Stability Through Structure: The Impact of Knowledge Graphs on Tiny Large Language Models

Anonymous ACL submission

Abstract

With billions of users communicating and expressing in various ways, it is unrealistic to expect uniformity in language input–for example, human error in text. As a result, a single question, even if it conveys the same intent, can be articulated in numerous ways and often imperfectly. Although large language models (LLMs) have demonstrated impressive capabilities, they still exhibit inconsistencies in handling diverse inputs. More importantly, this challenge also exists in small language models (SLMs), which are increasingly deployed in resource-constrained and edge settings. However, despite their growing adoption, very few studies have explored the effects of perturbations in SLMs. This leaves a critical gap in the development of reliable, lightweight NLP systems. In this study, we focused on both general question domains and the medical domain as a more complex domain where a knowledge source is generally referred to. We investigate the effect of different perturbation types on a deeper level by analyzing input perturbation using a knowledge graph(KG) as a factual base and minimizing the effect on the LLM's results. The main technique works by effectively integrating relevant knowledge(in the form of a subgraph) such that it contains all the knowledge required to answer the question, and any possible perturbations have minimal effect on the prediction, which ultimately enhances the robustness of a tiny LLM to input perturbations. On average, we observed an improvement of **133.15**% for original questions and **124.09**% for perturbed questions across models and datasets when compared with zero-shot. We also establish the usefulness of a knowledge graph in explaining the prediction.

1 Introduction

With the rapid advancement of language models (LMs), sensitivity to input perturbations and explainability continue to pose significant challenges. In this context, perturbations refer to both intentional and unintentional modifications to input text or model parameters that influence the model's outputs. These changes may range from minor character-level edits, such as typographical errors, to more substantial structural or semantic rephrasing. Understanding the consequences of such perturbations is crucial for enhancing the robustness, reliability, and interpretability of language models. Notably, many perturbations arise not from adversarial intent but from the natural diversity in human expression, particularly in multi-user environments (Ehsanzadeh and Dehnad, 2024). When different users interact with the same model, they often phrase semantically equivalent queries in diverse waysthrough variations in style, structure, or vocabulary. In these settings, it is reasonable to expect language models to produce consistent and accurate outputs despite differences in surface forms. However, studies (Rauba et al., 2024) have shown that models frequently exhibit unstable behaviour under such input variations, often resulting in inconsistent predictions or degraded performance (Hu et al., 2024). Even a slight change in the inputs results in a change in the prediction of the language models (Lin et al., 2025) or in other tasks, such as code generation (Chen et al., 2024). Despite the growing attention to this problem in large-scale language models (LLMs), to the best of our knowledge, no recent studies have examined perturbation sensitivity in smaller models. This is a critical gap, as small models are increasingly deployed in resource-constrained and on-device scenarios (Liu et al., 2024) where robustness remains important, but computational budgets restrict the use of heavyweight mitigation strategies such as ensemble decoding or paraphrase smoothing. Ensuring robustness under naturally occurring prompt variations is thus especially vital for the reliable deployment of small models in real-world applications. Another underexplored dimension is

the explanation and diagnosis of independent and perturbation-induced behaviors. While approaches such as chain-of-thought (CoT) prompting aim to improve reasoning and robustness, most existing studies focus on only LM (Zhao et al., 2024) or KG (Wang et al., 2019) or LMs using only KG triplates (Jiang et al., 2024). While a deeper understanding of the learned KG representation, along with language models, can factor in causal effects, it can also help build models that are not only more robust but also more explainable.

To fill these gaps, in this study, we analyze the effect of perturbation primarily on small LLMs and investigate the impact of the knowledge graph on the stability of the models. It also establishes the use of attention-guided explainability and analyzes the effect of perturbation on the model using a Knowledge Graph.

Contributions

(1) We analyse the efficiency of co-learning of language model with the knowledge graph for small language models on multiple datasets. This shows that including a knowledge graph for small languages can boost performance, even in the case of perturbation, owing to the co-learning of knowledge graph representations. This can be a very helpful tool for enhancing models for low-resource settings.

(2) Robustness analysis under perturbations: We systematically study the effect of input variations, demonstrating that KG integration enhances stability and consistency in model outputs.

(3) We investigate whether graph path extraction based on attention correlates with the model's predictability?

(4) Propose a novel way of understanding the effect of perturbations using a knowledge graph, which will help in making the model better for explainability.

(5) Toward trustworthy LMs: By combining interpretability with robustness, we provide evidence that knowledge graph–augmented Tiny LLMs move closer to the goal of transparent, reliable, and trustworthy AI.

2 Related Work

Perturbations and their effects are one of the highly studied problems, with earlier works mostly focusing on the techniques to propose different types of perturbations (Belinkov and Bisk, 2018; Li et al., 2020; Singh et al., 2024) such as the word-level perturbation proposed by (Hiraoka et al., 2022). Recent studies have mostly focused on its effect on Large Language models (Rauba et al., 2024). Recent research has begun to address the issue of inconsistent predictions or degraded performance. (Fu and Barez, 2025) proposed a Latent Adversarial Paraphrasing framework designed to expose and mitigate model brittleness under prompt-level variation. (Lin et al., 2025) shows how even a single character introduces changes in the output token probability of the model. They also analyzed the effect of the smallest change, such as adding a single space to the input prompt, on the model output. Sentence-Level Perturbations have also been studied, which involve synonym replacements, paraphrasing, token shuffling, or masking (Qiang et al., 2024). Structural changes, such as altering the order of tokens or introducing nonsensical strings, can disrupt the model's reasoning and understanding (Marjanović et al., 2024). For instance, (Chaudhary et al., 2024; Chatziveroglou et al., 2025) evaluate models for different prompt perturbations, and (Zhu et al., 2024) introduce PromptBench, a benchmark for evaluating model robustness under a wide spectrum of prompt perturbations. Another study on semantic and structural perturbations found that misleading comments in code, symbolic perturbations, or formatting changes (e.g., JSON vs. plain text) can introduce structured variations in latent embeddings, affecting the model's contextual sensitivity (Lam et al., 2025). Recent research has begun to address the issue of inconsistent predictions or degraded performance.

These studies underscore the vulnerability of models to even subtle, meaning-preserving rephrasings and highlight that perturbation sensitivity can lead to significant issues, including performance degradation, hallucinations, toxicity, bias, and inconsistent responses, even when inputs are semantically equivalent. While most studies focus on understanding the effect of perturbations on Large Language models, to the best of our knowledge, no recent study has been conducted to understand the perturbation sensitivity of small LLMs.

