RelEdit: Evaluating Conceptual Knowledge Editing in Language Models via Relational Reasoning

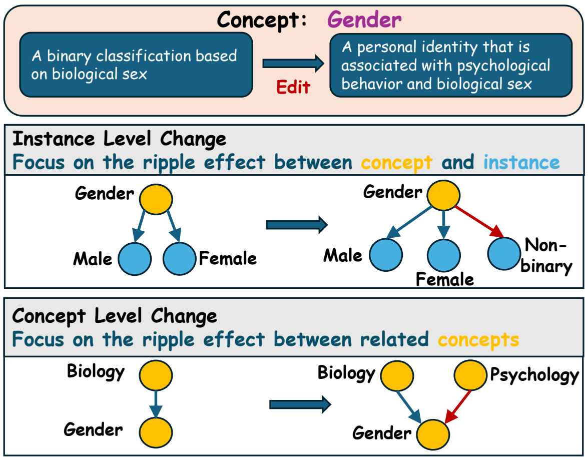
Yifan Niu[1]*, Miao Peng[1]*, Nuo Chen[1], Yatao Bian[2], Tingyang Xu[3,4], Jia Li[1,5]†
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]Tencent AI Lab [3]DAMO Academy, Alibaba Group [4]Hupan Lab
[5]The Hong Kong University of Science and Technology
{yniu669, mpeng885}@connect.hkust-gz.edu.cn jialee@ust.hk

Abstract

The conceptual knowledge in Large Language Models (LLMs) can become outdated over time, and concept editing is often an option. Current evaluations on conceptual knowledge editing primarily focus on whether the definitions of concepts are successfully edited, neglecting the impact on the model's related beliefs. To address this gap, we introduce a benchmark called **RelEdit**, which includes criteria and questions to assess both concept-level and instance-level relational reasoning abilities of edited models. Our findings reveal that existing knowledge editing methods struggle to reason about related conceptual knowledge effectively. Additionally, we introduce a simple memory-based in-context editing baseline, **MICE**, which prompts the language model to generate answers that align with the stored edited concepts in external memory. In addition, we find that MICE obtains the best scores on our benchmark, suggesting a promising research direction for model editing. The code is available at https://github.com/ivanniu/RelEdit.

1 Introduction

As large language models (LLMs) are widely deployed, it becomes increasingly important to maintain their knowledge accuracy and currency without incurring significant retraining costs (Sinitsin et al., 2020). Previous studies have introduced knowledge editing methods to gradually incorporate new concrete factual knowledge into language models (Zhu et al., 2020; De Cao et al., 2021; Meng et al., 2022a,b; Mitchell et al., 2021, 2022; Tan et al., 2023). However, these approaches of editing case-by-case factual knowledge are highly inefficient and lacks the modeling of relationships between instances (Wang et al., 2024). Inspired by cognitive science (Zhao et al., 2024; Holzinger et al., 2023), researchers have explored on *conceptual knowledge editing* (Lo et al., 2024; Lv et al., 2024; Suresh et al., 2023; Jamali et al., 2023), which involves modifying higher-level abstract concept rather than concrete facts in LLMs. For example, the concept "Gender" is originally a binary biological concept, including instances "male" and "female". In recent years, with the development of society, the definition of "Gender" has introduced psychological factors and derived "Non-Binary Gender".



Concept: Gender
A binary classification based on biological sex
Edit
A personal identity that is associated with psychological behavior and biological sex
Instance Level Change
Focus on the ripple effect between concept and instance
Gender
Male
Female
Gender
Male
Female
Non-binary
Concept Level Change
Focus on the ripple effect between related concepts
Biology
Gender
Biology
Psychology
Gender

Figure 1: Illustration of the relational reasoning on instance-level and concept-level when editing the concept definition of "Gender".

Although conceptual knowledge editing have raised great research interest, there is no systematic study of evaluating this task. Currently, the only existing benchmark, ConceptEdit (Wang et al., 2024), focuses on measuring whether the edited model can recall the newly injected concept definitions, typically involving three common metrics: Reliability (Re), Generalization (Ge), and Locality (Lo). These metrics evaluate the answers through queries on the edited concept definitions, as well as verifying that irrelevant concepts are not corrupted. However, an important question that has not been addressed is whether the edited model can handle relational reasoning questions where the an-

[*] Equal contribution.
[†] Corresponding author

Findings of the Association for Computational Linguistics: ACL 2025, pages 10220–10238
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

Benchmark	Re	Ge	Lo	IC	PO	IL	AB	AC
ConceptEdit	✓	✓	✓	✓	✗	✗	✗	✗
RelEdit	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Metrics in concept editing benchmarks.

swer should change as a logical consequence of the edited concepts.

