	70	$0.82_{-0.04\downarrow}$	0.00	75	$0.78_{-0.08\downarrow}$	0.13
	Gemini-2.5-Flash	0.82	70	$0.82_{+0.00}$	0.00	75	$0.70_{-0.12\downarrow}$	0.17
	Llama-3.1-70B	0.76	70	$0.70_{-0.06\downarrow}$	0.08	75	$0.66_{-0.10\downarrow}$	0.13
Qwen3.5-9B	Qwen3.5-9B †	0.72	6159	$0.40_{-0.32\downarrow}$	0.44	8872	$0.18_{-0.54\downarrow}$	0.75
	Gemma-3-12B ‡	0.69	6772	$0.64_{-0.05\downarrow}$	0.06	77	$0.52_{-0.17\downarrow}$	0.26
	Llama-3.1-8B ‡	0.62	6100	$0.60_{-0.02\downarrow}$	0.03	77	$0.48_{-0.14\downarrow}$	0.26
	GPT-5.4	0.89	66	$0.88_{-0.01\downarrow}$	0.07	77	$0.74_{-0.15\downarrow}$	0.16
	Gemini-2.5-Flash	0.88	66	$0.82_{-0.06\downarrow}$	0.09	77	$0.66_{-0.22\downarrow}$	0.25
	Qwen3.5-27B	0.80	66	$0.76_{-0.04\downarrow}$	0.05	77	$0.66_{-0.14\downarrow}$	0.18

Selective improvement is weaker than selective degradation in some settings, but it is reproducible across targets. This supports the view that the protocol probes directional model-specific sensitivity.

F. Character-Level Attacks

We evaluate DeepWordBug (Gao et al., 2018) as a character-level comparison. Table 12 reports the character-level results.

DeepWordBug produces smaller target drops than BAE and a smaller difference between selective and non-selective settings, suggesting that word-level substitutions are a stronger probe in this setup.

G. Extended Qualitative Analysis

The qualitative examples in Table 3 summarize successful selective perturbations for Gemma-3-12B and Qwen3.5-9B. The analysis is descriptive: it connects aggregate drops to concrete local changes in question wording.

Qwen3.5-9B. Qwen3.5-9B is often affected by coarse substitutions that remove or corrupt a domain anchor. Examples include replacing technical terms with generic or incompatible words. These changes can make the question semantically

Table 9. Intra-family selectivity evaluation of the BAE attack on 50 MMLU dev-split questions across Gemma-3 models of different scales. Each block groups a distinct target. **Selective**: attack uses our custom selectivity constraint; **Non-selective**: standard BAE. Models marked with † are targets; models marked with ‡ are in the optimization reference set (same-family models of different size); all others are *unseen* (not used during adversarial optimization). **Calls**: number of model queries during optimization (unseen models receive one call per question). **ASR**: attack success rate. For targets, lower Acc._{Δ} and higher ASR indicate a stronger attack; for non-targets, Δ close to zero indicates better selectivity.

