Unveiling the Butterfly Effect in Knowledge Editing for Large Language Models Using Knowledge Graph-based Analysis

Patipon Wiangnak*, Natthawut Kertkeidkachorn and Kiyoaki Shirai

Japan Advanced Institute of Science and Technology, Ishikawa, Japan

Abstract

Large Language Models (LLMs), particularly those based on Generative Pre-trained Transformers (GPT), have achieved strong performance in various natural language tasks. However, LLMs are limited by a knowledge cut-off, so their information is not updated. Common methods for updating LLM knowledge, such as fine-tuning, retrieval-augmented generation, and machine unlearning, are often resource-intensive and may introduce unintended effects, including the loss of relevant context or conflicts with existing knowledge. Knowledge Editing (KE) offers a more efficient alternative by enabling precise updates to specific facts without retraining the entire model, while preserving unrelated information. Still, such edits can trigger unexpected ripple effects, known as the Butterfly Effect, where modifying one fact causes errors in related knowledge. In this work, we introduce ButterflyKE, a knowledge graph-based analysis method that probes neighboring knowledge to identify local side effects caused by a single factual update. Using Wikidata as a reference knowledge graph, in ButterflyKE, we extract directly connected triples to provide a structural view of how knowledge propagates after editing. We evaluate three main KE approaches: External Memory-based, Global Optimization-based, and Local Modification-based approaches, using the `Llama-3.1-8B-Instruct` model. Our findings confirm the presence of the Butterfly Effect in KE, with side effects intensifying as the structural connections increase. To measure this impact, we propose the Butterfly Index, a metric to evaluate editing methods and their influence on surrounding knowledge. ButterflyKE serves as a practical method for extending existing benchmarks and supports a deeper analysis of knowledge integrity in LLM.

Keywords

Large Language Models, Knowledge Editing, Butterfly Effects, Hallucination, Knowledge Graph-Based Analysis

1. Introduction

In this modern era, Large Language Models (LLMs), particularly those based on Generative Pre-trained Transformers (GPT), have revolutionized various fields, including Question Answering, Machine Translation, and Natural Language Inference (NLI). Nevertheless, as black-box models, the complexity of LLMs presents challenges, as their limited by a knowledge cut-off, so their information is not updated. In recent years, Knowledge Editing (KE) [1] has emerged as a promising alternative for updating knowledge in LLMs without full retraining or harming unrelated information. However, LLM knowledge is often sensitive to edits, and a single update can introduce unintended consequences [2]. We define this phenomenon as the **Butterfly Effect**, where one edit disrupts related knowledge. While recent work focuses on making edits more accurate and precise, limited attention has been given to evaluating potential side effects. Current KE methods can be broadly categorized into three strategies:

