Understanding the Side Effects of Rank-One Knowledge Editing

Ryosuke Takahashi[1,2] **Go Kamoda**[3,4]
Benjamin Heinzerling[2,1] **Keisuke Sakaguchi**[1,2] **Kentaro Inui**[5,1,2]
[1]Tohoku University [2]RIKEN [3]SOKENDAI [4]NINJAL [5]MBZUAI
ryosuke.takahashi@dc.tohoku.ac.jp go.kamoda@ninjal.ac.jp
benjamin.heinzerling@riken.jp keisuke.sakaguchi@tohoku.ac.jp
kentaro.inui@mbzuai.ac.ae

Abstract

This study conducts a detailed analysis of the side effects of rank-one knowledge editing using language models with controlled knowledge. The analysis focuses on each element of knowledge triples (subject, relation, object) and examines two aspects: “knowledge that causes large side effects when edited” and “knowledge that is affected by the side effects.” Our findings suggest that editing knowledge with subjects that have relationships with numerous objects or are robustly embedded within the LM may trigger extensive side effects. Furthermore, we demonstrate that the similarity between relation vectors, the density of object vectors, and the distortion of knowledge representations are closely related to how susceptible knowledge is to editing influences. The findings of this research provide new insights into the mechanisms of side effects in LM knowledge editing and indicate specific directions for developing more effective and reliable knowledge editing methods.

1 Introduction

Language models (LMs) can store knowledge in their internal parameters through training and research focusing on analyzing the knowledge stored inside LMs have gained attention (Petroni et al., 2019; Jiang et al., 2020; Heinzerling and Inui, 2021; AlKhamissi et al., 2022). Although this capability is essential in building human-aiding assistants, challenges related to reliability and safety are also reported. For example, LMs possess knowledge only up to the point when their training data was collected, making them not robust to the constantly changing real-world knowledge (De Cao et al., 2021; Mitchell et al., 2022a; Kasai et al., 2023). Additionally, there is a risk that LMs could leak personal and confidential information contained in the training data, raising privacy concerns (Huang et al., 2022; Jang et al., 2023). To address these challenges, several studies have been conducted on *knowledge editing* (Feng et al., 2023; Zhang et al., 2024; Dai et al., 2022; Meng et al., 2022, 2023; Li et al., 2023) and *knowledge deletion* (Jang et al., 2023; Ishibashi and Shimodaira, 2023; Trippa et al., 2024; Wang et al., 2025). While these studies have reported some success in knowledge editing and deletion, they have identified problems with side effects caused by the editing process.

This study investigates the mechanisms underlying side effects in knowledge editing from multiple perspectives. Our hypothesis posits that side effects arise from the intrinsic characteristics of knowledge triple components (subject, relation, object). Based on this hypothesis, we focus on revealing: 1) characteristics of knowledge that cause large side effects when edited and 2) characteristics of knowledge that are affected by the side effects.

In analyzing knowledge that causes large side effects when edited, we focus on two properties of subjects: connectivity and embedding robustness. Highly connected subjects maintain numerous relationships with multiple objects through diverse relations, suggesting that modifications to these subjects are likely to produce widespread side effects. Additionally, when subjects are deeply embedded in the model’s internal representations, the substantial changes required for editing may propagate effects to other connected knowledge.

Regarding knowledge that is affected by the side effects, we focus on relation similarity, object space density, and knowledge distortion. Knowledge involving relations that are semantically similar to the relation in the edited knowledge instance is likely to be susceptible to side effects. In areas where object representations are densely clustered, even minor modifications can lead to unintended object substitutions. Moreover, when knowledge instances are distorted in the internal representation space, minor edits might trigger substantial variations in model outputs.

In this research, we quantitatively evaluate the

Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pages 189–205
November 9, 2025 ©2025 Association for Computational Linguistics

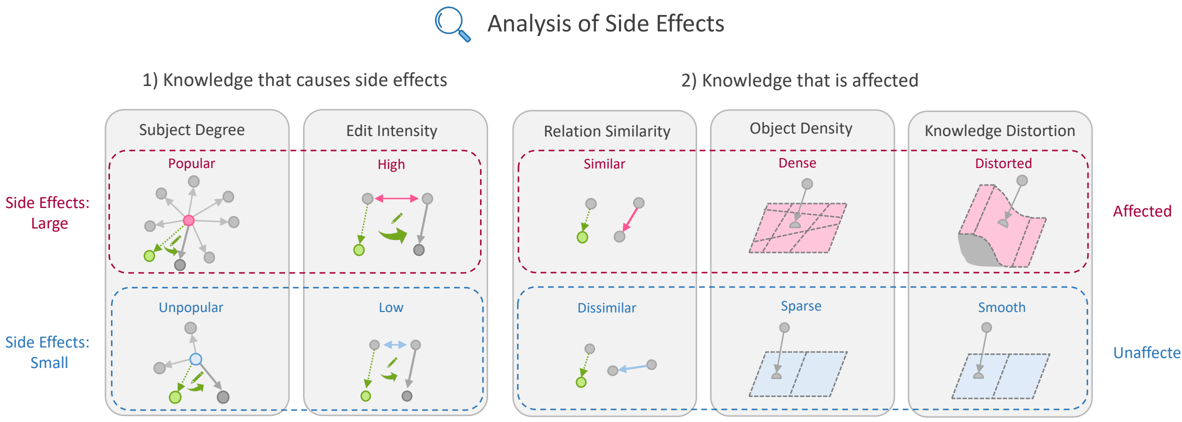

Figure 1: **Overview of our systematic analysis of knowledge editing side effects.** We analyze the side effects of knowledge editing from two perspectives: knowledge that causes large side effects when edited, focusing on subject degree and edit intensity, and knowledge that is affected by side effects,examining relation similarity, object density, and knowledge distortion.

effects and side effects of applying conventional editing methods using artificial knowledge graphs. By examining how each factor contributes to side effects, we systematically categorize the factors causing knowledge editing side effects and empirically demonstrate their impact.

2 Related Work

2.1 Knowledge Editing

Traditional paradigms for modifying the internal knowledge and the behavior of language models include supervised fine-tuning (SFT), reinforcement learning from human feedback (RLHF; Ouyang et al., 2022), and direct preference optimization (DPO; Rafailov et al., 2023), among others. These methods update models' knowledge via re-training and involve challenges such as the high cost of collecting training data and difficulties with generalization. In response, more efficient and flexible methods for knowledge updating are being investigated, and knowledge editing is receiving increasing attention.

Knowledge editing approaches can be broadly divided into two main categories. The first are methods that adjust outputs using external knowledge bases, exemplified by SERAC (Mitchell et al., 2022b) and T-patcher (Huang et al., 2023).

The second are methods that directly modify model parameters, such as FT-L (Zhu et al., 2020), KE (De Cao et al., 2021), MEND (Mitchell et al., 2022a), ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), and AlphaEdit (Fang et al., 2025), among others.

2.2 Side Effect of Knowledge Editing

Knowledge editing, such as directly modifying model parameters, is fundamentally a method to improve the factuality of the model. However, this approach has the side effect of unintentionally and significantly degrading the model's general capabilities (Gu et al., 2024).

The challenges of knowledge editing have been reported from multiple perspectives:

1. Continuous forgetting: Even a single knowledge edit can potentially cause model collapse and the ability to perform downstream tasks is also lost (Yang et al., 2024; Gupta et al., 2024).
2. Impact on neighbor knowledge: Investigations into the effects of model knowledge updates on adjacent knowledge have revealed that while new knowledge can be effectively added, there is also a problem of forgetting existing correct adjacent knowledge or unintentionally adding incorrect knowledge (Ma et al., 2024).
3. Lack of ripple effects: When updating certain knowledge, the changes should appropriately affect other related knowledge. However, it has become clear that current knowledge editing methods cannot consistently achieve such cascading updates (Onoe et al., 2023; Cohen et al., 2024).
4. Overfit: Through editing, models tend to learn excessively strong associations between the input prompt and the target object. As a result, they output the target object with inappropri-

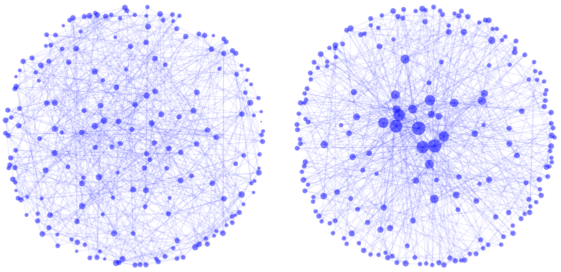

Figure 2: Visualization of two synthetic knowledge graphs: an Erdős-Rényi graph (left) , and a Barabási-Albert graph (right).

ately high probability in response to complex questions (Zhang et al., 2025).