Target	Evaluated model	Orig. Acc.	Selective			Non-selective		
			Calls	Acc.	ASR	Calls	Acc.	ASR
Gemma-3-27B	Gemma-3-27B †	0.64	4176	$0.20_{-0.44\downarrow}$	0.69	4705	$0.08_{-0.56\downarrow}$	0.88
	Gemma-3-12B ‡	0.70	4601	$0.62_{-0.08\downarrow}$	0.11	81	$0.52_{-0.18\downarrow}$	0.26
	Gemma-3-4B ‡	0.53	4129	$0.50_{-0.03\downarrow}$	0.04	81	$0.44_{-0.09\downarrow}$	0.19
	GPT-5.4	0.86	74	$0.80_{-0.06\downarrow}$	0.07	81	$0.72_{-0.14\downarrow}$	0.16
	Gemini-2.5-Flash	0.87	74	$0.82_{-0.05\downarrow}$	0.05	81	$0.78_{-0.09\downarrow}$	0.11
	Qwen3.5-9B	0.73	74	$0.66_{-0.07\downarrow}$	0.08	81	$0.60_{-0.13\downarrow}$	0.19
	Llama-3.1-8B	0.61	74	$0.58_{-0.03\downarrow}$	0.07	81	$0.50_{-0.11\downarrow}$	0.19
Gemma-3-12B	Gemma-3-12B †	0.67	4594	$0.24_{-0.43\downarrow}$	0.65	5634	$0.10_{-0.57\downarrow}$	0.85
	Gemma-3-27B ‡	0.64	5069	$0.64_{+0.00}$	0.00	78	$0.52_{-0.12\downarrow}$	0.19
	Gemma-3-4B ‡	0.50	4369	$0.46_{-0.04\downarrow}$	0.08	78	$0.44_{-0.06\downarrow}$	0.16
	GPT-5.4	0.83	72	$0.74_{-0.09\downarrow}$	0.08	78	$0.86_{+0.03\uparrow}$	0.04
	Gemini-2.5-Flash	0.90	72	$0.80_{-0.10\downarrow}$	0.11	78	$0.80_{-0.10\downarrow}$	0.11
	Qwen3.5-9B	0.73	72	$0.74_{+0.01\uparrow}$	0.03	78	$0.70_{-0.03\downarrow}$	0.06
	Llama-3.1-8B	0.61	72	$0.60_{-0.01\downarrow}$	0.10	78	$0.50_{-0.11\downarrow}$	0.27
Gemma-3-4B	Gemma-3-4B †	0.47	3484	$0.14_{-0.33\downarrow}$	0.70	7841	$0.06_{-0.41\downarrow}$	0.88
	Gemma-3-12B ‡	0.69	3650	$0.68_{-0.01\downarrow}$	0.03	71	$0.54_{-0.15\downarrow}$	0.21
	Gemma-3-27B ‡	0.63	3299	$0.64_{+0.01\uparrow}$	0.03	71	$0.50_{-0.13\downarrow}$	0.19
	GPT-5.4	0.89	66	$0.90_{+0.01\uparrow}$	0.02	71	$0.74_{-0.15\downarrow}$	0.14
	Gemini-2.5-Flash	0.87	66	$0.84_{-0.03\downarrow}$	0.05	71	$0.78_{-0.09\downarrow}$	0.12
	Qwen3.5-9B	0.74	66	$0.68_{-0.06\downarrow}$	0.06	71	$0.72_{-0.02\downarrow}$	0.05
	Llama-3.1-8B	0.61	66	$0.58_{-0.03\downarrow}$	0.06	71	$0.50_{-0.11\downarrow}$	0.20

odd, but reference models often preserve the original answer using the remaining context and answer choices.

Gemma-3-12B. Gemma-3-12B is more often affected by milder edits, including near-synonyms, register shifts, casing changes, and small changes in function words. These edits are closer to ordinary wording variation, although not all are strict paraphrases.

H. Surrogate-Model Pipeline

We also explored a sentence-level surrogate-model pipeline. A surrogate generator produced multiple rephrasings per item, which were scored using a target/reference objective. The resulting preferences were used for supervised fine-tuning or direct preference optimization. These experiments are preliminary and complementary to the TextAttack setting, where perturbation scope and constraints are easier to audit.

Table 10. Selectivity evaluation of the BAE attack against a frontier target (Gemini-2.5-Flash) on 50 MMLU dev-split questions. **Selective**: attack uses our custom selectivity constraint; **Non-selective**: standard BAE. The model marked with † is the target; models marked with ‡ are in the optimization reference set; the remaining models are *unseen* (not used during adversarial optimization). **Calls**: number of model queries during optimization (unseen models receive one call per question). **ASR**: attack success rate. For the target, lower Acc.$_\Delta$ and higher ASR indicate a stronger attack; for non-targets, Δ close to zero indicates better selectivity. The selective constraint maintains a near-zero drop on reference and unseen models while still degrading the frontier target by 0.41.