In this work, we introduce a benchmark **RelEdit** (Relational Reasoning for Conceptual Knowledge Editing). When evaluating concept editing, it is important to go beyond just examining the edited concept definition. We should also assess whether both the **concepts** and **instances** logically derived from the edit have been appropriately modified, as illustrated in Figure 1. Following existing benchmarks, we construct relational examples for evaluation. (i) At the instance level, we evaluate the relationship changes between concepts and concrete instances. (ii) At the concept level, we evaluate the relationship changes among concepts. From the two perspectives, we propose novel evaluation criteria to evaluate how well the model integrates the edit with the rest of its knowledge: Instance Change (IC), Portability (PO), Instance Locality (IL), Alignment Belong (AB), and Alignment Compare (AC), as listed in Table 1.

We observe that although current methods effectively edit concepts, they often struggle to infer the relational knowledge associated with those concepts. Additionally, our analysis reveals that: (a) larger models are better equipped to handle the relational reasoning challenge; (b) modifications between concepts that share similar semantic structures and superclasses facilitate the model's ability to update more related conceptual knowledge; and (c) instance-level evaluations reflect the ability of LLMs to maintain higher concept consistency by updating knowledge from related instances. Finally, we propose a memory-based in-context editing baseline, MICE, which prompts the LLMs to generate answers that align with the stored edited concepts in external memory instead of explicit parametric updates. While MICE achieves superior results compared to current parametric methods on RelEdit, there is still significant room for improvement, necessitating further research.

2 Problem Formulation

In contrast to concrete factual knowledge, conceptual knowledge refers to the understanding of concepts, categories, principles, and relationships. It involves understanding and applying abstract ideas. It is evidenced that language models are capable of memorizing conceptual knowledge and making inferences (Wu et al., 2023a). The conceptual knowledge stored in language models can be incorrect or become outdated over time. One potential solution is to update the knowledge without retraining.

A concept can be represented as $C = (c, d)$ (Wang et al., 2024), where c denotes the concept name (e.g., school) and d means the definition description of concept (e.g., an institution for the education of students by teachers). For a concept $C = (c, d)$, there are several instances e belonging to the category broadly defined by the concept, which can be denoted as $e \in C$ (e.g., *Microsoft Forecaster* is an instance of concept *software*).

A conceptual knowledge edit $\delta : (c, d) \rightarrow (c, d^*)$ is defined as modifying the concept $C = (c, d)$ into a refreshed one $C^* = (c, d^*)$, that is, updating the concept definition $d \rightarrow d^*$ for a given concept name c, in which d^* represents the alternate concept definition. Given a language model, conceptual knowledge editing aims to inject the edited concept to the language models' inner beliefs. Since high-order concepts preserve meta knowledge in a hierarchical form, it is quite essential to investigate the relational reasoning of a single concept editing to certain related concepts and instances.

3 RelEdit Benchmark Construction

In this section, we introduce the data construction of RelEdit. It is difficult to have a universal principle for all concepts in the world. To ensure the rationality of the concepts, our concepts follow the well-established DBpedia (Auer et al., 2007), also with human judgement. Some details about data construction are discussed in the Appendix A.

3.1 Ontology Building

To obtain high-order meta ontological knowledge, we adopt DBpedia (Auer et al., 2007), a widely used knowledge graph with tree-like structured ontology, constructing a hierarchical concept set with classes and corresponding individual instances. Specifically, we extract 783 distinct classes in total from DBpedia as performed by Wu et al., in which each class represents the typical ontological information. Note that several classes in DBpedia are free of instance facts, which lead to a marginal contribution to concept editing task. To this end,