3 Experimental Setup

3.1 Knowledge Graphs

This work addresses relational knowledge represented in triples, such as (s, r, o). We define a knowledge graph as a representation where the subject s and object o correspond to vertices, and the relation r corresponds to an edge, as a knowledge graph. Here, s and o are elements of the entity set $\mathcal{E}$ with $|\mathcal{E}| = 200$, and r is an element of the relation set $\mathcal{R}$ with $|\mathcal{R}| = 50$.

Previous studies on knowledge editing have assumed the existence of knowledge graphs composed of relational knowledge between entities expressed in natural language (Zhu et al., 2020; Meng et al., 2022; Fang et al., 2025). This study introduces an approach by creating a synthetic knowledge graph, allowing for precise control over the information LMs acquire through training. We created two synthetic knowledge graphs with different characteristics (Figure 2). The first is an Erdős-Rényi (ER) graph (Erdös and Rényi, 1959), structured to ensure the probability of forming edges between vertices is uniform. The second is a Barabási-Albert (BA) graph (Barabási and Albert, 1999), characterized by a vertex degree distribution that follows a power law, thus resembling the structure of the real world more closely. This graph, with varied vertex degrees, enables analyses of the relation between the degree of each entity (vertex) and the side effects of knowledge editing.

3.2 Storing Knowledge Graphs in LMs

We first assign five names to each entity e^i ($0 \leq i < |\mathcal{E}|\,; i \in \mathbb{N}$), denoted as e^i_j ($0 \leq j < 5\,; j \in \mathbb{N}$). Similarly, we assign five names to each relation r^i ($0 \leq i < |\mathcal{R}|\,; i \in \mathbb{N}$), denoted as r^i_j ($0 \leq j < 5\,; j \in \mathbb{N}$) and include these names in the model vocabulary and tokenizer. Hereafter, names referring to the same entity (or relation) are termed “paraphrases.” We then create a corpus composed of sequenecs each with three tokens (e.g., “e^0_0 r^1_1 e^1_4”) using the synthetic knowledge graphs created in Section 3.1, and train models with GPT-2 architecture with 6, 12, and 24 layers (Radford et al., 2019; Sanh et al., 2019).

Knowledge Graph	Model	Accuracy on Training Data	Accuracy on Full Data
ER	6 layers	0.9998	0.9941
	12 layers	0.9985	0.9983
	24 layers	0.9900	0.9905
BA	6 layers	0.9991	0.9987
	12 layers	0.9991	0.9989
	24 layers	0.9972	0.9964

Table 1: Accuracy of training data and full data for each model after training the RA graph and the BA graph.

During inference, we input two tokens into the model, and the model predicts one token. A prediction is correct if the predicted token represents any paraphrases indicating the gold entity. During training, we use 20% of the entire knowledge base, which includes paraphrased knowledge, intending to achieve generalization across all paraphrased knowledge. After training, the model achieves an accuracy rate of approximately 99% not only on the training data but also on the full data, indicating that it has successfully memorized the provided knowledge (see Table 1). Additionally, by principal component analysis on the word embeddings before and after training, it was suggested that appropriate embeddings were acquired after training (for details, see Appendix A).

3.3 Knowledge Editing Method

While our analysis framework is agnostic to other knowledge editing methods, we opt for ROME (Meng et al., 2022), one of the primary knowledge editing methods for causal LMs, in this work. ROME attempts to edit knowledge by updating model weight through the following two steps.

Step 1: Causal Tracing The first step is to identify a model component that plays a crucial role in knowledge association. This is achieved by analyzing the contribution of each hidden state of the model to the prediction regarding the target knowledge[1] (for details, see Appendix B). The results of Causal Tracing revealed that the initial Feed-Forward (FF) layer significantly contributes to knowledge association, supporting research indicating that FF layers serve as key-value memory storage for knowledge (Geva et al., 2021).

Step 2: Rank-One Model Editing The second step is to add a rank-1 matrix to the weights W_2 of the FF layer (defined by Equation 1 where σ is the activation function), which was identified as most contributing to the prediction through Causal Tracing.

$$\mathrm{FFN}(x) = W_2\,\sigma(W_1 x + b_1) + b_2 \tag{1}$$

In the ROME approach, W_2 is conceptualized as an associative memory for existing key-value pairs (K, V)[2], and perform edits to insert a new key-value pair (k_*, v_*). This editing objective can be formulated as a constrained least-squares optimization problem. The solution yields the updated weight matrix $\hat{W}_2$, which takes the form:

$$\hat{W}_2 = W_2 + \Lambda(C^{-1}k_*)^\top \tag{2}$$

Here, $C = KK^\top$ represents the uncentered covariance of K, and Λ is a vector proportional to the residual error of the new key-value pair (k_*, v_*).

3.4 Knowledge Editing from LMs

Knowledge editing can be treated as reassociating an entity with a different entity from one already connected. In this work, we introduce a new entity e_{target} as the target object entity to be newly associated, and formalize knowledge editing as the process of updating the target knowledge instance (s, r, o) such that s, which was originally associated with o via r, is instead associated with e_{target}.

4 Evaluation of Knowledge Editing

4.1 Evaluation Metrics

To comprehensively evaluate knowledge editing, we introduce three categories of metrics: Efficacy, Generalization, and Specificity. These metrics aim to assess not only the direct effectiveness of editing but also its broader implications on the model's behavior.

Efficacy measures how successfully the model incorporates the newly edited knowledge. We define two metrics in this category:

- Efficacy Score (ES): The proportion of cases where the probability of the target object (e_{target}) exceeds that of the original object (o) after editing, i.e., $\mathbb{P}[e_{\text{target}}] > \mathbb{P}[o]$.
- Efficacy Match (EM): The proportion of cases where the model outputs the target object (e_{target}) as its primary prediction after editing.

Generalization evaluates whether the editing effects extend to paraphrased versions of the same knowledge. This is measured through:

- Paraphrase Score (PS): Similar to ES, but calculated using prompts (s, r_p) where r_p is a paraphrase of the original relation r.
- Paraphrase Match (PM): Similar to EM, but for paraphrased prompts.

Specificity assesses how localized the editing effects are by examining impacts on neighboring knowledge. We consider two types of neighborhood relationships. Let us define **subject-sharing knowledge** as knowledge instances that share the same subject as the edited knowledge. For these subject-sharing knowledge instances:

- Subject Sharing Score (Subj. SS): The proportion of instances where $\mathbb{P}[o] > \mathbb{P}[e_{\text{target}}]$.
- Subject Sharing Match (Subj. SM): The proportion of instances where the original knowledge is maintained.

Similarly for **relation-sharing knowledge**, defined as knowledge instances that share the same relation as the edited knowledge:

- Relation Sharing Score (Rel. SS): The proportion of instances where $\mathbb{P}[o] > \mathbb{P}[e_{\text{target}}]$.
- Relation Sharing Match (Rel. SM): The proportion of instances where the original knowledge is maintained.

We also compute the harmonic mean of ES, PS, Subj. SS, and Rel. SS to capture the overall performance balancing these aspects, which we denote as **Score** (S). This combined metric helps evaluate the trade-off between effective editing (Efficacy), robust generalization (Generalization), and minimal side effects (Specificity).

In all metrics, larger values indicate small side effects.

4.2 Quantitative Results

Table 2 shows the quantitative evaluation results of knowledge editing across different model architectures and knowledge graph structures. The results reveal several vital patterns in how ROME affects different aspects of model behavior.