3 Methodology

In this study, we analyzed the effect of perturbation on the language model with and without including a knowledge graph architecture, using **GreaseLM** (Graph Reasoning Enhanced Language

Model) (Zhang et al., 2022) and its enhanced architecture for multiple language models. GreaseLM is a hybrid neuro-symbolic architecture designed to augment language model reasoning with structured knowledge from external knowledge graphs (KGs). It aims to improve the performance of knowledge-intensive tasks by integrating symbolic graph-based reasoning into pretrained transformer-based language models. The GreaseLM pipeline consists of the following key components: Subgraph Retrieval and Pruning, Dual Encoding via LM and GNN, Fusion via Cross-Attention and Prediction. In **Subgraph Retrieval and Pruning**, given a question and its multiple-choice answer options, GreaseLM performs entity linking using a linker (e.g., TAGME or spaCy+Wikipedia) to identify the relevant KG concepts mentioned in the input. Based on these anchor nodes, a local subgraph is retrieved from the KG (e.g., ConceptNet or Wikipedia-derived graph). The subgraph is then pruned using heuristic rules, such as node centrality, relevance to the question, and maximum hop distance, to maintain tractability and focus on the salient reasoning paths. In **Dual Encoding via LM and GNN**, the model processes the input through two parallel encoders: a transformer-based language model (e.g., RoBERTa) encodes the concatenated question and answer option text, generating contextualized token representations. Simultaneously, a Graph Neural Network (e.g., GAT or GATConvE) encodes the subgraph extracted from the KG, producing embeddings for the linked and intermediate KG nodes. In **Fusion via Cross-Attention and Prediction**, a cross-attention module integrates the LM token representations with the GNN node embeddings. This joint reasoning allows the model to attend to both textual and graph-derived knowledge representations. The fused output is used to compute scores for each answer option, typically followed by a classification layer (e.g., softmax) to predict the most probable answers. The original GreaseLM model is Roberta-based, and in the modified versions, we replaced Roberta with different small LLMs.

Language Model Only-Setup: To isolate the contribution of the language model, we conducted experiments using language models without incorporating any graph-based components. Our implementation closely follows the GreaseLM training setup with identical preprocessing, loss functions, and optimizer configurations.

Explanation Path Extraction GreaseLM (LM + GNN) and similar models often act as black-box predictors, This makes it difficult to understand how decisions are made. Small perturbations (typos, passive voice, gibberish) can alter reasoning paths or even flip predictions. To address this, we investigate how the model connects *question concepts* (QC) to *answer concepts* (AC). Visualizing attention-based reasoning paths reveals which parts of the model's reasoning remains stable and which parts become noisy under perturbations.

We extract and visualize attention-based paths from question concepts to answer concepts using the following pipeline:

1. **Grounded Subgraph Construction:** Build a subgraph with concepts from the question and candidate answers, using ConceptNet.

2. **Inference with GreaseLM:** The LM encodes the question, while The GNN processes the subgraph.

3. **Attention Collection:** Extract edge attention weights from GNN layers.

4. **Path Extraction & Visualization:** Identify top-k QC→AC paths scored by attention. Nodes are color-coded as follows: red = question concepts, blue = answer concepts, green = context nodes, orange = interaction node, and grey = others.

4 Experimental Setup

In this section, we describe our framework for evaluating and improving the robustness of small language models (SLMs) using knowledge graphs. We built upon the GreaseLM architecture by modifying subgraph extraction to enhance stability across a range of perturbed inputs. Our methodology comprises three key components: (1) perturbation setup, (2) GreaseLM baseline, and (3) attention-based explanation in graph to understand the perturbation effects.

4.1 Datasets and Preprocessing

We evaluate our methods on three multiple-choice QA benchmarks. *CommonsenseQA (**CSQA**)*: A 5-option commonsense reasoning dataset grounded in ConceptNet, requiring selection of the most plausible answer to a natural question (Talmor et al., 2018). *OpenBookQA (**OBQA**)*: is a 4-way multiple-choice question answering dataset that tests elementary scientific knowledge. It contains 5,957

questions along with an open book of scientific information. We use the official data splits from (Mihaylov and Frank, 2018). ***MedQA (USMLE)***: A medical multiple-choice QA dataset containing professional medical exam-style questions. It requires deep domain knowledge and long-context reasoning and is typically evaluated using the 4-option version (USMLE format) (Jin et al., 2021).

We used the original GreaseLM preprocessing pipeline for each dataset, which involved the following: *Entity Linking*: Concepts in the question and answer options are linked to corresponding KG nodes (e.g., ConceptNet or UMLS). *Subgraph Construction*: A local subgraph is extracted using a relevance-driven hop-based method centred on linked entities. *Graph Pruning*: The subgraph is pruned based on edge weights, path types, and size constraints.

4.2 Perturbation Setup

To simulate realistic input variations encountered in user-generated text, we apply a diverse set of perturbation types to the question inputs. These perturbations aimed to preserve the original semantic intent while altering the surface form, thereby testing the robustness of both the language model and knowledge graph pipeline. Our perturbation set includes a simple lexical change to multiple-word changes, where we keep the sentiment of the question the same, even though the words are changed in the question. All perturbations were designed to retain the original intent of the question without modifying the answer options. All perturbations maintained the original multiple-choice answer set, ensuring consistency in the evaluation targets. This setup enables a controlled analysis of how each perturbation type influences the subgraph retrieval process, attention distribution over the knowledge nodes, and ultimately, the prediction consistency of the model. Full set of perturbations are given in Table 1 with the complete description mentioned in the **appendix A.**

4.3 Models

We used the GreaseLM model as a baseline and its modified version, as described in Section 3. We selected multiple open-source LLMs with varying sizes and capabilities to ensure robust evaluation. The sizes range from 360M to 1B, and these are *LLaMA 3.2 1B* (Jayalath et al., 2024), *Qwen3-0.6B* (Zhang et al., 2025), SmolLM2-360M (Dubey et al., 2025) and Roberta-Large (Liu et al., 2019). In the case of MedQA for Roberta comparison, we used the same architecture and training setup but replaced the default tokeniser with SapBERT[1], a biomedical domain-adapted RoBERTa model pretrained for medical entity similarity tasks, in the case of original GreaseLM finetuning. This choice aligns with GreaseLM's MedQA setup, allowing for a direct comparison between LM-only and LM+KG configurations. All other hyperparameters, including dropout rates and batch size, remained consistent with the GreaseLM defaults. For all models, we used the base variants to compare their performances under various prompting strategies. All the LLMs selected in this study are open-source models, which will help interested researchers to continue with this analysis. Here, we use model abbreviations such as Llama, Qwen, and SmolLM.

4.4 Experimental Procedure

For all GreaseLM experiments, input pairs were tokenized with truncation and padding to a maximum LM sequence length of 128 tokens for CSQA ,OBQA and 512 tokens for MedQA. Each example's pruned concept subgraph (from CPNet) was capped at 200 nodes, and message passing exchanged information with the LM every other layer . We used an effective batch size of 128 and learning rate of 1×10^{-5}. Training ran for up to 30 epochs for CSQA, 40 epochs for OBQA and 20 epochs for MedQA with early stopping after 10 epochs without validation improvement. Checkpoints were saved every epoch and training could be safely resumed. Experiments were executed on 40GB *NVIDIA A100* GPUs. These LM+GNN settings are reported alongside LM-only baselines to isolate the contribution of structured knowledge, especially under perturbed-question evaluation. The total compute was approximately **84** GPU-hours.