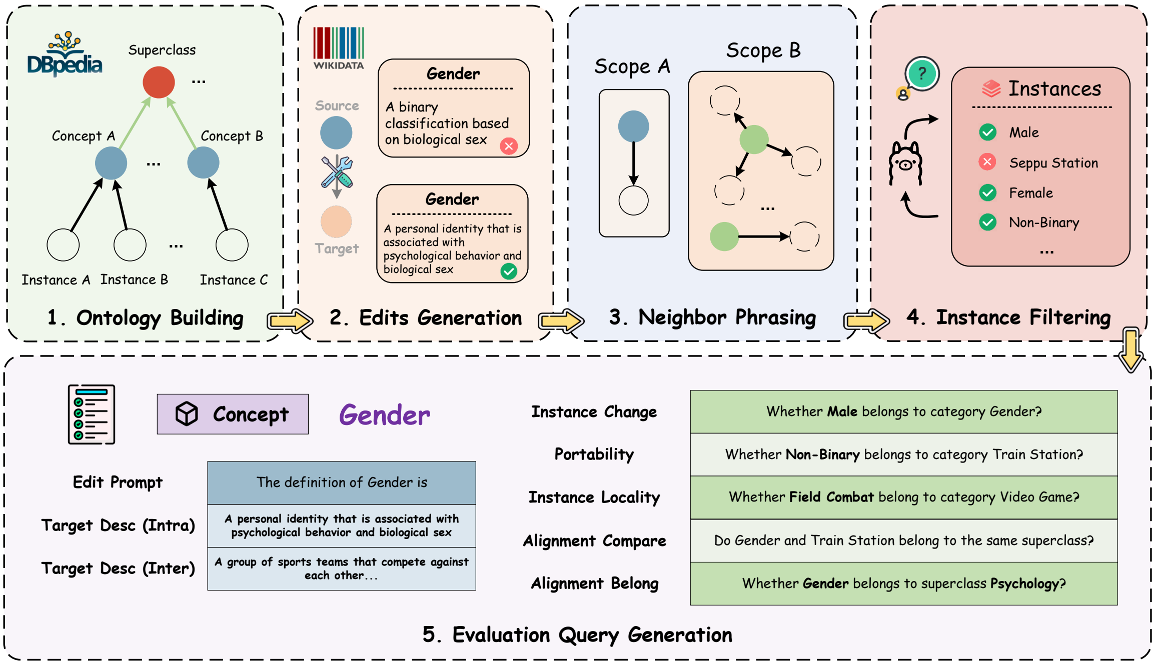

DBpedia
Superclass
...
Concept A
Concept B
...
...
Instance A
Instance B
Instance C
1. Ontology Building
WIKIDATA
Source
Target
Gender
A binary classification based on biological sex
Gender
A personal identity that is associated with psychological behavior and biological sex
2. Edits Generation
Scope A
Scope B
...
3. Neighbor Phrasing
?
Instances
Male
Seppu Station
Female
Non-Binary
...
4. Instance Filtering
Concept
Gender
Edit Prompt
Target Desc (Intra)
Target Desc (Inter)
The definition of Gender is
A personal identity that is associated with psychological behavior and biological sex
A group of sports teams that compete against each other...
Instance Change
Portability
Instance Locality
Alignment Compare
Alignment Belong
Whether Male belongs to category Gender?
Whether Non-Binary belongs to category Train Station?
Whether Field Combat belong to category Video Game?
Do Gender and Train Station belong to the same superclass?
Whether Gender belongs to superclass Psychology?
5. Evaluation Query Generation

Figure 2: Overall illustration of data generation pipeline: (1) We start by building up the ontology set, (2) then we generate the edit requests in intra and inter settings. (3) Then we rephrase the request prompt to neighbor edit request. (4) Next, we filter out those instances are not inherently stored in an LLM. (5) Last, we generate RelEdit evaluation queries for each concept edit.

we filter out those classes with the amount of instances less than k. Besides, we employ SPARQL to query the corresponding instances of each class in DBpedia via the *Type* relation, and superclass of each class via the *subclass of* relation (We keep the highest-level class in the hierarchical superclass tree). These retrieved concepts, instances and superclasses formulate a hierarchical ontology set that can be utilized in the following data construction. To ensure the dataset is balanced, we randomly choose 20 instances for each class. Considering the lack of descriptive texts of ontological classes in DBpedia, we turn to retrieve corresponding definition descriptions of classes in Wikidata (Vrandečić and Krötzsch, 2014).