[1] Here, predicting o for an input (s, r) is referred to.

[2] $K = [k_1|k_2|\ldots]$ and $V = [v_1|v_2|\ldots]$, where k and v represent vectors.

Knowledge Graph	Model	Efficacy		Generalization		Specificity				Score
		ES ↑	EM ↑	PS ↑	PM ↑	Subj. SS ↑	Subj. SM ↑	Rel. SS ↑	Rel. SM ↑	S ↑
ER	6 layers	0.9877	0.9824	0.9635	0.9508	0.7676	0.5724	1.0000	0.9996	0.9189
	12 layers	0.9947	0.9912	0.6134	0.4517	0.8982	0.3958	1.0000	0.9975	0.8423
	24 layers	0.9982	0.9965	0.4490	0.1727	0.8398	0.3592	1.0000	0.9905	0.7381
BA	6 layers	1.0000	1.0000	0.8790	0.7657	0.8270	0.4757	1.0000	0.9986	0.9202
	12 layers	0.9915	0.9882	0.8574	0.7234	0.8225	0.2482	1.0000	0.9993	0.9110
	24 layers	0.9729	0.9509	0.9264	0.7593	0.4892	0.1653	1.0000	0.9980	0.7765

Table 2: Quantitative Evaluation of Knowledge Editing: Comparison of editing effectiveness metrics, including efficiency measures (ES, EM), generalization to paraphrases (PS, PM), and retention of subject-sharing (Subj. SS, SM) and relation-sharing (Rel. SS, SM) knowledge. The overall Score represents the harmonic mean of ES, PS, Subj. SS, and Rel. SS.

First, regarding Efficacy, we observe consistently high performance across all configurations, with ES and EM exceeding 97% in most cases. This indicates that ROME successfully modifies the target knowledge regardless of model size or knowledge graph structure. The slight decrease in efficiency metrics for larger models (24 layers) in the BA graph suggests that knowledge editing becomes marginally more challenging as model complexity increases.

Generalization performance exhibits considerable variation across model configurations. For the ER graph, we observe an apparent degradation in generalization ability as the number of layers increases, with PS dropping from 96.35% (6 layers) to 44.90% (24 layers). Interestingly, the BA graph maintains more robust generalization across model sizes, with PS remaining above 85% even in larger models.

The Specificity metrics reveal a striking asymmetry in how knowledge editing affects subject-sharing and relation-sharing knowledge. Across all configurations, relation-sharing knowledge is exceptionally well preserved (Rel. SS and Rel. SM consistently near 100%). However, subject-sharing knowledge shows substantially lower preservation rates, particularly in Subject Sharing Match (Subj. SM). This effect becomes more pronounced in larger models, with Subj. SM dropping to 16.53% in the 24-layer BA graph model.

When comparing the ER and BA graphs, we observe that while both achieve similar overall Scores for smaller models, the BA graph maintains better performance in larger architectures, particularly in terms of generalization. This suggests that the more realistic power-law structure of the BA graph might facilitate more robust knowledge representations.

These results highlight a fundamental challenge in knowledge editing. While ROME can effectively edit specific knowledge and maintain relation-sharing knowledge, it significantly impacts other knowledge associated with the edited subject, particularly in deeper models. This observation motivates our subsequent analysis of the factors influencing these subject-related side effects.

4.3 Analysis of Weight Update Matrix

Our quantitative results in Section 4.2 demonstrated that knowledge editing predominantly affects subject-sharing knowledge instances. We further investigate this phenomenon by analyzing the weight update matrix $\Delta W = \Lambda(C^{-1}k_*)^\top \in \mathbb{R}^{d\times 4d}$[3] that is added to the W_2 of FFN by ROME. This weight update affects model inference through the transformation $\Delta W k$, where $k \in \mathbb{R}^{4d}$ is the input vector to the second linear transformation in the FF layer.

We investigate the effect of ΔW on the output of FFN by analyzing the singular value decomposition of ΔW, inspired by (Millidge and Black, 2022). Since ROME performs rank-one updates, ΔW can be decomposed as $\Delta W = \sigma u v^\top$, where $\sigma \in \mathbb{R}$ denotes the singular value, $u \in \mathbb{R}^d$ is the left singular vector, and $v \in \mathbb{R}^{4d}$ is the right singular vector. We quantify the impact of the weight update by the *update magnitude*, defined as $m = \sigma v^\top k$. This scalar value quantifies how strongly the update ΔW affects the output of FFN.

In Table 3, we show update magnitudes averaged over subject-sharing knowledge and relation-sharing knowledge. The results demonstrate consistently higher update magnitudes for subject-sharing knowledge across all model configurations, suggesting that the weight updates are inherently structured to have a stronger impact on knowledge in-

[3] d denotes the model's hidden dimension.

Knowledge Graph	Model	Update Magnitude m	
		Subj. Sharing Knowledge	Rel. Sharing Knowledge
ER	6 layers	**19.1609**	3.7725
	12 layers	**8.4228**	5.3302
	24 layers	**7.3184**	5.6542
BA	6 layers	**6.3663**	3.0569
	12 layers	**5.8021**	3.8179
	24 layers	**5.8738**	4.7027

Table 3: Comparison of average update magnitude for subject-sharing knowledge instances and relation-sharing knowledge instances.

stances that share the same subject.

This analysis of the weight update matrix provides mechanistic evidence for why ROME exhibits a stronger effect on subject-sharing knowledge instances, corroborating our empirical findings from the quantitative evaluation.

5 Analysis of Side Effects

Our analysis in Section 4 reveals that knowledge editing primarily affects subject-sharing knowledge instances. In light of these findings, we examine two key aspects: 1) the characteristics of subject entities that correlate with side effects (Sections 5.1 and 5.2), and 2) the distinguishes factors between knowledge instances that are affected by side effects from those that are not (Sections 5.3 to 5.5).

5.1 Impact of Subject Degree

Procedure We investigate how a subject's degree in the knowledge graph correlates with the magnitude of editing side effects. To quantify the impact of editing knowledge related to a subject s, we define the impact measure $\mathrm{I}(s)$ as:

$$\mathrm{I}(s) = \mathrm{acc}_{\mathrm{pre}} - \mathrm{acc}_{\mathrm{post}}(s) \tag{3}$$

where $\mathrm{acc}_{\mathrm{pre}}$ represents the accuracy before editing and $\mathrm{acc}_{\mathrm{post}}(s)$ denotes the accuracy after editing knowledge with subject s. Using this measure, we examine the relationship between the magnitude of side effects and the subject's degree in the knowledge graph.

Results Figure 3 illustrates the relationship between entity degrees and editing impact $I(\cdot)$ in the 6-layer model trained on ER and BA graphs. We observe no significant correlation between entity degrees and impact for the ER graph model, indicated by a relatively flat trend line. In contrast, the BA graph model exhibits a clear positive correlation: entities with higher degrees show larger impact values, while those with lower degrees show smaller impacts. This pattern persists across models with different numbers of layers. Table 4 shows the Spearman correlation coefficients between entity degrees and their impact across different model architectures. The ER graph maintains low correlation coefficients for all model sizes (6, 12, and 24 layers), while the BA graph consistently shows significant positive correlations. These results suggest that in knowledge graphs with power-law degree distributions (such as the BA graph), which better reflect real-world knowledge structures, editing knowledge related to highly connected entities leads to more extensive side effects. Conversely, editing knowledge about entities with fewer connections results in more localized changes. This relationship is not observed in the ER graph, where the uniform degree distribution results in more consistent editing impacts regardless of entity connectivity.