5 Results

The results in Table 2 present the performance of four models—RoBERTa-large, SmolLM2-360M, Qwen3-0.6B, and LLaMA3.2-1B—on the CSQA, OBQA, and MedQA datasets under both original and perturbed conditions. Across all datasets, a consistent trend emerged: the model performance declined under perturbation, demonstrating the sensitivity of these models to input variations. For

[1] https://huggingface.co/cambridgeltl/SapBERT-from-PubMedBERT-fulltext

Table 1: Examples of Perturbation Types

Perturbation Type	Example Question
Original	The archaeologist was seeing artifacts that made no sense, so what did he do?
Insert	The archaeologist was seeing **seeing** artifacts that made no no sense , so what did he do ?
Repeat	The archaeologist was **was** seeing **seeing** artifacts that made no sense , so what did he do ?
Delete	The archaeologist was ~~seeing~~ artifacts that ~~made~~ no sense , so what did he do ?
Replace	The archaeologist **live** seeing artifacts that made no smell out , so what did he do ?
Typo	ihe archaeolgpizt mas yeeinq artifacec that mkde no stnse, so ieaj zid hs do?
Passive	Artifacts that made no sense were being seen by the archaeologist, so what was done by him?
Contextual	the man was seeing artifacts that made no sense, so what did he do?
Leet/Gibberish	h3 arch43olog!57 w4s 5|eeing ar7ifacts t&hat mad3 no s3n6s3, $o wh4t d1d he% do?
Max Bleu	The archaeologist was seeing artifacts that made no sense, so what did he do?
Min Bleu	Upon seeing artifacts with no clear explanation, what did the archaeologist choose to investigate?
Paraphrase	Upon seeing artifacts with no clear explanation, what did the archaeologist choose to investigate?
Single Char	The archaeologist was seeing artifacts that made no sense, so what did he do?**?**

CSQA, the GreaseLM variant achieved the highest original accuracy (73.81%) and maintained relatively strong robustness under perturbation (63.04%), outperforming both the fine-tuned and base configurations across models. Similarly, for OBQA, GreaseLM and Finetuned models achieved notably higher accuracy compared to base versions, with LLaMA3.2-1B Finetune showing the best perturbed accuracy (83.21%), reflecting strong resilience. In contrast, MedQA results were generally lower across all models, indicating the greater difficulty of medical reasoning tasks; nonetheless, GreaseLM again delivered the best balance between original (38.49%) and perturbed (35.81%) performance. Overall, these findings underscore that while fine-tuning improves task-specific accuracy, the GreaseLM approach consistently enhances both generalization and robustness, particularly under perturbations.

5.1 Ablation Studies

Model Robustness Analysis We evaluate LM-only and GreaseLM models under various perturbations on multiple datasets. Each subsection reports the test accuracy across perturbation types, comparing a baseline LM-only model against GreaseLM using separate subgraphs. It substantially enhances the robustness across the CSQA, OBQA, and MedQA datasets. It consistently outperforms zero-shot and fine-tuned baselines, mitigating the impact of perturbations such as Typo, Leetgibberish, Paraphrase, Insert, Repeat, and Replace. Gains are most pronounced in the CSQA, where GreaseLM stabilizes reasoning under noisy conditions, whereas OBQA maintains a strong performance with minimal degradation. MedQA benefits modestly, reflecting the challenges associated with domain-specific reasoning. Although extreme surface-level noise can still reduce accuracy, GreaseLM demonstrates clear improvements in semantic generalization and resilience, enabling more reliable model predictions under diverse perturbations. Across Roberta-large, LLaMA-3.2-1B, Qwen3-0.6B, and SmolLM2-360M, GreaseLM consistently provides the largest robustness gains, with the most pronounced benefits observed in commonsense-heavy tasks, such as CSQA. In contrast, domain-specific tasks, such as MedQA, see more modest but consistent improvements. Overall, these results highlight GreaseLM's effectiveness in enhancing the stability and generalization of language models under clean and noisy input conditions.

Prediction Flip Analysis Figure 1 shows model-variant and dataset-specific patterns in total flip counts. On CSQA, finetuned models and especially GreaseLM produce more flips for Llama and Qwen, whereas SmolLM shows the opposite tendency. GreaseLM produces significantly fewer flips on average on MEDQA than both zero-shot and fine-tuned versions. On OBQA, total flips are small overall but are typically highest for the GreaseLM variants. Taken together, these results suggest that GreaseLM and finetuned models are less robust on

Table 2: Performance on CSQA, OBQA, and MedQA datasets across models and perturbations

Dataset	Method	Roberta-large		Smollm2-360M		Qwen3-0.6B		Llama3.2-1B	
		Original	Perturbed	Original	Perturbed	Original	Perturbed	Original	Perturbed
CSQA	Zero-shot	20.78	20.07	21.19	21.38	32.31	25.30	24.01	23.19
	Finetune	67.68	58.45	20.23	19.83	62.61	55.39	59.31	52.08
	GreaseLM	**73.81**	63.04	62.61	53.99	63.01	53.78	66.72	54.58
OBQA	Zero-shot	26.2	23.71	22	21.82	45	44.54	28.6	27.86
	Finetune	80	78.62	75.6	74.14	81.8	80.33	84.6	83.21
	GreaseLM	**85**	82.83	79.2	76.49	80.8	79.13	81.8	80.72
MedQA	Zero-shot	24.66	24.23	22.86	23.21	27.1	26.3	23.72	25.01
	Finetune	37.31	35.35	27.89	26.84	30.32	30.89	24.74	25.25
	GreaseLM	**38.49**	35.81	25.53	25.59	35.74	33.95	29.54	30.05

CSQA and OBQA, while GreaseLM is comparatively more robust on MEDQA.

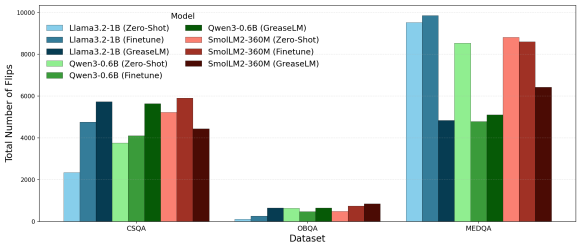

Figure 1: This figure shows the total number of flips for each dataset and model. For MEDQA GreaseLM variants on average has significantly lesser number flips compared to other variants.

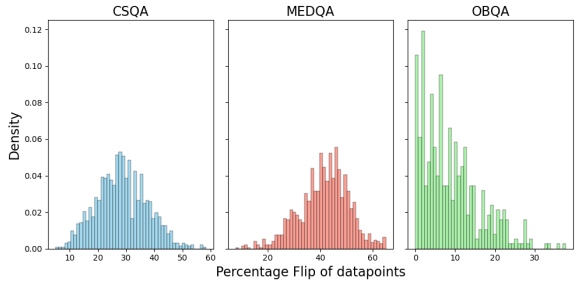

Figure 2: This figure shows the distribution of the number of flips for datapoints across models and perturbations. For the obqa dataset, many data points were predicted consistently to have the same value.