3.2 Data Generation Pipeline

Edits generation. For a edit request $\delta : d \rightarrow d^*$ about concept $C = (c, d)$, target concept description d^* is chosen from a different concept $C^* = (c, d^*)$. Considering the different ontological scope, we categorize the edit request δ into two different settings (*intra* and *inter*) according to the homogeneity between origin concept C and target concept C^*. Specifically, under the *intra* setting, origin concept C and target concept C^* are subclasses of the same superclass, sharing close relationship to high-order semantics. In contrast, *inter* setting indicates the disparate category of C and C^* in high-level superclass. Thus, concept edit request can be formulated as “The definition of [concept name c] is [concept description d^*]", in which d^* is the definition replacement from either an *intra* or *inter* setting. Since our concepts are collected from real-world DBpedia, the rationale behind the two settings is to simulate real-world scenarios where concepts may evolve, ranging from minor (intra) to significant (inter) changes.

Neighbor phrasing. With obtained edit requests, the post-edited models are able to answer corrupted target definitions towards the query like “What is the definition of [concept name]?". To this end, we rephrase the request prompt and construct neighbor edit requests with a different format but similar semantics with concept name c and target definition d^*. The underlying reason is that these equivalent neighbors are supposed to be edited and maintain similar conceptual knowledge in the post-edited models. To keep close semantics between neighbor prompts, we apply GPT-4 to generate 20 equivalent neighbor query prompts for each concept edit request. We uphold text accuracy by conducting thorough checks on all generated query prompts, correcting any that are confusing or not precise.

Instance filtering. Considering the inconsistency between inherent prior knowledge of LLMs and instances from external knowledge graphs, the next step is instance filtering. Concretely, to evaluate the relational reasoning challenges caused by concept editing, instances related to the concept are essential to be taking into consideration. However, some instances of a concept retrieved from DBpedia are not inherently stored in an LLM. To this end, we employ few-shot in-context learning approach to prompts LLM to judge the affiliation of instances to a specific concept. For those discrepancies between answers from LLM and facts from external knowledge graphs, we filter out these instances to ensure they have been “seen" by the LLM model. After completing all filtering steps for the backbone LLMs, we take the intersection of their results to form a uniform instance set.

Evaluation query generation. For each concept edit request, we generate test queries corresponding to the evaluation criteria we proposed in Section 4, aiming to evaluate the relational reasoning of concept editing. In what follows, we provide details on our implementation with built concept set.

For a subject concept $C = (c, d)$ in concept set, we denote the set of instances that belong to concept C as $I(C) = \{e|e \in C\}$. Similarly, instance set of target concept C^* can be denoted as $I_a(C^*)$ in *intra* setting and $I_e(C^*)$ in *inter* setting, separately. Noting the aforementioned instance sets are subsets after filtering. Under this circumstance, given a concept edit $\delta : d \to d^*$, we first retrieve the instance set $I(C)$ of origin concept, $I_a(C^*)$ and $I_e(C^*)$ of target concept, then we construct the evaluation queries for *Instance Change* and *Portability* in the format of “Whether [Instance] belongs to category [Concept]?". For *Instance Locality*, we adopt the same query formula with concepts and instances unrelated to C or $C*$. For *Alignment Belong*, we retrieve superclasses of C and $C*$ in DBpedia and construct corresponding queries in the format of “Whether [Concept] belongs to superclass [Superclass]?". We also utilize C and $C*$ to construct query in a comparison form like “Do [Concept C] and [Concept C^*] belong to the same superclass?". A complete example of all queries is demonstrated in Figure 2.

Quality Control. Ideally, the original description text of a concept should align with and be similar to the concept name, while the description of the edit target should remain unrelated. To ensure this, we utilize the powerful LLM Qwen2.5-72B to evaluate and retain cases where: (1) the original concept description supports the original concept name, and (2) the target concept description is irrelevant to the original concept name. Subsequently, we manually review all the collected descriptions, replacing any that are unclear or ambiguous. Details of Human Evaluation are shown in Appendix A.3.

Property	Number
# of concepts (edit requests)	452
# of instances	8,767
# of superclasses	22
Average token length per description	12.95
Max/Min number of superclass	163 / 1
# of target concept (intra)	243
# of target concept (inter)	286
Max/Min number of target concept (intra)	7 / 1
Max/Min number of target concept (inter)	6 / 1

Table 2: RelEdit Benchmark Statistics

3.3 Data Statistics

We construct RelEdit benchmark following above mentioned data construction pipeline, which contains 452 conceptual edits in both *intra* and *inter* settings. The overview pipeline is described in Figure 2. More detailed statistics are listed in Table 2, showing that our generation pipeline results in 19.40 instances per concept and 12.95 token length per description on average. Datasets in inter setting involve more target concepts.