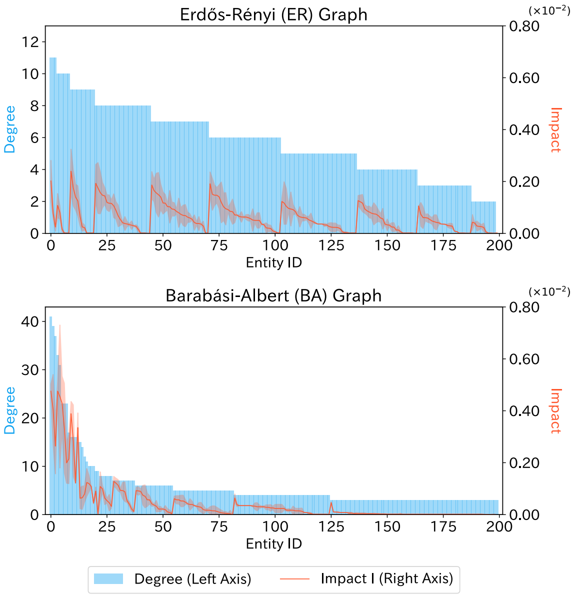

Figure 3: Analysis of the relationship between subject degrees and editing impacts in a 6-layer model trained on the knowledge graph.

5.2 Impact of Edit Intensity

Procedure We examine the relationship between editing side effects and the magnitude of change in subject representations caused by editing operations. We introduce Edit Intensity (EI) for this analysis, quantifying the extent of representation change. EI is defined as the Euclidean distance

Knowledge Graph	Model	Degree		Edit Intensity	
		ρ	p-value	ρ	p-value
ER	6 layers	0.2623	0.0002	0.3135	0.0000
	12 layers	0.3155	0.0000	0.3027	0.0000
	24 layers	0.3300	0.0000	0.3021	0.0000
BA	6 layers	0.8190	0.0000	0.6405	0.0000
	12 layers	0.8933	0.0000	0.7054	0.0000
	24 layers	0.9428	0.0000	-0.0490	0.2343

Table 4: Spearman correlation coefficients (ρ) and their corresponding p-values between side effects and two subject-related metrics: the degree of the target knowledge's subject and the editing intensity of the target knowledge.

between a subject's feature vectors before and after editing:

$$\text{EI}(s) = \|h_{\text{pre}}(\text{“}s\text{”}, 0, 0) - h_{\text{post}}(\text{“}s\text{”}, 0, 0)\|_2 \tag{4}$$

where $h(p, t, l)$ represents hidden-state immediately after layer l at position t when prompted with prompt p. We set $l = 0$, corresponding to the layer where ROME applies the edit. t is set to 0 as we only feed one subject token to the model in this analysis. The subscripts “pre” and “post” refer to the hidden state before and after ROME is applied. We then analyze the correlation coefficient between this EI metric and the impact of editing I defined by Equation 3.

Results Table 4 shows the Spearman correlation coefficients and p-values between the Edit Intensity of subject entities and the impact of side effects I. Except for a 24-layer model trained on BA graphs, which is an exception, we generally observe positive correlations. Notably, strong positive correlations are found in the 6-layer and 12-layer models trained on BA graphs. These results suggest that knowledge editing operations requiring larger changes to subject entity representations tend to produce more substantial side effects. This finding indicates that the robustness of subject entity embeddings within the model significantly influences the extent of side effects during knowledge editing.

5.3 Relation Similarity

Procedure We investigate how the similarity between relation vectors of knowledge instances affects the occurrence of side effects, focusing on instances that share the same subject as the edited knowledge. Here, relation vectors are defined as $h(\text{“}r\text{”}, 0, 0)$.

Results The results in Table 5 reveal significant differences in relation similarity between knowledge instances that are affected by editing and those that are not. Specifically, knowledge instances affected by editing showed higher average relation vector similarities compared to unaffected instances.

5.4 Object Density

Procedure We examine how the density of object vectors in the embedding space affects the occurrence of side effects, focusing on subject-sharing knowledge. We define object vectors as $h(\text{“}s\ r\text{”}, 1, -1)$, where $l = -1$ denote the final layer. The density is computed using the k-Nearest Neighbor algorithm (Fix, 1985), where we calculate the density of each instance with respect to the complete set of object vectors from all knowledge instances[4].

Results As shown in Table 5, there are differences in object density distributions between knowledge instances that experience side effects and those that do not. Knowledge instances affected by editing exhibit consistently higher average object vector densities compared to unaffected instances. This pattern persists across various model architectures and graph types, suggesting that knowledge instances whose objects are located in dense regions of the embedding space are more sensitive to editing operations and are more prone to side effects.

5.5 Knowledge Distortion

Procedure We examine how distortions in knowledge instance embeddings influence the occurrence of side effects. While measuring the smoothness of knowledge instance embeddings is non-trivial, inspired by (Jukić and Šnajder, 2024), we analyze knowledge distortions through the lens of Lipschitz continuity. The Lipschitz constant (LC) is defined as:

$$\text{LC}(s, r, o) = \frac{\|h_{\text{pre}}(\text{“}s\ r\text{”}, 1, -1) - h_{\text{post}}(\text{“}s\ r\text{”}, 1, -1)\|_2}{\|h_{\text{pre}}(\text{“}s\text{”}, 0, 0) - h_{\text{post}}(\text{“}s\text{”}, 0, 0)\|_2} \tag{5}$$

This metric quantifies the relative magnitude of changes in the object embeddings with respect to changes in the subject embeddings. A higher LC indicates greater distortion in the knowledge representation.

[4] See Appendix C for detailed calculations.

Graph	Model	Relation Similarity		Object Density		Knowledge Distortion	
		Affected	Unaffected	Affected	Unaffected	Affected	Unaffected
ER	6-layer	**0.5931**	0.4865	0.3668	**0.3818**	**1.6360**	1.4216
	12-layer	**0.4923**	0.4371	**0.3463**	0.3374	**2.4217**	1.8883
	24-layer	**0.5293**	0.4227	**0.4137**	0.4043	**4.2525**	3.4570
BA	6-layer	**0.4476**	0.3728	**0.4656**	0.4563	**1.3327**	1.1151
	12-layer	**0.3822**	0.3285	**0.6236**	0.6220	**1.8367**	1.4594
	24-layer	**0.5423**	0.5115	**0.4768**	0.4020	**3.6963**	3.1530

Table 5: Analysis of subject-sharing knowledge based on three metrics (Relation Similarity, Object Density, and Knowledge Distortion), comparing knowledge affected by side effects versus unaffected knowledge. Values represent the mean scores for each metric across respective knowledge groups.

Results As demonstrated in Table 5, knowledge instances affected by side effects consistently show larger Lipschitz constants than unaffected instances. This finding suggests that knowledge instances lacking smooth representations in the embedding space are more susceptible to editing operations. The results indicate a strong relationship between the local geometric properties of learned knowledge representations and their stability under editing. Specifically, knowledge encoded with smooth embeddings demonstrates greater robustness to editing operations, while knowledge characterized by abrupt changes in the embedding space is more vulnerable to editing effects and has a higher probability of producing unintended side effects.

6 Discussion

Findings from Section 5 provide important implications for developing more effective knowledge editing methods. First, the correlation between Edit Intensity and side effects suggests that constraining the magnitude of changes in subject entity representations before and after editing could enable more localized knowledge editing. Second, understanding the propagation mechanism of side effects among subject-sharing knowledge instances is expected to contribute to designing more sophisticated editing techniques.

Furthermore, the characteristics of side effects we show in this study have significant implications for evaluating knowledge editing methods. While conventional evaluation metrics have primarily measured the effectiveness of editing on target knowledge instances, our findings suggest the need for more comprehensive evaluation approaches, including (1) weighted evaluation based on subject entity degree, (2) quantification of side effects based on relation vector similarity, and (3) stability assessment considering local density of knowledge representations.

Continued verification in environments that more closely approximate real-world knowledge structures is also important. With the above considerations, we expect the establishment of more robust knowledge editing methods and evaluation techniques.

7 Conclusion

In this study, we conducted a comprehensive analysis of the side effects caused by knowledge editing of LMs. In the analysis of what kinds of knowledge instances cause large side effects when edited, we showed that instances with higher-degree subject entities and greater edit intensity lead to larger side effects. Further analyses on which kinds of knowledge instances get affected by editing other knowledge revealed that, particularly among the subject-sharing knowledge with edited knowledge, (1) knowledge instances with high relation vector similarity to the edited instance, (2) knowledge instances with objects existing in dense regions, and (3) knowledge instances with distorted representations are susceptible to side effects. These findings provide valuable insights into the mechanisms behind side effects in knowledge editing and highlight important considerations for developing more precise and reliable editing methods.