Error Consistency A consistency analysis was conducted across three datasets—CSQA, OBQA, and MedQA—to examine the stability of the model predictions under various perturbations. Consistency was defined as the proportion of data points whose predictions never changed across all perturbations, whereas the always flip count represented those that changed every time. The results indicated that both CSQA (data size: 1241) and MedQA (data size: 1273) exhibited no completely stable data points, reflecting a consistency of 0%. In contrast, OBQA (data size: 500) showed a small but notable stability, with 20 data points never flipping, corresponding to a consistency rate of 4%. Importantly, no dataset contained data points that consistently flipped across all perturbations. Overall, these findings suggest that model predictions are generally sensitive to perturbations, highlighting their limited robustness across datasets. OBQA demonstrates relatively higher stability; OBQA datasets utilize the provided context, which may contribute to stable predictions over different settings.

Data Distribution Shift Different types of text perturbations affect datasets in various ways. Minor changes, such as single-character edits or repeated words, have a minimal impact on semantic embeddings. Moderate changes, such as paraphrasing, word replacements, or deletions, cause noticeable shifts, whereas more extreme modifications, such as passive voice, severe typos, or leetGibberish, have the largest effect. Examining the datasets individually, MEDQA was the most sensitive, with passive voice and typos resulting in significant drops, likely because of its rigid medical language. OBQA is generally robust but struggles with leet gibberish, whereas CSQA is more affected by the passive voice than OBQA. Overall, both the type of perturbation and the linguistic characteristics of the dataset shape the resilience of the embeddings to noise.

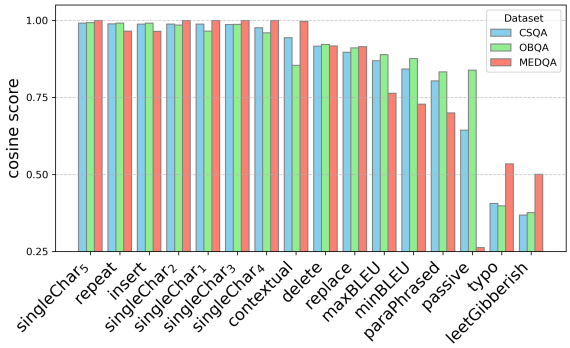

Figure 3: Distribution shift of data in different perturbation cases.

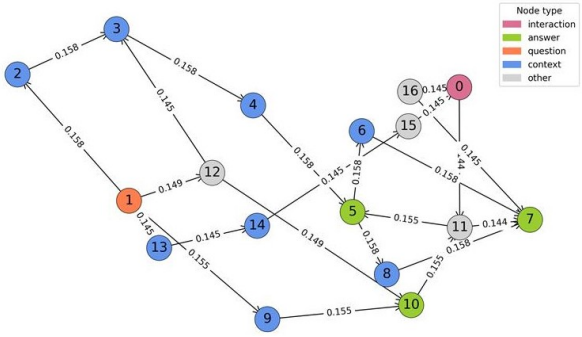

Figure 4: Question Subgraph, the nodes are represented as 0: ab extra, 1: value, 2: concede, 3: arbitrary, 4: graffiti, 5: save money, 6: buy cheaper flowers, 7: save, 8: small medium, 9: home team, 10: money, 11: being outdoors on hot day, 12: disloyal, 13: vote against, 14: savings, 15: like price, 16: deflationary

5.2 Reasoning Pattern Analysis on Prediction Flips

Correct to Wrong Analysis Here, we analyse the case when the nonperturbed answer was predicted correctly, but the perturbed question was wrongly predicted. Consider the question: *James wanted to fight inflation because he did not like losing value when he did what? A. more money B. save money C. can D. costs money E. pay less*, where 'save money' is the correct answer. We analyzed the knowledge graph (KG) paths to explain the model's predictions.

From the graph 4, we can see that the model chose to save money because its graph kept focusing on value and money. The strongest routes ran along value → money → save/save money, with the biggest edge scores around 0.158 (others 0.155, 0.149, 0.145, 0.144). Several top paths also passed through related nodes, such as savings, price, and deflationary, but they still ended at save money. Adding the shown top paths gives approximately

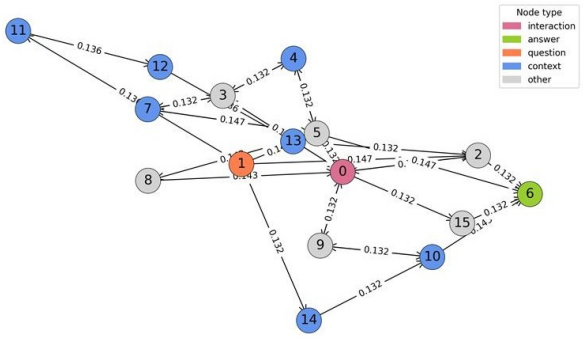

Figure 5: Perturbed(Max BLEU) Question Subgraph, the nodes are represented as 0: ab extra, 1: inflation, 2: worthlessness, 3: for free, 4: surprised, 5: save, 6: money, 7: chump chop, 8: deflationary, 9: like price, 10: well off, 11: home team, 12: dexterous, 13: support, 14: concede, 15: losingest.

1.50, which is the largest total going into any answer. Paths to the other choices were rare or extremely weak. In short, the graph shifted its focus from value to money and then to saved money, aligning with the idea that inflation erodes the value of saved money.

For the paraphrase question: *What did James want to combat specifically, 'fight inflation' due to his aversion?* the focus shifted away from the previous path. The most-attended pieces became money and generic connectors such as worthlessness, for free, support, and concede. The biggest edge scores were now around 0.147 (others 0.143, 0.136, 0.132). Multiple routes now ended at costs money, flowing through these generic hubs instead of the earlier value → money link. The total over the top paths was about 1.38—close, but the attention that previously powered value → money → save money dropped significantly. Consequently, routes to save money were no longer preferred, and the model switched to costs money.

Correct to Correct Analysis Here, we analyse the case when both the nonperturbed and perturbed question was predicted correctly to analyse how the changes are happening in the graph even in the case of the same prediction.

Consider the question: *To locate a choker not located in a jewelry box or boutique where would you go? A. jewelry store B. neck C.jewlery box D. jewelry box E. boutique*, where 'jewelry store' is the correct answer. We analyzed the knowledge graph (KG) paths to explain the model's predictions.

From the graph 6, we can see that the model predicted jewelry store. In the graph, the question side

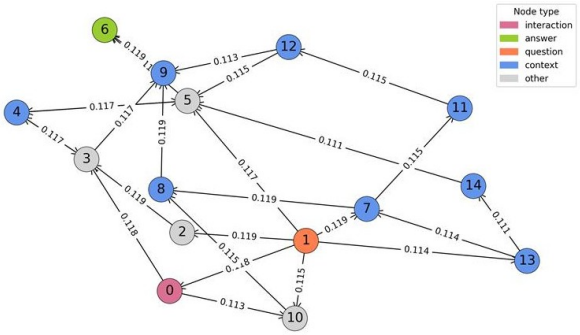

Figure 6: Question Subgraph, the nodes are represented as 0: ab extra 1: boutique 2: dark alley late at night 3: licensed 4: jewellary store 5: braces 6: jewelry store 7: decoration day 8: ear lobe 9: necklace 10: jewlery box 11: rubbish 12: jewelery shop 13: ring down curtain 14: mens clothing store.