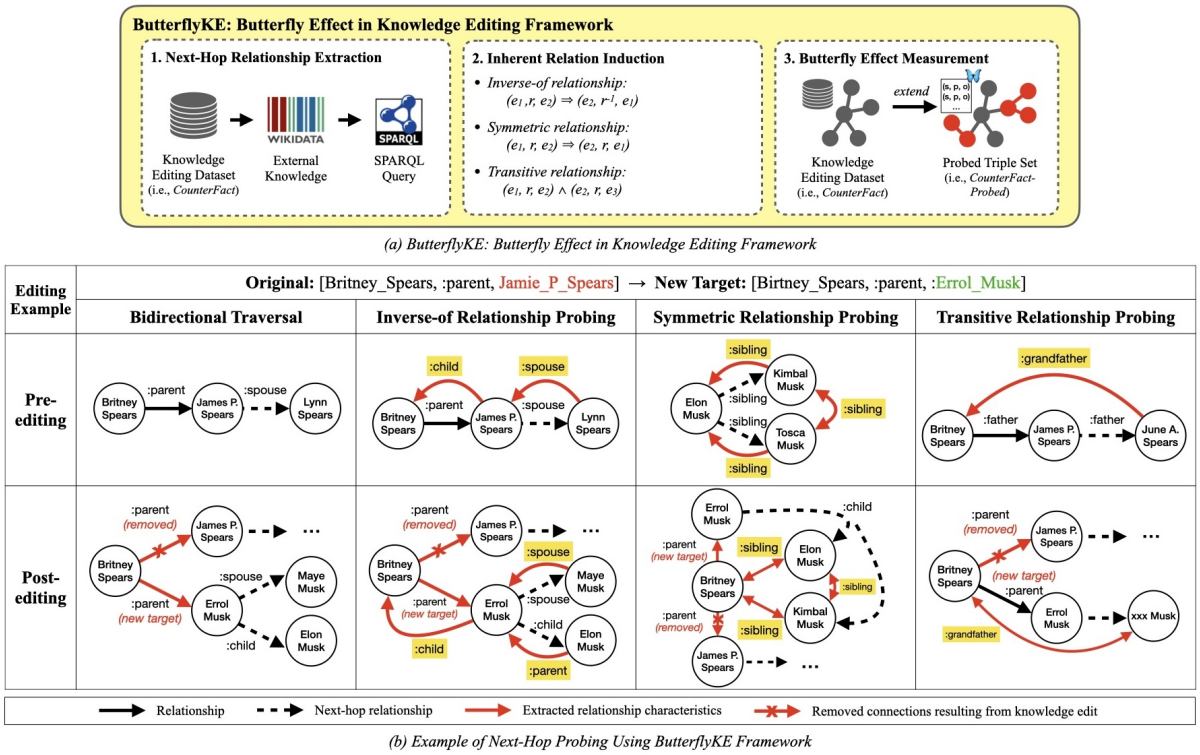
1. **External Memory-based Approach**: Stores new knowledge externally without changing internal weights, such as RAG and In-context Knowledge Editing (IKE) [3].
2. **Global Optimization-based Approach**: Updates the model using gradients from new knowledge, such as Model Editor Networks with Gradient Decomposition (MEND) [4].
3. **Local Modification-based Approach**: Locates and updates only specific parameters related to the target fact, such as Rank-One Model Editing (ROME) [5].

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan

*Corresponding author.

w.patipon@jaist.ac.jp (P. Wiangnak); natt@jaist.ac.jp (N. Kertkeidkachorn); kshirai@jaist.ac.jp (K. Shirai)

0009-0007-2509-433X (P. Wiangnak); 0000-0003-4527-776X (N. Kertkeidkachorn); 0009-0009-7591-2155 (K. Shirai)



(a) ButterflyKE: Butterfly Effect in Knowledge Editing Framework

(b) Example of Next-Hop Probing Using ButterflyKE Framework

Figure 1: Overview of the ButterflyKE framework and next-hop probing example.

In this work, we introduce **ButterflyKE: Butterfly Effects in Knowledge Editing**, a method designed to systematically probe the side effects of knowledge editing in LLMs. By leveraging structured knowledge from knowledge graph, it traces how a single factual update may propagate through semantically connected facts. We evaluate these effects using the proposed **Butterfly Index**, which quantifies the model's ability to maintain factual correctness in next-hop neighboring knowledge.

2. Butterfly Effect in Knowledge Editing for Large Language Models

To identify and interpret the side effects of Knowledge Editing in Large Language Models, we introduce **ButterflyKE**, a knowledge graph-based framework that probes next-hop neighboring knowledge to detect localized effects from a single factual update. Instead of constructing a new dataset, we use the public knowledge graph such as Wikidata to enable structural reasoning and trace how edits can propagate through semantically connected facts in the model's internal knowledge. Figure 1 (a) illustrates the three main components of the framework, while Figure 1(b) illustrates an example of next-hop probing across different types of relationships. In this figure, solid black arrows indicate original triples, dashed arrows represent next-hop connections retrieved from the knowledge graph, red arrows denote induced triples inferred from ontological properties, and red arrows marked with a cross indicate removed connections resulting from knowledge edit.

1. **Next-Hop Relation Extraction**: Given an edit instance in the form of a triple (subject, predicate, object), we first retrieve its adjacent triples from an external knowledge graph using SPARQL queries. Here, adjacency refers to triples that share either the subject entity or the object entity with the edited triple. For example, for the triple [Britney_Spears, :parent, Jamie_P_Spears], adjacent triples include [Jamie_P_Spears, :spouse, Lynn_Spears] and [Britney_Spears, :sibling, Jamie_Lynn_Spears]. This step defines the immediate structural context in which the edit may have side effects.
2. **Inherent Relation Induction**: After performing the edit by editing the original target entity with a new entity, we expand the neighborhood by enforcing inherent relation constraints derived

Table 1
Probing statistics for *CounterFact* using the ButterflyKE framework.