Future works include validating these findings using larger-scale models, exploring generalization across different editing algorithms, and conducting comprehensive evaluations with real-world datasets. Additionally, we aim to leverage these insights to develop more robust knowledge editing techniques and establish complete evaluation frameworks to ensure the reliability of edited models in practical applications.

Limitations

One limitation of this study is the constraint on the choice of model architecture. Specifically, while we adopted the GPT-2 architecture and conducted verification with multiple layer settings, this is just one of the basic architectures in natural language processing tasks. By conducting verification with models of different architectures, it may be possible to more broadly evaluate the generalizability of the insights gained in this study.

Second, there is a limitation regarding the scope of verification of knowledge editing methods. In this study, we focused only on ROME, an editing method based on rank-1 matrices, and analyzed its side effects in detail. However, the analysis framework established in this study is also applicable to other different editing methods, and it is expected that more comprehensive insights into the side effects of knowledge editing can be obtained through comparative analysis between methods.

Third, the simplification of the knowledge representation used in the analysis is another limitation. In this study, as an initial stage of the analysis, we adopted an artificial knowledge graph that represents each element of subject, relation, and object as a single token and treats it as a static embedding. However, in actual large-scale language models, representations consisting of multiple tokens are dynamically interpreted through the attention mechanism to generate rich semantic representations depending on the context. To conduct an analysis that is more in line with the actual language processing situation, it is essential to verify the knowledge graph including multi-token representations and context-dependent embeddings. Furthermore, it is considered possible to deepen the understanding of the impact of knowledge editing in natural language by developing the analysis using knowledge extracted from actual text corpora such as Wikipedia.

Ethical Considerations

This research focuses on understanding knowledge editing mechanisms in language models through the use of synthetic knowledge graphs. Our methodological choice of employing symbolic synthetic data instead of real-world datasets naturally mitigates potential ethical concerns regarding privacy, bias, and fairness that often arise in language model research. This approach enables a systematic investigation of fundamental mechanisms while avoiding risks associated with sensitive or personally identifiable information.

During code development and writing, we used AI assistants including language models. All the generated code snippets and texts are checked and modified by the authors to scientific integrity and accuracy.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR20D2, AMED Grant Number JP25wm0625405, JST BOOST Grant Number JPMJBS2412 and JPMJBS2421, JSPS KAKENHI Grant Number 25K03175, and The Nakajima Foundation.

References

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona Diab, and Marjan Ghazvininejad. 2022. A Review on Language Models as Knowledge Bases. *arXiv preprint*, arXiv:2204.06031.

Albert-László Barabási and Réka Albert. 1999. Emergence of Scaling in Random Networks. *Science*, 286:509–512.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the Ripple Effects of Knowledge Editing in Language Models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge Neurons in Pretrained Transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing Factual Knowledge in Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6491–6506.

P. Erdös and A. Rényi. 1959. On Random Graphs I. *Publicationes Mathematicae Debrecen*.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. AlphaEdit: Null-Space Constrained Model Editing for Language Models. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Zhangyin Feng, Weitao Ma, Weijiang Yu, Lei Huang, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. Trends in Integration of Knowledge and Large Language Models: A Survey and Taxonomy of Methods, Benchmarks, and Applications. *arXiv preprint*, arXiv:2311.05876.

Evelyn Fix. 1985. *Discriminatory analysis: nonparametric discrimination, consistency properties*, volume 1. USAF school of Aviation Medicine.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer Feed-Forward Layers Are Key-Value Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model Editing Harms General Abilities of Large Language Models: Regularization to the Rescue. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 16801–16819.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024. Model Editing at Scale leads to Gradual and Catastrophic Forgetting. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15202–15232.

Benjamin Heinzerling and Kentaro Inui. 2021. Language Models as Knowledge Bases: On Entity Representations, Storage Capacity, and Paraphrased Queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are Large Pre-Trained Language Models Leaking Your Personal Information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-Patcher: One Mistake Worth One Neuron. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Yoichi Ishibashi and Hidetoshi Shimodaira. 2023. Knowledge Sanitization of Large Language Models. *arXiv preprint*, arXiv:2309.11852.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. Knowledge Unlearning for Mitigating Privacy Risks in Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*.

Josip Jukić and Jan Šnajder. 2024. From Robustness to Improved Generalization and Calibration in Pre-trained Language Models. *arXiv preprint*, arXiv:2404.00758.

Jungo Kasai, Keisuke Sakaguchi, yoichi takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2023. RealTime QA: What's the Answer Right Now? In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023. PMET: Precise Model Editing in a Transformer. *arXiv preprint*, arXiv:2308.08742.

Jun-Yu Ma, Zhen-Hua Ling, Ningyu Zhang, and Jia-Chen Gu. 2024. Neighboring Perturbations of Knowledge Editing on Large Language Models. In *Forty-first International Conference on Machine Learning (ICML)*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and Editing Factual Associations in GPT. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass Editing Memory in a Transformer. *The Eleventh International Conference on Learning Representations (ICLR)*.

Benjamin Millidge and Sam Black. 2022. The Singular Value Decompositions of Transformer Weight Matrices are Highly Interpretable. *AI Alignment Forum*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. Fast Model Editing at Scale. In *International Conference on Learning Representations (ICLR)*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *International Conference on Machine Learning (ICML)*, pages 15817–15831.

Yasumasa Onoe, Michael Zhang, Shankar Padmanabhan, Greg Durrett, and Eunsol Choi. 2023. Can LMs Learn New Entities from Descriptions? Challenges in Propagating Injected Knowledge. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5469–5485.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? 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)*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, heaper and lighter. In *NeurIPS EMC^2 Workshop*.

Daniel Trippa, Cesare Campagnano, Maria Sofia Bucarelli, Gabriele Tolomei, and Fabrizio Silvestri. 2024. $\nabla\tau$: Gradient-based and task-agnostic machine unlearning. *arXiv preprint*, arXiv:2403.14339.

Yu Wang, Ruihan Wu, Zexue He, Xiusi Chen, and Julian McAuley. 2025. Large Scale Knowledge Washing. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin, and Xueqi Cheng. 2024. The Butterfly Effect of Model Editing: Few Edits Can Trigger Large Language Models Collapse. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5419–5437.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Shu Wu, Pengjie Ren, and Zhumin Chen. 2025. Uncovering Overfitting in Large Language Model Editing, author=Mengqi Zhang and Xiaotian Ye and Qiang Liu and Shu Wu and Pengjie Ren and Zhumin Chen. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. 2024. A Comprehensive Study of Knowledge Editing for Large Language Models. *arXiv preprint*, arXiv:2401.01286.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X. Yu, and Sanjiv Kumar. 2020. Modifying Memories in Transformer Models. *CoRR*, abs/2012.00363.

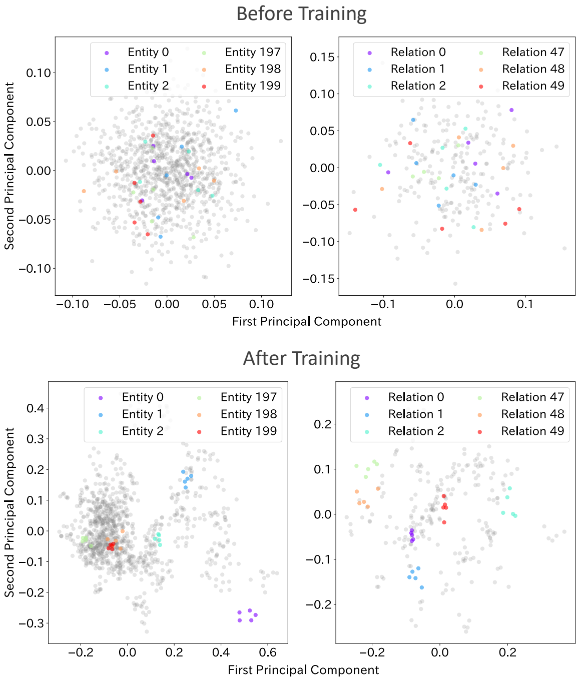

Figure 4: The visualization compares the word embedding spaces before training (top) and after training (bottom). In each row, the left plot displays the PCA results for entity embeddings, while the right plot shows the results for relation embeddings. Within these plots, points of the same color represent five different paraphrases of the same concept, allowing us to observe how semantically similar expressions are clustered in the embedding space.