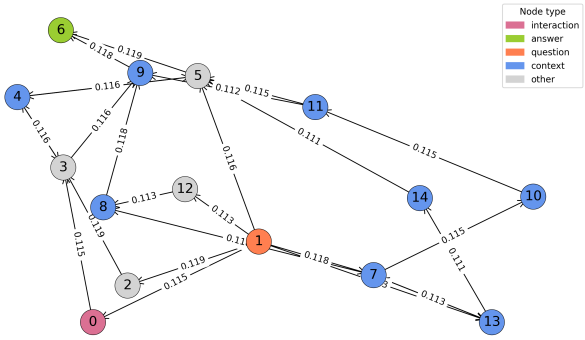

Figure 7: Perturbed(Contextual) Question Subgraph, the nodes are represented as 0: ab extra 1: boutique 2: dark alley late at night 3: licensed 4: jewellary store 5: braces 6: jewelry store 7: decoration day 8: ear lobe 9: necklace 10: jewlery box 11: rubbish 12: jewelery shop 13: ring down curtain 14: mens clothing store.

centers on boutique, and the answer side on jewelry store. The highest ranked path is boutique → ring down curtain → use all → licensed → jewellary store → braces → jewelry store (0.144). Several other strong routes also reach jewelry store, for example through the necklace chain (boutique → decoration day → ear lobe → necklace → jewelry store, 0.119) and through dark alley late at night → licensed → jewellary store → braces → jewelry store (0.119). Summing the top-10 path scores gives about 1.23, clearly concentrating mass on jewelry store. Choices like neck, jewlery/jewelry box, and boutique either violate the question constraint or do not collect comparable path support.

For the paraphrase question:*to find a choker not located in a jewelry box or boutique where would you go?* The model again predicted jewelry store, but the traffic through the graph shifted slightly. The top path is still the same long bridge via licensed → jewellary store → braces (now 0.136), and the necklace route remains close in strength (0.118). New or reweighted trails (e.g., via grocery store → garbage can → braces) appear among the top-10. The total mass over the top-10 paths is 1.20 very close to the original—showing that, although individual routes changed and some edges weakened/strengthened, multiple independent paths still converge on jewelry store.

Both versions guide attention from boutique through retail/ornament concepts (e.g., licensed, jewellary store, necklace, braces) and end at jewelry store. The perturbed wording alters which paths are most prominent, but the graph continues to supply several strong, complementary routes to the same correct answer, jewelry store.

6 Conclusion

In this research, we investigated how incorporating a knowledge graph (KG) improves the reasoning capabilities of tiny language models and ensures their resilience to input changes or noise. When compared to standard or fine-tuned models, we discovered that co-learning with a KG significantly improves performance across datasets such as CSQA, OBQA, and MedQA. Because of this, the method is particularly useful in on-device and low-resource scenarios where model size and dependability are crucial. Although small models are still susceptible to perturbations, our robustness and error assessments demonstrated that models improved with a KG manage fluctuations significantly better. We saw how perturbations might change the thinking of the model in the Correct-to-Wrong analysis. For example, the perturbed version diverted attention to unrelated concepts like costs money or worthlessness, which caused the model to make the wrong choice when the original question correctly guided the model to save money through a clear path.
All things considered, this work shows that combining knowledge graphs with tiny LLMs improves their resilience and clarifies and interprets their reasoning, which advances our goal of creating small AI models that are reliable and understandable.

Limitations

Although our analysis sheds light on the explainability and resilience of small language models enhanced with knowledge graphs, there are still a number of limitations. Initially, our research is limited to four models, which restricts the findings'

applicability to a wider variety of architectures and scales. which could serve as a strong comparative baseline for evaluating the relative benefits . Future research will examine this comparison. Furthermore, we only use a small number of datasets for our experiments, most of which deal with commonsense and domain-specific question answering. Adding additional varied datasets, such as benchmarks that rely heavily on reasoning, could help us better understand how generalizable the approach is.

Lastly, although our path-level explainability and perturbation-based robustness analyses provide qualitative insights, they might not adequately account for all types of model uncertainty or reasoning bias. To further validate the method, future research might include higher model scales, real-world noisy inputs, and more quantitative explainability criteria.

References

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. Preprint, arXiv:1711.02173.

Giannis Chatziveroglou, Richard Yun, and Maura Kelleher. 2025. Exploring llm reasoning through controlled prompt variations. arXiv preprint arXiv:2504.02111.

Manav Chaudhary, Harshit Gupta, Savita Bhat, and Vasudeva Varma. 2024. Towards understanding the robustness of llm-based evaluations under perturbations. arXiv preprint arXiv:2412.09269.

Junkai Chen, Li Zhenhao, Hu Xing, and Xia Xin. 2024. Nlperturbator: Studying the robustness of code llms to natural language variations. ACM Transactions on Software Engineering and Methodology.

Akshita Dubey, Loubna Ben Allal, Anton Lozhkov, Leandro von Werra, and Thomas Wolf. 2025. Smollm2: When smol goes big – data-centric training of a small language model. Preprint, arXiv:2502.02737.

Seyed Jafar Ehsanzadeh and Afsaneh Dehnad. 2024. Analysis of high-frequency errors and linguistic patterns in efl medical students' english writing: Insights from a learner corpus. BMC medical education, 24(1):1264.

Tingchen Fu and Fazl Barez. 2025. Same question, different words: A latent adversarial framework for prompt robustness. arXiv preprint arXiv:2503.01345.

Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. 2022. Word-level perturbation considering word length and compositional subwords. In Findings of the Association for Computational Linguistics: ACL 2022, pages 3268–3275.

Zhibo Hu, Chen Wang, Yanfeng Shu, Hye-Young Paik, and Liming Zhu. 2024. Prompt perturbation in retrieval-augmented generation based large language models. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 1119–1130.

Dulhan Jayalath, Shashwat Goel, Thomas Simon Foster, Parag Jain, Suchin Gururangan, Cheng Zhang, Anirudh Goyal, and Alan Schelten. 2024. The llama 3 herd of models. Preprint, arXiv:2407.21783.

Zhouyu Jiang, Ling Zhong, Mengshu Sun, Jun Xu, Rui Sun, Hui Cai, Shuhan Luo, and Zhiqiang Zhang. 2024. Efficient knowledge infusion via kg-llm alignment. arXiv preprint arXiv:2406.03746.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. Applied Sciences, 11(14):6421.

Man Ho Lam, Chaozheng Wang, Jen-tse Huang, and Michael R Lyu. 2025. Codecrash: Stress testing llm reasoning under structural and semantic perturbations. arXiv preprint arXiv:2504.14119.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. arXiv preprint arXiv:2004.09984.