Configuration	#Triple	#Entity	#Property
CounterFact	10,000	862	34
CounterFact-Probed	27,495	5,197	166

from ontological properties. Specifically, we consider: *inverse relations*, if (e_1, r, e_2) holds, then (e_2, r^{-1}, e_1) should also hold; *symmetric relations*, if (e_1, r, e_2) holds, then (e_2, r, e_1) should also hold; *transitive relations*, if (e_1, r, e_2) and (e_2, r, e_3) hold, then (e_1, r, e_3) can be inferred. For example, if Britney is edited to have Errol Musk as a parent, the induced inverse relation makes Britney a child of Errol; if Elon is a sibling of Kimbal, then Kimbal must also be a sibling of Elon; and if Errol is the parent of Elon and Elon is the parent of another entity, then Errol becomes the grandparent of that entity. These induced triples represent the logical consequences of the edit that may introduce inconsistencies or propagate across unrelated domains.

3. **Butterfly Effect Measurement**: We aims to probe the impact of a knowledge edit using factual questions derived from next-hop knowledge and inherent relations identified in previous steps. After performing the edit, the model is queried to observe changes in its responses. For example, from [Britney_Spears, :parent, Errol_Musk] we ask "Who is the parent of Britney Spears?", and from the induced inverse relation [Errol_Musk, :child, Britney_Spears] we ask "Who is the child of Errol Musk?". By comparing the model's answers before and after the edit, we identify discrepancies in correctness that signal local disruptions.

To evaluate the side effects of knowledge editing on semantically related information, we introduce the **Butterfly Index** (Equation 1). This metric quantifies the degradation in factual accuracy on next-hop knowledge due to the edit by comparing the model's answers before and after the update.

$$\text{ButterflyIndex} = \frac{1}{n}\sum_{i=1}^{n}\left[\mathbb{1}(f_{\text{orig}}(q_i) = g_i) - \mathbb{1}(f_{\text{edit}}(q_i) = g_i)\right] \tag{1}$$

Here, f_{orig} and f_{edit} denote the language model before and after editing, respectively; q_i is the factual question derived from the i-th probed triple; g_i is the ground truth answer; and $\mathbb{1}(\cdot)$ is the indicator function, returning 1 if the answer is correct and 0 otherwise. A higher **Butterfly Index** reflects a greater loss in accuracy on neighboring knowledge, thereby indicating stronger unintended side effects of the edit.

3. Experiment

3.1. Experimental Setup

Table 1 presents the probing statistics for the original *CounterFact* dataset, a benchmark for evaluating knowledge editing in LLMs that tests whether a model can internalize edits while preserving unrelated knowledge [5], as well as our extended version, *CounterFact-Probed*, constructed using the **ButterflyKE** approach. By incorporating next-hop neighboring triples, the extended setting substantially increases the number of entities and relations, enabling a more thorough evaluation of local side effects in knowledge editing. These additional triples are not part of the core dataset but are dynamically generated to probe the model's behavior after an edit. In total, over 20,000 such triples are used to assess the impact of updates on semantically related knowledge.

All experiments are conducted using the `LLaMA-3.1-8B-Instruct` model as the backbone. As shown in Table 2, we evaluate representative KE-for-LLMs methods under two configurations: the original *CounterFact*, which tests whether the target fact is correctly internalized, and the extended *CounterFact-Probed*, which measures unintended effects on adjacent knowledge captured via the **ButterflyKE** approach.

Table 2
Performance of knowledge editing methods on edited and probed knowledge using the Butterfly Index.