A Recognition of Paraphrased Representation in LMs

As described in Section 3.2, the synthetic knowledge graphs created in this study were designed with a structure where each entity and relation has multiple paraphrases, more closely replicating real-world scenarios where concepts can be expressed in various ways. For training the model on these knowledge graphs, we utilized only a sampled subset of all possible knowledge instances as training data. This approach is based on the consideration that if the LM learned all relational knowledge, it might result in the LM merely memorizing each instance of relational knowledge, thereby hindering its generalization ability to recognize paraphrased expressions.

Figure 4 presents a visualization of the model's word embedding spaces before and after training on an Erdős-Rényi (ER) graph, projected into two dimensions using Principal Component Analysis (PCA). In the pre-training state, the embeddings of paraphrases for both entities and relationships exhibit considerable dispersion in the embedding space. After training, however, the embeddings of paraphrases for each entity and relation demonstrate clear clustering behavior, despite the model being trained on only 20% of all possible knowledge instances. These results suggest that LMs do not possess an inherent ability to recognize paraphrases, but rather acquire this capability through the learning process.

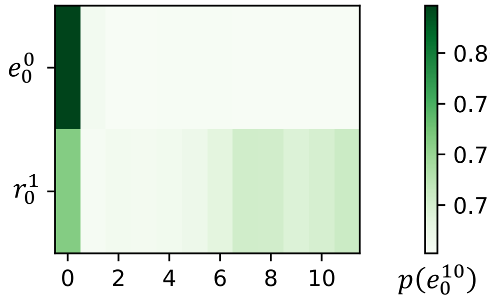

Figure 5: Example of analysis results for the contribution of FF layers in each layer by Causal Tracing (in the case of a 12-layer model). The horizontal axis represents the layers of the LM, and the vertical axis corresponds to each input token. The values indicate the difference in the probability of generating the correct token before and after the corrupted-with-restoration run. The darker areas indicate that restoring the corresponding FF layers allows the model to generate the correct token again, demonstrating that those FF layers contribute to knowledge prediction.

B Supplemental Information on the Knowledge Editing Method

In Section 3.3, we briefly introduced ROME, an existing knowledge editing method used in our experiments. This section provides a more detailed explanation of Causal Tracing, a step crucial for locating the parts that play a significant role when the LM associates knowledge. Causal Tracing analyzes the contribution of each hidden state of the LM during inference through the following procedure[5].

1. *clean run*: First, predictions related to knowledge are made to obtain the normally hidden states of the model. Specifically, all hidden states $\{h_i^{(l)} \mid i \in [1, T], l \in [1, L]\}$ are determined when the model predicts o from input $x = (s, r)$. Here, T is the length of the input x (in this work, $T = 2$), and L is the number

[5] Manipulations to the hidden states of LMs can also be conceptualized as to the FF layers or attention layers.

of layers in the model.

2. *corrupted run*: Next, when making predictions related to knowledge, the hidden states of the corrupted model are determined by hiding information about the subject. Specifically, when input x is provided, noise is added to the embedding representation $h_1^{(0)}$ corresponding to the subject ($h_1^{(0)} := h_1^{(0)} + \epsilon$). Afterward, predictions related to knowledge are made, and the corrupted hidden states $\{h_{i*}^{(l)} \mid i \in [1, T], l \in [1, L]\}$ are determined. As a result, the correct output that could be output during the clean run can no longer be output during the corrupted run.
3. *corrupted-with-restoration run*: Finally, for the model with the corrupted hidden states obtained from the corrupted run, specific hidden states $h_{i*}^{(l)}$ are restored to the normally hidden states $h_i^{(l)}$ obtained during the clean run. This process is performed for each hidden state individually, and predictions related to knowledge are made. When the correct output can be output again by restoring a specific hidden state, it indicates that the hidden state contributes to knowledge prediction.

Figure 5 presents an example result of analyzing the contributions of the FF layers at each layer using Causal Tracing. In this case, the FF layer in the first layer plays a significant role when the LM outputs the correct token e_0^{10} from the input token sequence (e_0^0, r_0^1). Thus, Causal Tracing facilitates the identification of parts that play crucial roles in the LM's knowledge prediction.

C Object Density Calculation

The density of object vectors is estimated using the k-Nearest Neighbors (k-NN) algorithm. For each object vector v_{o_i}, we compute its local density δ_i as:

$$\delta_i = \frac{1}{\frac{1}{k}\sum_{j=1}^{k} d_{ij} + \epsilon} \tag{6}$$

where:

- k is the number of nearest neighbors (set to 10 in our implementation)
- d_{ij} is the Euclidean distance to the j-th nearest neighbor
- ϵ is a small constant (10^{-6}) to prevent division by zero

The density scores are then normalized to the range [0,1] using min-max normalization:

$$\delta_i^{\text{norm}} = \frac{\delta_i - \min_j \delta_j}{\max_j \delta_j - \min_j \delta_j} \tag{7}$$

This normalized density score provides a relative measure of how clustered the object vectors are in the embedding space. Higher values indicate regions where object vectors are more densely packed, while lower values correspond to more sparse regions.

D Additional Experimental Results

In Section 5.1, we demonstrated the relationship between subject degree and side effects for the 6-layer model. In Figures 6 and 7, we present the results for both 12-layer and 24-layer models.

Furthermore, in Sections 5.3 to 5.5, we compared the mean values of each metric between the Affected knowledge group and the Unaffected knowledge group. For more detailed results, histograms showing the frequency distributions of each metric are presented in Figures 8 to 10. Additionally, Tables 6 to 8 show the statistical values and results of the Kolmogorov-Smirnov test.

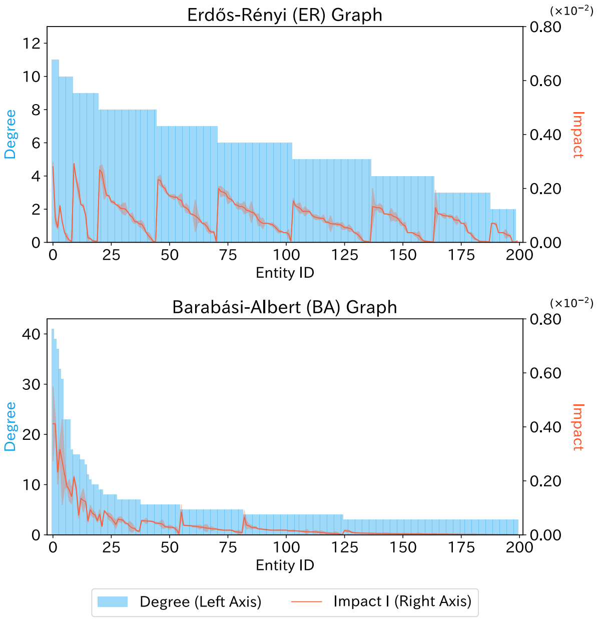

Figure 6: Analysis of the relationship between subject degrees and editing impacts in a 12-layer model trained on the knowledge graph.

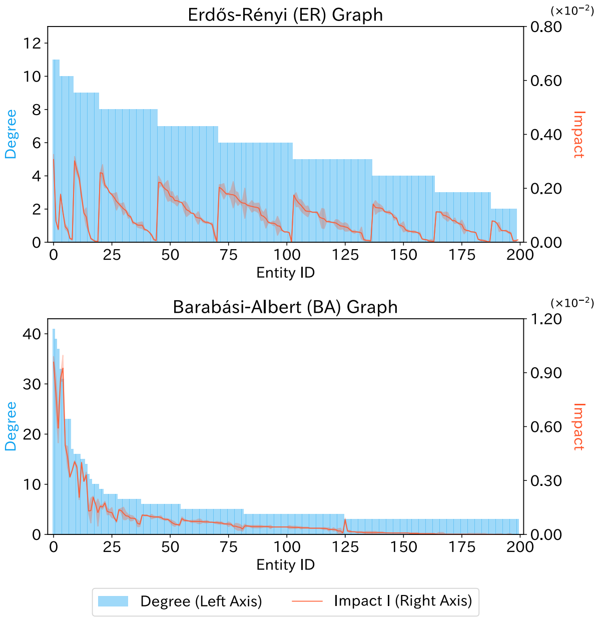

Figure 7: Analysis of the relationship between subject degrees and editing impacts in a 24-layer model trained on the knowledge graph.