Leon Lin, Hannah Brown, Kenji Kawaguchi, and Michael Shieh. 2025. Single character perturbations break llm alignment. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pages 27473–27481.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. 2024. Mobilellm: optimizing sub-billion parameter language models for on-device use cases. In Proceedings of the 41st International Conference on Machine Learning, ICML'24. JMLR.org.

Sara Vera Marjanović, Isabelle Augenstein, and Christina Lioma. 2024. Investigating the impact of model instability on explanations and uncertainty. arXiv preprint arXiv:2402.13006.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. arXiv preprint arXiv:1805.07858.

Yao Qiang, Subhrangshu Nandi, Ninareh Mehrabi, Greg Ver Steeg, Anoop Kumar, Anna Rumshisky, and Aram Galstyan. 2024. Prompt perturbation consistency learning for robust language models. arXiv preprint arXiv:2402.15833.

Paulius Rauba, Qiyao Wei, and Mihaela van der Schaar. 2024. Quantifying perturbation impacts for large language models. arXiv preprint arXiv:2412.00868.

Ayush Singh, Navpreet Singh, and Shubham Vatsal. 2024. Robustness of llms to perturbations in text. arXiv preprint arXiv:2407.08989.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. arXiv preprint arXiv:1811.00937.

Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In Proceedings of the AAAI conference on artificial intelligence, volume 33, pages 5329–5336.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. 2022. Greaselm: Graph reasoning enhanced language models for question answering. Preprint, arXiv:2201.08860.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. arXiv preprint arXiv:2506.05176.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. ACM Transactions on Intelligent Systems and Technology, 15(2):1–38.

Kaijie Zhu, Qinlin Zhao, Hao Chen, Jindong Wang, and Xing Xie. 2024. Promptbench: A unified library for evaluation of large language models. Journal of Machine Learning Research, 25(254):1–22.

A
Perturbation Details

For generating different perturbations, the choices, labels, and IDs in each question are left unchanged. Tokenization/POS come from NLTK, synonyms from WordNet, dependency parsing from spaCy, and the contextual fill uses a Hugging Face fill-mask model. We implement a suite of stochastic text operators. *Leetspeak + gibberish* replaces letters with fixed leet equivalents and occasionally inserts a random symbol or digit after alphanumeric characters. *Strong typos* randomly substitute alphabetic characters with other letters. *Insert* duplicates randomly chosen content words (nouns, verbs, adjectives, or adverbs); *Repeat* is the same duplication behavior under a different name. *Delete* removes randomly chosen content words. *Replace (WordNet)* swaps randomly chosen content words with WordNet synonyms, skipping tokens without synonyms and converting underscores in returned lemmas to spaces. For *Passive voice*, when an SVO structure is detected, the sentence is rewritten as Object was VERB by Subject, using the verb's lemma; if no SVO is found, the sentence is left unchanged. Finally, *Contextual substitution (MLM)* masks a single eligible token and fills it with the top prediction from a masked-language model, stripping any special tokens. All operators are stochastic, and setting a random seed makes their outputs reproducible.

Beyond the rule-based operators, we also incorporated model-driven paraphrased perturbations. Using *Llama 3.2–1B*, we prompted the model to preserve the core *question concept* while freely paraphrasing all other surface content, and generated ten candidate perturbations per item. From these, we took a panel of five perturbation sets that best balanced diversity and faithfulness. For each original question, we then computed *BLEU* between the source and every candidate, and selected the *max-BLEU* (closest paraphrase) and *min-BLEU* (most divergent yet still on-concept) variants; aggregating these across items yielded two additional perturbation tracks that bracket semantic drift. Finally, inspired by the "Single Character Perturbations Break LLM Alignment" (Lin et al., 2025) line of work, we synthesized five *single-character* variants to probe brittleness under minimal orthographic noise.

B
Detailed results

In this section, we have added the results and plot for individual cases.