Evaluated model	Orig. Acc.	Selective			Non-selective		
		Calls	Acc.	ASR	Calls	Acc.	ASR
Gemini-2.5-Flash †	0.83	6038	$0.42_{-0.41\downarrow}$	0.49	6666	$0.12_{-0.71\downarrow}$	0.86
Qwen3.5-9B ‡	0.72	6746	$0.72_{+0.00}$	0.00	88	$0.58_{-0.14\downarrow}$	0.19
Llama-3.1-8B ‡	0.62	6248	$0.62_{+0.00}$	0.03	88	$0.46_{-0.16\downarrow}$	0.29
Gemma-3-12B	0.68	70	$0.58_{-0.10\downarrow}$	0.18	88	$0.54_{-0.14\downarrow}$	0.26
GPT-5.4	0.89	70	$0.84_{-0.05\downarrow}$	0.05	88	$0.70_{-0.19\downarrow}$	0.22

Table 11. Selectivity evaluation of the BAE attack in the *improvement* setting on 50 MMLU dev-split questions: the attack is configured to **increase** target accuracy while preserving non-target behavior. Each block groups a distinct target. **Selective**: attack uses our custom selectivity constraint; **Non-selective**: standard BAE. Models marked with † are targets; models marked with ‡ are in the optimization reference set; all others are *unseen* (not used during adversarial optimization). **Calls**: number of model queries during optimization (unseen models receive one call per question). **ASR**: attack success rate (fraction of questions where the target switches from incorrect to correct). For targets, higher Acc.$_\Delta$ and higher ASR indicate stronger selective improvement; for non-targets, Δ close to zero indicates better selectivity.

Target	Evaluated model	Orig. Acc.	Selective			Non-selective		
			Calls	Acc.	ASR	Calls	Acc.	ASR
Gemma-3-12B	Gemma-3-12B †	0.66	2041	$0.86_{+0.20\uparrow}$	0.30	4292	$0.96_{+0.30\uparrow}$	0.45
	Qwen3.5-9B ‡	0.73	2626	$0.74_{+0.01\uparrow}$	0.01	65	$0.68_{-0.05\downarrow}$	0.07
	Llama-3.1-8B ‡	0.63	2234	$0.62_{-0.01\downarrow}$	0.02	65	$0.62_{-0.01\downarrow}$	0.02
	GPT-5.4	0.87	60	$0.88_{+0.01\uparrow}$	0.01	65	$0.84_{-0.03\downarrow}$	0.03
	Gemini-2.5-Flash	0.86	60	$0.86_{+0.00}$	0.00	65	$0.80_{-0.06\downarrow}$	0.07
	Gemma-3-27B	0.63	60	$0.66_{+0.03\uparrow}$	0.05	65	$0.64_{+0.01\uparrow}$	0.02
Llama-3.1-8B	Llama-3.1-8B †	0.58	2564	$0.86_{+0.28\uparrow}$	0.48	4508	$0.88_{+0.30\uparrow}$	0.52
	Gemma-3-12B ‡	0.67	2854	$0.68_{+0.01\uparrow}$	0.01	65	$0.66_{-0.01\downarrow}$	0.01
	Qwen3.5-9B ‡	0.69	2486	$0.74_{+0.05\uparrow}$	0.07	65	$0.76_{+0.07\uparrow}$	0.10
	GPT-5.4	0.88	64	$0.82_{-0.06\downarrow}$	0.07	65	$0.90_{+0.02\uparrow}$	0.02
	Gemini-2.5-Flash	0.86	64	$0.84_{-0.02\downarrow}$	0.02	65	$0.80_{-0.06\downarrow}$	0.07
	Llama-3.1-70B	0.76	64	$0.74_{-0.02\downarrow}$	0.03	65	$0.74_{-0.02\downarrow}$	0.03
Qwen3.5-9B	Qwen3.5-9B †	0.71	1698	$0.90_{+0.19\uparrow}$	0.27	3311	$0.98_{+0.27\uparrow}$	0.38
	Gemma-3-12B ‡	0.62	2205	$0.70_{+0.08\uparrow}$	0.13	64	$0.70_{+0.08\uparrow}$	0.13
	Llama-3.1-8B ‡	0.60	1841	$0.58_{-0.02\downarrow}$	0.03	64	$0.64_{+0.04\uparrow}$	0.07
	GPT-5.4	0.89	59	$0.86_{-0.03\downarrow}$	0.03	64	$0.86_{-0.03\downarrow}$	0.03
	Gemini-2.5-Flash	0.88	59	$0.84_{-0.04\downarrow}$	0.05	64	$0.86_{-0.02\downarrow}$	0.02
	Qwen3.5-27B	0.84	59	$0.82_{-0.02\downarrow}$	0.02	64	$0.86_{+0.02\uparrow}$	0.02