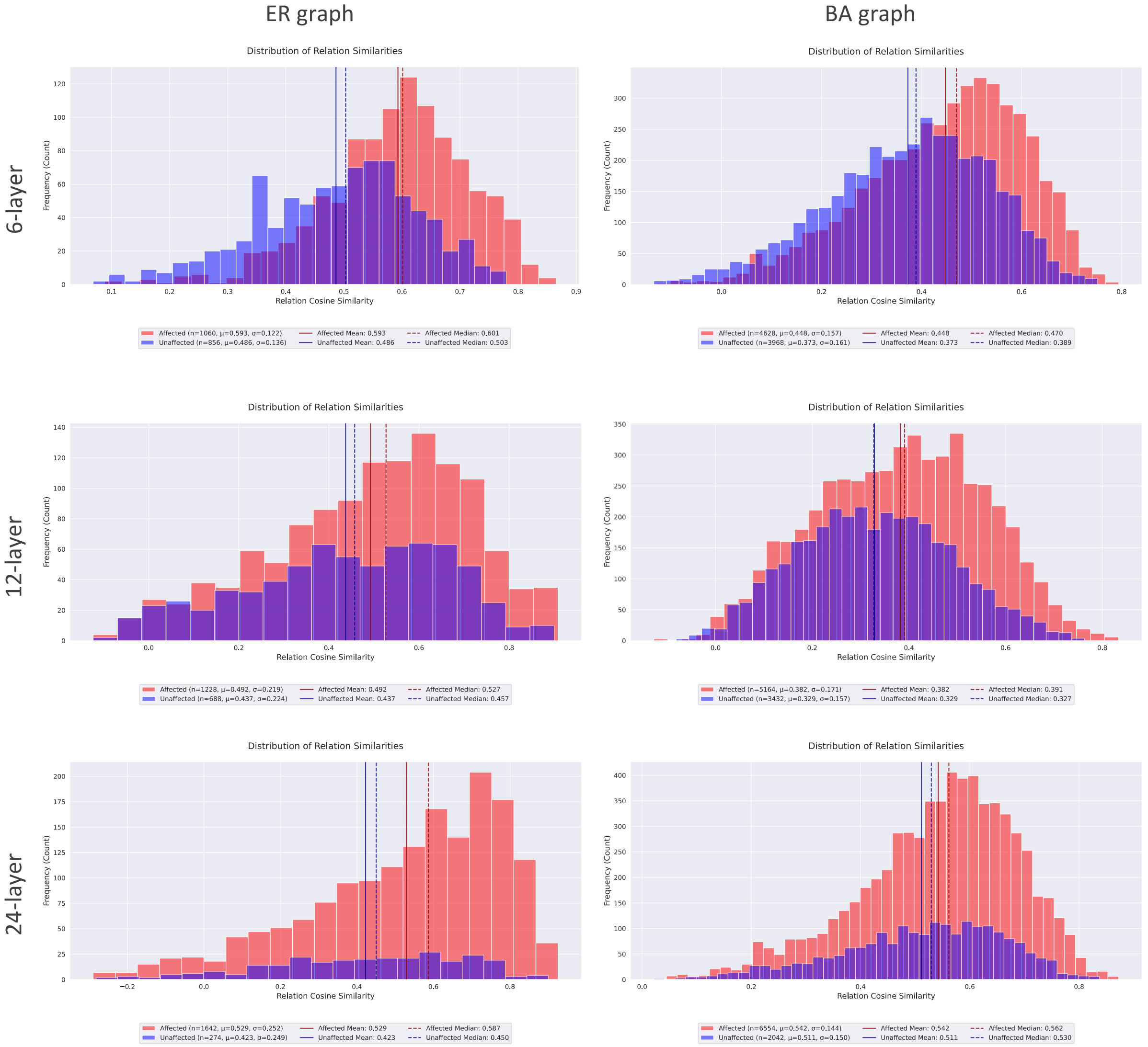

Figure 8: Distribution of relation vector similarities between edited knowledge and subject-sharing instances, comparing affected (red) and unaffected (blue) cases.

Graph	Model	Affected			Unaffected			KS Test	
		mean	median	std	mean	median	std	statistic	p-value
ER	6-layer	0.5931	0.6012	0.1219	0.4865	0.5031	0.1363	0.3356	9.12E-48
	12-layer	0.4923	0.5269	0.5269	0.4371	0.4572	0.2237	0.2237	5.24E-06
	24-layer	0.5293	0.5869	0.2517	0.4227	0.4504	0.2486	0.2220	1.25E-10
BA	6-layer	0.4476	0.4697	0.1571	0.3728	0.3890	0.1607	0.2043	1.67E-78
	12-layer	0.3822	0.3909	0.1710	0.3285	0.3273	0.1571	0.1496	1.10E-40
	24-layer	0.5423	0.5617	0.1436	0.5115	0.5295	0.1495	0.0941	1.90E-12

Table 6: Comparison of relation similarity distributions between affected and unaffected knowledge in subject-shared neighborhood. For knowledge sharing the same subject as the editing target, we analyze the relation similarity with the target. The Kolmogorov-Smirnov test results show significant differences between affected and unaffected groups across different model architectures and graph types, suggesting that the editing impact on knowledge with similar relations is distinguishable even within the same subject group.