4 Evaluation Criteria

4.1 Instance-level Evaluation

We assess the cascading consequences between the concept and instances. The principle of instance-level evaluation primarily follows: *when the definition of a concept is edited, the relationships between instances and the edited concept should also undergo corresponding changes.* For example, we prompt the LLMs to answer the question “whether [instance] belongs to category [concept]?”

Instance Change (IC). Following ConceptEdit (Wang et al., 2024), we check whether the instance $t \in C$ belonging to the original concept C now belongs to the edited concept C^*. This evaluation metric, called Instance Change, is defined as:

$$1 - G_{\theta_e}\left(C^*, t\right), \tag{1}$$

where the $G_{\theta_e}\left(C^*, t\right)$ returns value 0 when $t \notin C^*$. Conversely, it returns a value of 1 when $t \in C^*$. θ_e is the parameter after concept editing.

Portability (PO). Portability evaluates the instances t^* belonging to the target concept of description d^*. We check whether the instance t^* of the target concept now belongs to the edited concept C^*. The Portability is formulated as follows:

$$G_{\theta_e}(C^*, t^*). \tag{2}$$

Instance Locality (IL). This metric is assessed based on the frequency at which the predictions of the post-edit model remain unchanged in out-scope neighbors. It assesses the model's ability to correctly classify instances that are unrelated to the concept being edited. Given an unrelated concept C and instance t, the Instance Locality is formulated as

$$\mathrm{XNOR}\left(G_\theta(C, t), G_{\theta_e}(C, t)\right), \tag{3}$$

where $\mathrm{XNOR}(\cdot)$ returns 1 when the relationship remains unchanged and 0 when it changes.

4.2 Concept-level Evaluation

We assess the cascading consequences of concepts related to the edit from the hierarchical concept structure. The principle of concept-level evaluation primarily follows: *When the definition of a concept is updated, the attribution of related concepts should also change.*

Alignment Belong (AB). We evaluate the relationship between sub-concepts C and parent concepts C_p after editing. For example, we prompt the LLMs to answer the question "whether [concept A] belong to category [concept B]?" Given that relational reasoning regarding conceptual knowledge can potentially span a large range, we focus on a 1-hop distance from the edit. AB is defined as:

$$G_\theta(C_p^*, C^*), \tag{4}$$

where C_p^* is the parent concept of the the target concept with description d^*.

Alignment Compare (AC). In the hierarchical structure of concepts, we evaluate whether two sibling concepts belong to the same parent concept. For example, we prompt the language model to answer the question "Whether [concept A] and [concept B] share a superclass?" In this test, we adopt a more challenging inter-editing setting, indicating that after editing, the sibling concept C_s that originally belonged to the same parent concept become different. AC is defined as:

$$G_\theta(C_s^*, C^*), \tag{5}$$

where C_s^* is the superclass of the the target concept with description d^*.

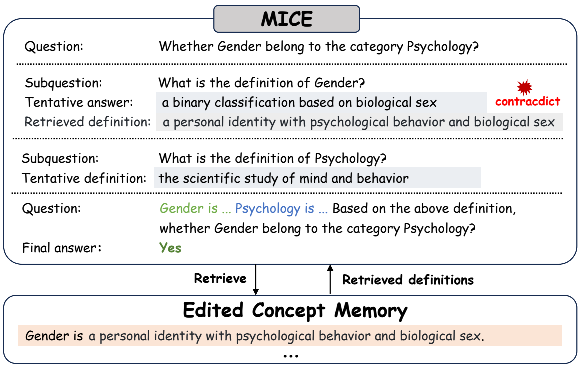

MICE
Question:
Whether Gender belong to the category Psychology?
Subquestion:
What is the definition of Gender?
Tentative answer:
a binary classification based on biological sex
contracdict
Retrieved definition:
a personal identity with psychological behavior and biological sex
Subquestion:
What is the definition of Psychology?
Tentative definition:
the scientific study of mind and behavior
Question:
Gender is ... Psychology is ... Based on the above definition, whether Gender belong to the category Psychology?
Final answer:
Yes
Retrieve
Retrieved definitions
Edited Concept Memory
Gender is a personal identity with psychological behavior and biological sex.
...