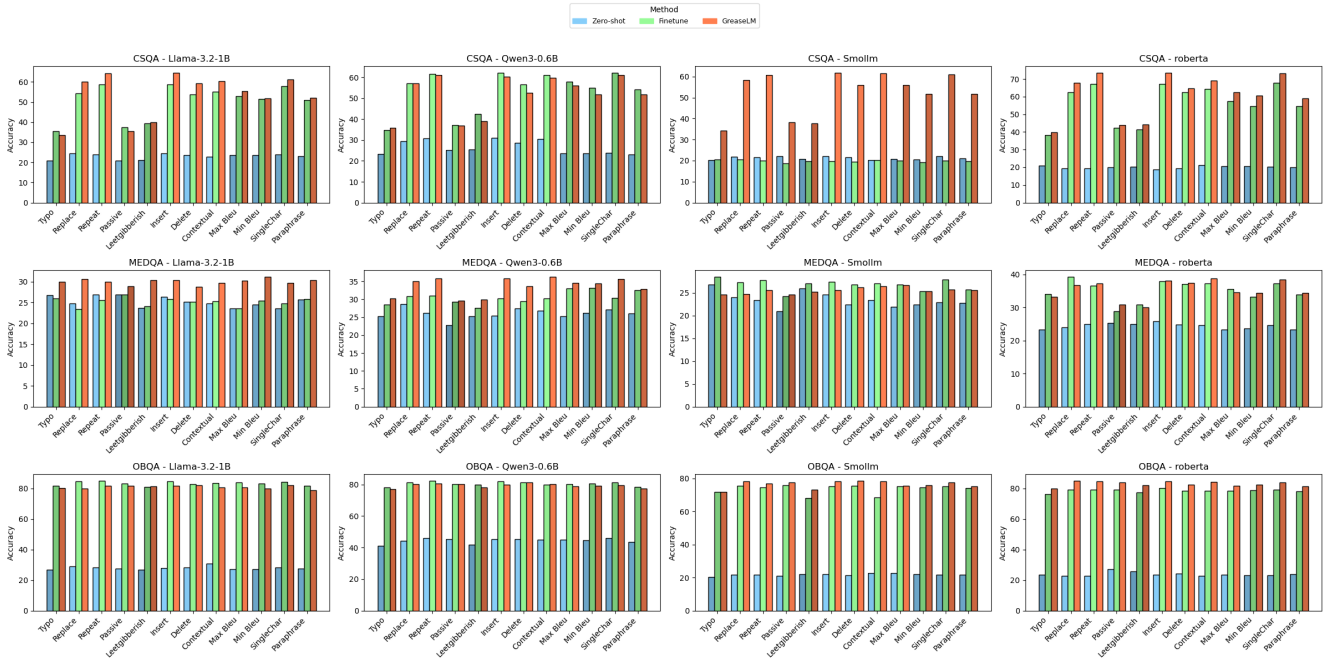

Figure 8: Accuracy Comparison

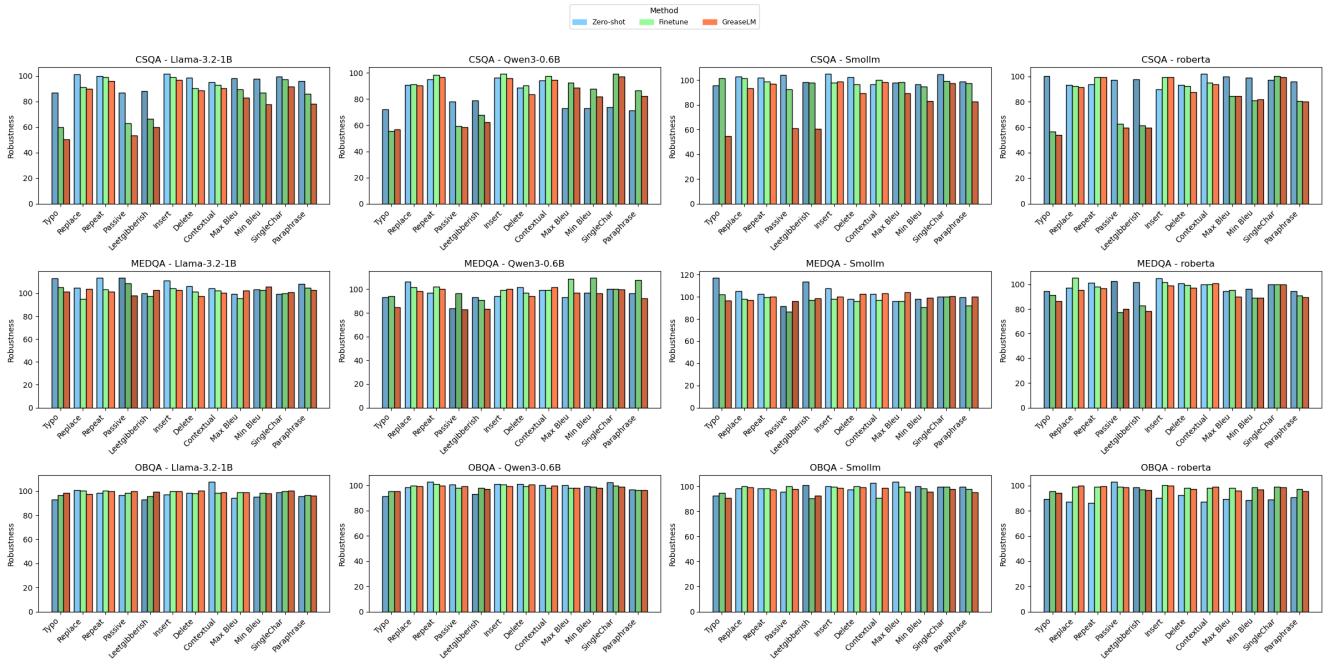

Figure 9: Robustness comparison

Table 3: Performance comparison of RoBERTa-large across CSQA, OBQA, and MedQA under various perturbations

Perturbation	**CSQA**			**OBQA**			**MedQA**		
	Zero-Shot	**Finetune**	**GreaseLM**	**Zero-Shot**	**Finetune**	**GreaseLM**	**Zero-Shot**	**Finetune**	**GreaseLM**
Inhouse Split	20.78	67.68	73.81	26.20	80.00	85.00	24.66	37.31	38.49
Typo	20.79	38.28	39.97	23.40	76.40	80.00	23.25	34.09	33.23
Replace	19.42	62.45	67.69	22.80	79.20	85.00	23.96	39.28	36.68
Repeat	19.50	67.12	73.49	22.60	79.20	84.60	24.98	36.61	37.23
Passive	20.15	42.30	44.00	27.00	79.20	83.80	25.29	28.91	30.79
Leetgibberish	20.31	41.42	44.16	25.80	77.40	82.00	25.06	30.95	30.09
Insert	18.69	67.28	73.49	23.60	80.20	84.80	25.84	37.86	38.18
Delete	19.34	62.37	64.71	24.20	78.60	82.60	24.82	37.08	37.39
Contextual	21.19	64.22	69.06	22.80	78.60	84.20	24.67	37.23	38.81
Max BLEU	20.71	57.29	62.37	23.40	78.40	81.60	23.33	35.66	34.56
Min BLEU	20.55	54.71	60.52	23.20	78.80	82.40	23.72	33.23	34.33
Single Char - 1	20.55	67.85	73.49	24.20	78.60	83.80	24.67	37.31	38.57
Single Char - 2	20.23	68.01	74.05	24.80	79.40	83.20	24.67	37.31	38.41
Single Char - 3	20.55	67.85	73.17	22.00	79.80	84.20	24.67	37.31	38.49
Single Char - 4	19.18	68.17	73.25	22.80	79.00	83.40	24.67	37.31	38.49
Single Char - 5	20.55	67.12	72.28	22.60	79.60	85.00	24.74	37.31	38.57
Paraphrase	19.94	54.50	59.02	23.80	78.00	81.20	23.27	33.91	34.47

	CSQA			OBQA			MedQA		
Perturbation	Zero-shot	Finetune	GreaseLM	Zero-shot	Finetune	GreaseLM	Zero-shot	Finetune	GreaseLM
Inhouse split	24.01	59.31	66.72	28.60	84.60	81.80	23.72	24.74	29.54
Typo	20.87	35.62	33.68	26.60	81.80	80.40	26.79	26.00	29.93
Replace	24.34	54.23	60.03	28.80	84.80	79.80	24.82	23.49	30.64
Repeat	24.01	58.74	64.14	28.20	85.00	81.60	26.94	25.53	29.93
Passive	20.87	37.39	35.54	27.60	83.20	81.80	26.94	26.94	28.91
Leetgibberish	21.19	39.48	39.97	26.60	81.00	81.40	23.72	24.12	30.40
Insert	24.42	58.66	64.54	27.80	84.60	81.60	26.32	25.84	30.40
Delete	23.69	53.67	59.23	28.20	82.80	82.13	25.22	25.14	28.83
Contextual	22.88	55.12	60.35	30.80	83.40	80.80	24.74	25.37	29.69
Max Bleu	23.61	53.02	55.36	27.00	83.80	80.80	23.64	23.64	30.24
Min Bleu	23.53	51.49	51.89	27.20	83.20	80.00	24.51	25.45	31.19
Single Char 1	24.25	58.50	61.56	29.00	84.40	81.20	23.64	24.74	29.77
Single Char 2	23.85	58.58	60.84	28.40	84.60	81.40	23.64	24.82	29.69
Single Char 3	24.09	58.50	63.01	28.40	83.40	81.80	23.64	24.82	29.77
Single Char 4	23.29	55.76	61.32	27.60	84.60	82.80	23.64	24.74	30.01
Single Char 5	23.85	57.61	58.98	28.40	84.60	82.60	23.64	24.82	29.62
Paraphrase	23.03	51.06	52.23	27.32	81.80	78.84	25.66	25.89	30.38

Table 4: Performance of LLaMA-3.2-1B on CSQA, OBQA, and MedQA under various perturbations