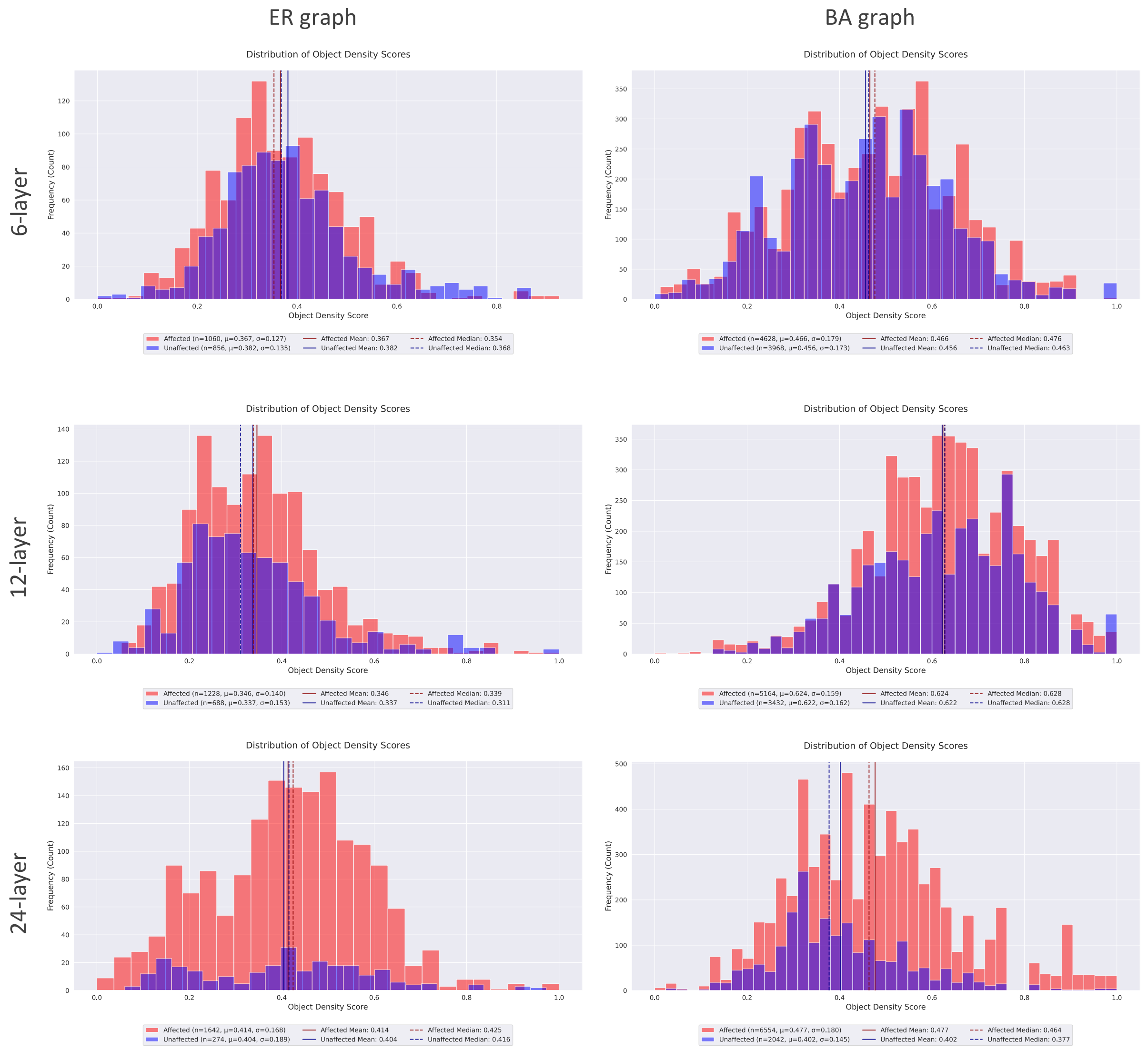

Figure 9: Distribution of object vector densities in subject-sharing instances, comparing affected (red) and unaffected (blue) cases.

Graph	Model	Affected			Unaffected			KS Test	
		mean	median	std	mean	median	std	statistic	p-value
ER	6-layer	0.3668	0.3538	0.1274	0.3818	0.3684	0.1353	0.0619	5.01E-02
	12-layer	0.3463	0.3385	0.1398	0.3374	0.3109	0.1531	0.0890	1.71E-03
	24-layer	0.4137	0.4246	0.1681	0.4043	0.4156	0.1888	0.1063	9.14E-03
BA	6-layer	0.4656	0.4764	0.1788	0.4563	0.4630	0.1732	0.0524	1.50E-05
	12-layer	0.6236	0.6277	0.1593	0.6220	0.6277	0.1621	0.0423	1.21E-03
	24-layer	0.4768	0.4638	0.1798	0.4020	0.3774	0.1453	0.2206	6.62E-67

Table 7: Comparison of object density (kNN) distributions between affected and unaffected knowledge in subject-shared neighborhood. For knowledge sharing the same subject as the editing target, we analyze the object density with the target. The Kolmogorov-Smirnov test results show significant differences between affected and unaffected groups across different model architectures and graph types, suggesting that the editing impact on knowledge with similar object density is distinguishable even within the same subject group.

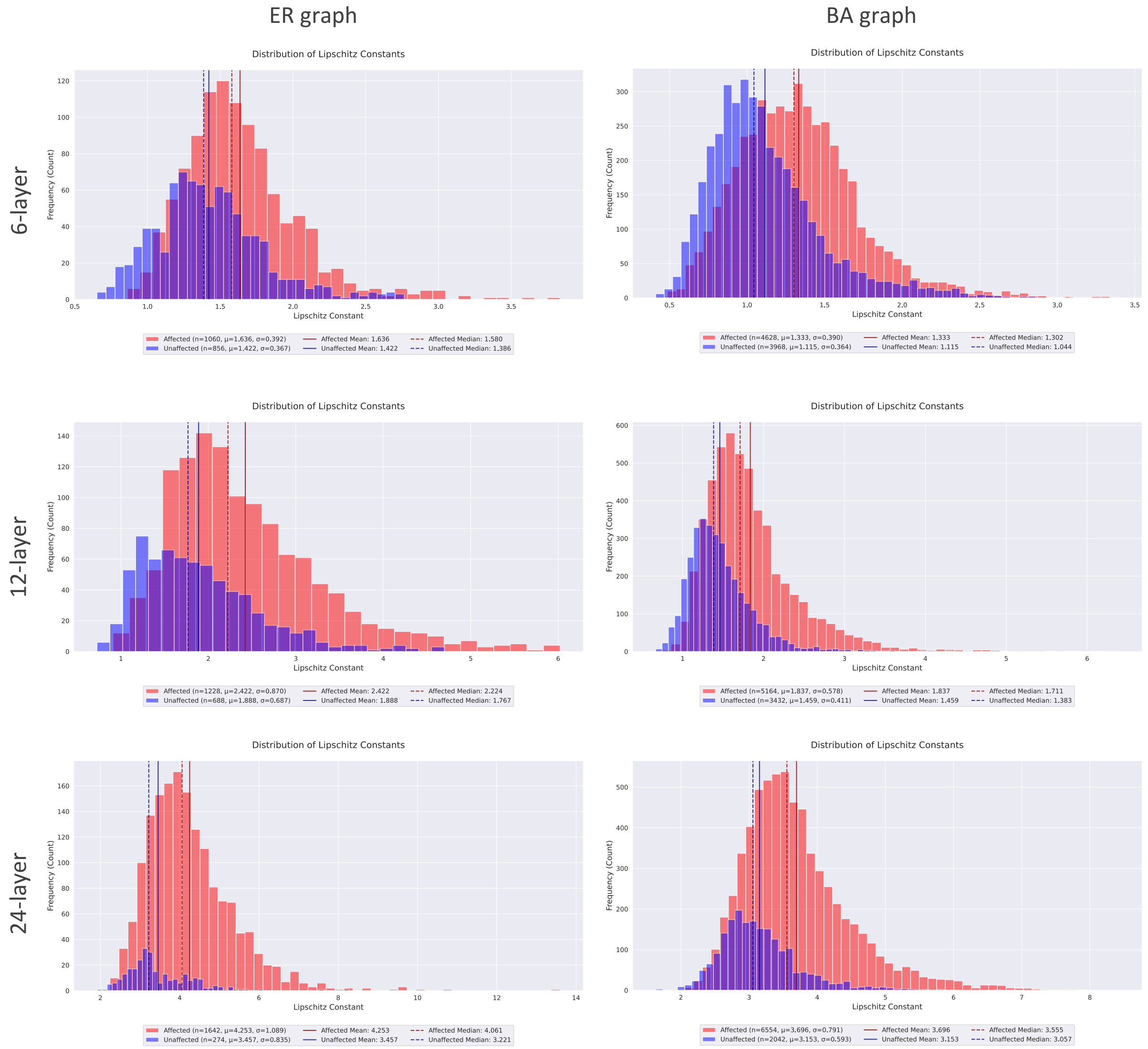

Figure 10: Distribution of Lipschitz constant in subject-sharing instances, comparing affected (red) and unaffected (blue) cases

Graph	Model	Affected			Unaffected			KS Test	
		mean	median	std	mean	median	std	statistic	p-value
ER	6-layer	1.6360	1.5796	0.3918	1.4216	1.3858	0.3667	0.2528	4.31E-27
	12-layer	2.4217	2.2237	0.8699	1.8883	1.7668	0.6874	0.2872	1.44E-32
	24-layer	4.2525	4.0613	1.0888	3.4570	3.2211	0.8346	0.3981	5.72E-34
BA	6-layer	1.3327	1.3017	0.3904	1.1151	1.0440	0.3644	0.2851	1.76E-153
	12-layer	1.8367	1.7107	0.5775	1.4594	1.3829	0.4106	0.3422	1.26E-214
	24-layer	3.6963	3.5548	0.7914	3.1530	3.0567	0.5934	0.3384	9.72E-159

Table 8: Comparison of Lipschitz constants between affected and unaffected knowledge in subject-shared neighborhood. For knowledge sharing the same subject as the editing target, we analyze the Lipschitz constants with respect to the target. The Kolmogorov-Smirnov test results show significant differences between affected and unaffected groups across different model architectures and graph types, suggesting that the editing impact on knowledge with similar Lipschitz constants is distinguishable even within the same subject group.