KE-for-LLMs Approaches	Method	Accuracy		Butterfly Index
		CounterFact	*CounterFact-Probed*	
External Memory-based	IKE	1.0	0.511	0.489
Global Optimization-based	MEND	0.903	0.522	0.381
Local Modification-based	ROME	0.87	0.26	0.61

3.2. Results and Discussion

Table 2 presents the performance of representative KE-for-LLMs approaches evaluated on the original *CounterFact* dataset and its probed counterpart, *CounterFact-Probed*, which includes next-hop neighboring triples generated using the **ButterflyKE** framework. All methods achieve high accuracy on *CounterFact*, confirming their effectiveness at injecting and retrieving the edited knowledge. However, when evaluated on *CounterFact-Probed*, accuracy drops significantly across all methods, revealing local inconsistencies is introduced by the edit. This degradation is quantified by the **Butterfly Index**, which measures the difference in accuracy before and after editing on next-hop knowledge. For example, IKE achieves a perfect editing accuracy of 1.0, but its accuracy on neighboring facts drops to 0.511, resulting in a Butterfly Index of 0.489. Similarly, ROME drops from 0.87 to 0.26, yielding the highest Butterfly Index of 0.61 among all methods. These results indicate that although edits are successful in isolation, they often disrupt related factual knowledge embedded within the model.

These findings demonstrate the presence of the **Butterfly Effect** in knowledge editing. A localized factual change can unintentionally affect semantically related information within the model. This observation reveals a key limitation of current knowledge editing techniques, which often fail to maintain the broader contextual consistency of the model's internal knowledge. The **Butterfly Index** helps bridge this gap by offering a principled metric that captures not only the factual accuracy but also the semantic stability of the model after a edit.

4. Conclusion

In this study, we presented **ButterflyKE**, a framework to evaluate local side effects of KE-for-LLMs. By enriching the *CounterFact* dataset with next-hop neighboring triples, we constructed *CounterFact-Probed*, enabling probing of unintended impacts on semantically related knowledge. To quantify these effects, we proposed the **Butterfly Index**, measuring accuracy differences on surrounding facts before and after editing. Experimental results show that while KE methods succeed in updating the target information, they vary significantly in their ability to preserve adjacent facts. In particular, they experience substantial drops in accuracy on neighboring triples, revealing local disruptions despite successful edits. These findings confirm the presence of the **Butterfly Effect** in KE. This highlights a key limitation of current approaches and emphasizes the need for methods that ensure both factual precision and semantic stability. In future work, we will broaden the evaluation to diverse editing techniques and domains, and extend analysis to advanced foundation models such as ChatGPT, DeepSeek, and Gemini. We also plan to investigate deeper graph structures and multi-hop interactions to better understand interference mechanisms and guide the design of more robust editing strategies.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4o to: Grammar, paraphrase, and reword. After using this tool, the authors reviewed and edited the content as needed and assumed full responsibility for the publication's content.

References

[1] S. Wang, Y. Zhu, H. Liu, Z. Zheng, C. Chen, J. Li, Knowledge Editing for Large Language Models: A Survey, ACM Comput. Surv. 57 (2024) 59:1–59:37. URL: https://dl.acm.org/doi/10.1145/3698590. doi:10.1145/3698590.

[2] Z. Li, N. Zhang, Y. Yao, M. Wang, X. Chen, H. Chen, Unveiling the Pitfalls of Knowledge Editing for Large Language Models, 2024. URL: http://arxiv.org/abs/2310.02129. doi:10.48550/arXiv.2310.02129, arXiv:2310.02129 [cs].

[3] C. Zheng, L. Li, Q. Dong, Y. Fan, Z. Wu, J. Xu, B. Chang, Can We Edit Factual Knowledge by In-Context Learning?, 2023. URL: http://arxiv.org/abs/2305.12740. doi:10.48550/arXiv.2305.12740, arXiv:2305.12740 [cs].

[4] E. Mitchell, C. Lin, A. Bosselut, C. Finn, C. D. Manning, Fast Model Editing at Scale, 2022. URL: http://arxiv.org/abs/2110.11309. doi:10.48550/arXiv.2110.11309, arXiv:2110.11309 [cs].

[5] K. Meng, D. Bau, A. Andonian, Y. Belinkov, Locating and Editing Factual Associations in GPT, 2023. URL: http://arxiv.org/abs/2202.05262. doi:10.48550/arXiv.2202.05262, arXiv:2202.05262 [cs].