Figure 3: The illustration of MICE.

Remark Following the idea of established factual knowledge editing benchmark RIPPLEEDITS (Cohen et al., 2024), our evaluation criteria are designed to check whether the structure of ontology in the edited model has corresponding changes. All our metrics are binary. If it returns 1, it means that the corresponding ontology structure in the model has successfully changed. If it returns 0, it means that the model fails to capture the related changes. Therefore, our metrics are objective and reasonable. In this work, we do not consider polysemous concepts. Existing knowledge editing methods will overwrite the original concept meaning, thus can not tackle polysemous concepts.

5 MICE: A Memory-Based Approach

In this section, we introduce MICE, a simple but effective baseline. MICE only requires updating the question in the prompt with relevant feedback, and no retraining is need. Our proposed MICE is shown in Figure 3. The idea of MICE is inspired by MeLLo (Zhong et al., 2023), and all edited concept names and their definitions are explicitly stored in an external memory. To achieve this, we convert all edited concepts into sentence statements using predefined templates. The concept name serves as the retrieval index, allowing us to retrieve the corresponding edited concept definition. MICE performs the following steps: (1) It uses language models to extract the involved concepts in the question; (2) It extracts the involved concept definitions in the internal memory; (3) It checks whether the concept in LLM is contradicted with memory, and make corrections; and (4) It prompts the LLM to answer the questions with the corrected concept. Details about MICE are illustrated in Appendix B.

Method	**Intra Setting**							**Inter Setting**							
	Re	Ge	Lo	IC	PO	IL	AB	Re	Ge	Lo	IC	PO	IL	AB	AC
GPT2-XL															
FT	25.55	23.80	61.99	15.71	83.85	85.40	72.12	26.24	23.17	64.21	15.27	85.40	83.41	67.26	46.46
MEND	89.77	76.60	76.27	0.22	**98.23**	98.67	93.81	89.97	76.22	79.21	0.22	**98.23**	98.89	**95.58**	67.70
ROME	86.47	49.68	84.76	**23.23**	75.22	87.39	76.77	82.85	45.52	86.21	**20.35**	82.08	86.73	81.64	55.31
MEMIT	43.98	33.09	**96.11**	3.10	94.25	**99.56**	91.15	39.28	29.77	**95.93**	3.32	94.47	**99.34**	93.14	57.08
PROMPT	88.26	**86.30**	70.54	4.65	94.69	48.01	76.55	88.54	86.24	70.59	3.76	96.90	46.68	74.34	87.39
MICE (Ours)	**90.02**	85.64	81.65	18.32	96.86	65.42	**94.63**	**92.92**	**88.68**	90.62	17.63	97.22	65.64	84.63	**89.36**
GPT-J-6B															
FT	56.19	55.70	36.71	19.25	68.14	65.93	69.03	52.68	53.23	37.87	20.13	71.90	71.02	72.12	67.48
MEND	49.42	42.61	51.50	32.52	28.98	80.31	38.72	49.35	43.24	55.65	32.08	23.67	82.96	33.85	18.58
ROME	99.20	83.01	70.14	**32.74**	57.52	83.19	48.67	99.21	81.94	71.07	31.86	55.53	80.97	47.79	42.04
MEMIT	**99.83**	59.86	**94.20**	31.64	17.92	**99.56**	38.94	**99.55**	56.15	**94.80**	**33.85**	11.95	**98.89**	32.52	18.36
PROMPT	88.41	**86.42**	69.10	0.44	95.58	87.17	85.40	88.66	**87.01**	70.14	0.88	**96.46**	86.28	78.98	91.59
MICE (Ours)	90.22	84.63	78.38	17.38	**97.74**	94.15	**88.49**	91.17	86.42	82.75	14.70	96.08	94.56	**84.72**	**92.34**
LLaMA-2-7B															
FT	47.02	42.11	79.90	15.93	11.06	90.49	49.78	43.26	38.39	80.74	14.60	4.20	88.50	30.53	8.85
MEND	93.17	83.72	87.65	14.38	11.50	89.82	52.21	93.76	83.28	89.10	13.94	3.76	86.95	31.64	9.73
ROME	**99.66**	75.22	**92.41**	**40.71**	19.47	92.70	65.27	**99.63**	74.58	**92.91**	**43.81**	11.95	87.61	49.34	32.30
MEMIT	96.86	80.23	89.28	34.07	25.66	91.15	64.82	97.71	80.76	90.02	40.49	20.35	**89.16**	51.77	29.65
PROMPT	89.35	87.28	76.84	6.86	38.50	89.82	84.51	88.88	**87.89**	78.06	6.19	30.31	85.84	**79.42**	93.36
MICE (Ours)	92.32	**88.83**	83.22	18.32	**45.58**	**93.15**	**86.74**	91.37	85.78	89.75	13.42	**36.08**	87.31	75.32	**94.45**
Mistral-7B															
FT	36.12	33.51	**97.62**	0.00	50.66	**100.00**	91.37	34.31	32.30	**97.70**	0.00	35.84	**100.00**	64.16	9.51
MEND	92.94	83.61	83.47	0.22	56.64	99.56	93.14	93.52	83.10	84.88	0.44	42.92	99.78	68.36	13.72
ROME	**96.47**	76.11	93.99	10.62	78.32	99.56	**95.58**	**96.56**	76.00	94.37	11.50	70.58	**100.00**	**89.82**	65.27
MEMIT	95.35	78.95	91.98	**16.59**	65.04	99.56	87.17	95.39	77.18	91.02	**16.15**	61.50	**100.00**	74.78	36.28
PROMPT	90.22	**88.65**	81.31	0.44	94.25	94.69	88.50	90.17	88.68	82.75	0.22	92.48	94.47	77.65	95.13
MICE (Ours)	93.14	86.13	86.72	6.23	**95.52**	**100.0**	90.68	92.46	**90.35**	89.74	5.57	**94.93**	98.32	76.02	**95.64**