Table 12. Selectivity evaluation of the DeepWordBug (character-level) attack on 50 MMLU dev-split questions. Each block groups a distinct target. **Selective**: attack uses our custom selectivity constraint; **Non-selective**: standard DeepWordBug. Models marked with † are targets; models marked with ‡ are in the optimization reference set; all others are *unseen* (not used during adversarial optimization). **Calls**: number of model queries during optimization (unseen models receive one call per question). **ASR**: attack success rate. Compared with BAE (Table 8), DeepWordBug induces smaller target drops overall and produces less consistent selectivity, indicating that character-level perturbations are weaker drivers of model-specific failures than word-level substitutions.

Target	Evaluated model	Orig. Acc.	Selective			Non-selective		
			Calls	Acc.	ASR	Calls	Acc.	ASR
Gemma-3-12B	Gemma-3-12B †	0.65	1727	$0.42_{-0.23\downarrow}$	0.34	2032	$0.36_{-0.29\downarrow}$	0.45
	Qwen3.5-9B ‡	0.71	1314	$0.76_{+0.05\uparrow}$	0.00	65	$0.72_{+0.01\uparrow}$	0.05
	Llama-3.1-8B ‡	0.61	1294	$0.62_{+0.01\uparrow}$	0.03	65	$0.52_{-0.09\downarrow}$	0.14
	GPT-5.4	0.85	61	$0.84_{-0.01\downarrow}$	0.00	65	$0.80_{-0.05\downarrow}$	0.07
	Gemini-2.5-Flash	0.87	61	$0.82_{-0.05\downarrow}$	0.05	65	$0.82_{-0.05\downarrow}$	0.07
	Gemma-3-27B	0.61	61	$0.62_{+0.01\uparrow}$	0.03	65	$0.52_{-0.09\downarrow}$	0.13
Llama-3.1-8B	Llama-3.1-8B †	0.62	1689	$0.38_{-0.24\downarrow}$	0.41	1759	$0.36_{-0.26\downarrow}$	0.40
	Gemma-3-12B ‡	0.68	1295	$0.68_{+0.00}$	0.00	62	$0.68_{+0.00}$	0.00
	Qwen3.5-9B ‡	0.76	1157	$0.74_{-0.02\downarrow}$	0.03	62	$0.72_{-0.04\downarrow}$	0.05
	GPT-5.4	0.86	63	$0.86_{+0.00}$	0.00	62	$0.86_{+0.00}$	0.00
	Gemini-2.5-Flash	0.88	63	$0.84_{-0.04\downarrow}$	0.07	62	$0.86_{-0.02\downarrow}$	0.00
	Llama-3.1-70B	0.77	63	$0.78_{+0.01\uparrow}$	0.00	62	$0.74_{-0.03\downarrow}$	0.03
Qwen3.5-9B	Qwen3.5-9B †	0.72	1954	$0.58_{-0.14\downarrow}$	0.19	2128	$0.56_{-0.16\downarrow}$	0.22
	Gemma-3-12B ‡	0.68	1589	$0.68_{+0.00}$	0.03	58	$0.68_{+0.00}$	0.03
	Llama-3.1-8B ‡	0.62	1482	$0.58_{-0.04\downarrow}$	0.03	58	$0.60_{-0.02\downarrow}$	0.06
	GPT-5.4	0.87	57	$0.82_{-0.05\downarrow}$	0.05	58	$0.86_{-0.01\downarrow}$	0.02
	Gemini-2.5-Flash	0.88	57	$0.86_{-0.02\downarrow}$	0.00	58	$0.92_{+0.04\uparrow}$	0.00
	Qwen3.5-27B	0.82	57	$0.80_{-0.02\downarrow}$	0.02	58	$0.82_{+0.00}$	0.00