	CSQA			OBQA			MedQA		
Perturbation	Zero-shot	Finetune	GreaseLM	Zero-shot	Finetune	GreaseLM	Zero-shot	Finetune	GreaseLM
Inhouse split	32.31	62.61	63.01	45.00	81.80	80.80	27.10	30.32	35.74
Typo	23.29	34.81	35.70	41.00	78.00	77.00	25.22	28.59	30.24
Replace	29.41	57.21	57.05	44.20	81.40	80.20	28.75	30.79	35.11
Repeat	30.78	61.56	61.08	46.20	82.40	80.60	26.24	30.95	35.90
Passive	25.22	37.23	36.91	45.20	80.20	80.20	22.70	29.30	29.62
Leetgibberish	25.54	42.39	39.16	41.80	80.00	78.20	25.29	27.57	29.85
Insert	31.10	62.13	60.44	45.40	82.20	80.00	25.45	30.16	35.82
Delete	28.69	56.49	52.70	45.40	81.20	81.20	27.49	29.38	33.70
Contextual	30.38	61.24	59.71	45.00	80.00	80.40	26.87	30.16	36.29
Max Bleu	23.61	57.94	56.00	45.00	80.20	79.00	25.22	32.99	34.64
Min Bleu	23.53	55.04	51.73	44.60	80.60	79.20	26.24	33.23	34.49
Single Char - 1	24.25	62.85	61.64	45.60	81.60	80.00	27.02	30.32	35.66
Single Char - 2	23.85	62.37	60.84	46.20	81.40	79.00	27.26	30.32	35.74
Single Char - 3	24.09	62.13	61.08	45.60	81.00	80.40	27.26	30.32	35.66
Single Char - 4	23.29	61.48	62.05	46.40	81.80	79.60	27.10	30.32	35.74
Single Char - 5	23.85	62.13	60.35	45.80	81.60	79.60	27.26	30.32	35.74
Paraphrase	23.03	54.17	51.85	43.48	78.60	77.60	26.11	32.62	32.94

Table 5: Performance of Qwen3-0.6B across CSQA, OBQA, and MedQA datasets under various perturbations

	CSQA			OBQA			MedQA		
Perturbation	Zero-shot	Finetune	GreaseLM	Zero-shot	Finetune	GreaseLM	Zero-shot	Finetune	GreaseLM
Inhouse split	21.19	20.23	62.61	22.00	75.60	79.20	22.86	27.89	25.53
Typo	20.23	20.55	34.33	20.40	71.80	71.80	26.79	28.52	24.67
Replace	21.76	20.47	58.34	21.60	75.60	78.40	24.04	27.34	24.74
Repeat	21.60	19.98	60.76	21.60	74.40	77.00	23.41	27.73	25.61
Passive	22.08	18.69	38.28	21.00	75.80	77.60	20.97	24.19	24.59
Leetgibberish	20.79	19.74	37.79	22.20	68.20	73.20	26.00	27.10	25.22
Insert	22.24	19.82	61.80	22.00	75.40	78.20	24.59	27.42	25.53
Delete	21.68	19.50	55.92	21.40	75.60	78.60	22.47	26.87	26.24
Contextual	20.39	20.23	61.56	22.60	68.60	78.20	23.41	27.10	26.39
Max Bleu	20.71	19.90	55.92	22.80	75.40	75.60	21.92	26.87	26.63
Min Bleu	20.47	19.10	51.81	22.00	74.40	75.80	22.39	25.29	25.29
Single Char - 1	22.88	21.35	61.16	22.00	76.40	77.20	22.94	27.97	26.08
Single Char - 2	22.16	19.10	60.60	22.00	75.20	77.20	22.78	27.97	25.77
Single Char - 3	21.84	20.06	61.72	22.00	75.40	79.00	22.94	27.97	25.53
Single Char - 4	21.76	20.39	60.92	22.20	75.60	76.20	22.86	27.89	25.61
Single Char - 5	22.32	19.34	60.19	21.20	74.40	78.80	22.86	27.89	25.77
Paraphrase	20.93	19.66	51.73	21.88	74.12	75.40	22.78	25.72	25.62

Table 6: Performance of SmolLM2-360M on CSQA, OBQA, and MedQA under various perturbations

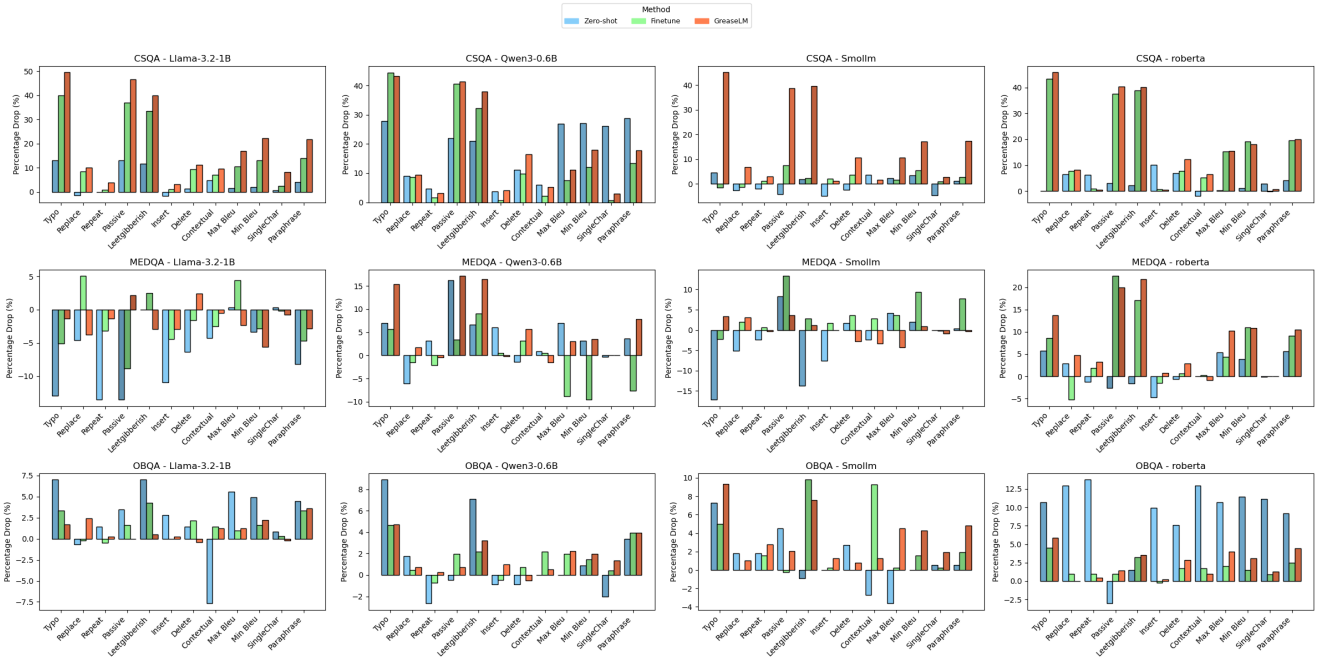

Figure 10: Percentage Performance Drop Comparison