Table 3: Main results of baselines on RelEdit in both intra and inter setting with backbone model GPT2-XL, GPT-J-6B, LLaMA-2-7B and Mistral-7B. The best results are in **bold** and the second best results are underlined. **Re**, **Ge** and **Lo** are the abbreviation of metric Reliability, Generalization and Locality.

6 Experiments

In this section, we evaluate on RelEdit. Experimental details and results including the Impact of LLM Size, Conceptual Patterns across Methods and Editing Cases are in Appendix C and D.

6.1 Experimental Setup

Language models Four most prevalent open-source LLMs are used as base models for editing tasks. We use GPT-J (6B) (Wang and Komatsuzaki, 2021) and GPT2-XL (1.5B) (Radford et al., 2019a), LLaMA-2-7B (Touvron et al., 2023a) and Mistral-7B-v0.1 (Jiang et al., 2023). It is important to mention that current parameter-update methods require a white-box language model and are highly computationally expensive. In Section 5, we present our approach MICE, which can be applied to large black-box language models.

Baselines We evaluate the following state-of-the-art knowledge editing approaches on our datasets: **Fine-tuning (FT)** (Zhu et al., 2020), **MEND** (Mitchell et al., 2021), **ROME** (Meng et al., 2022a), **MEMIT** (Meng et al., 2022b) and **PROMPT** (Wang et al., 2024).

Evaluation metrics Besides metrics we proposed, we also follow the commonly used metrics (Meng et al., 2022a; Wang et al., 2024) to conduct *definition-level* evaluation. **Reliability (RE)** measures the mean accuracy on a specific collection of pre-defined input-output pairs. **Generalization (GE)** measures the average accuracy on equivalent neighbor. **Locality (LO)** assesses the post-edit model remain unchanged in out-scope neighbor.

6.2 Main Results

We conduct experiments on RelEdit benchmark to evaluate the performance of 1) all baselines on conceptual knowledge editing; 2) our proposed MICE on conceptual knowledge editing.

Comparison among baseline editing methods. We report the evaluation results of **all baselines** on RelEdit benchmark in Table 3. It can be found that larger models are better handling relational reasoning challenge during conceptual knowledge editing. Considering results among five methods, we can observe that methods like ROME and MEMIT achieve great performance on traditional metrics like *reliability*, *generation* and *locality*, but struggle to handle relational reasoning challenges related to conceptual editing (e.g., from 16 to 68 on AB criteria across all models in intra setting). This demonstrates that basic conceptual editing methods mostly focus on updating local factual knowledge but neglect the propagation of related concepts and instances. Besides, comparing results across evaluation criteria shows that some relational reasoning challenges are handled better than others. To be specific, IL mainly remains a high score across methods on four baselines with positive correlations to Locality (e.g., approximately 80-100 on both settings, compared to AB range from 16 to 84), while other criteria is generally low and vary greatly among methods. PROMPT practically achieves the best performance on both instance-level and concept-level metrics, while slightly lags behind ROME and MEMIT on traditional metrics. This indicates PROMPT captures relational conceptual knowledge due to the update editing.

Performance of MICE. We apply MICE on dour backbone models, and Table 3 shows the performance of MICE on intra setting and inter setting of RelEdit. We find that with the same base model, MICE outperforms FT, MEND and PROMPT significantly across all the settings while being more efficient and requiring no training. We find that in-context learning methods (MICE and PROMPT) significantly outperform traditional retraining methods in PO, IL and AB. Intuitively, locate-and-edit methods generalize well on original instances that are strongly related to concepts explicitly contained in edit prompts. This explains why ROME and MEMIT perform better on IC but lags behind in PO. Overall, the results suggest that MICE works particularly well on strong base language models. Along with its simplicity and efficacy, MICE can serve as a strong conceptual knowledge editing baseline for future research.

6.3 Analysis

Q1: How does the high-level superclass of edited concepts affect the relational updates of broader facts? We analyze the effect induced by knowledge editing methods to the model's knowledge in different settings, to investigate the impact of the homogeneity between origin concept C and target concept C^*. We calculate the average scores of editing methods on RelEdit criteria, and results in Figure 4 show that scores in intra setting outperforms those in inter setting. Furthermore, the scores of concept-level (AB) between intra and inter settings (91.10 v.s. 77.65) have a larger gap than those of instance-level (IC and PO).

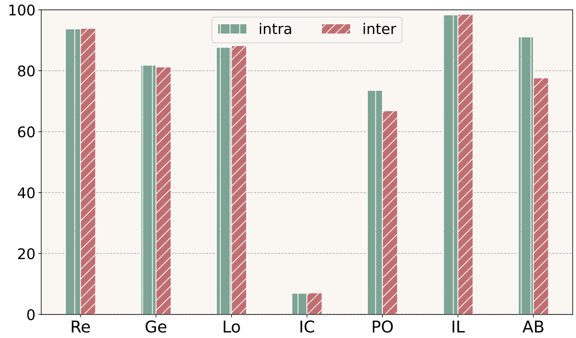
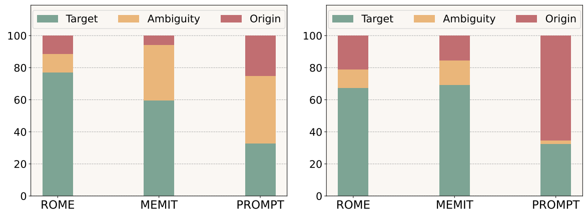


Figure 4: The average scores of Mistral-7B in both intra and inter settings. Results are averaged over MEND, ROME, MEMIT and PROMPT.

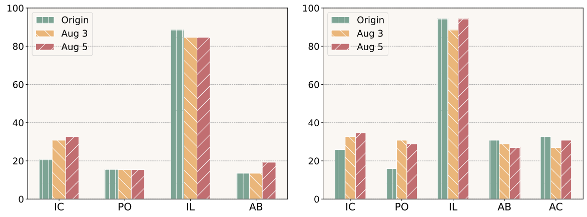
This phenomenon can be attributed to the diverse superclasses of concepts. In intra setting, LLM pre-exists higher-level connection of two concepts, and they share similar semantic structures and are likely to be represented in close proximity within the model's internal knowledge manifold. As a result, updates to one concept can naturally propagate to related concepts and their instances, with minimal disruption to the overall knowledge structure. This explains the worse performance of editing models on concept-level criteria in inter setting, which indicates editing knowledge in such a heterogeneous context requires the model to form new or cross-cutting associations that may not exist in its pre-trained structure. This not only increases the difficulty of precise knowledge insertion but also raises the risk of unintended interference with unrelated concepts.

Q2: What does instance-level criteria reflect about the knowledge that LLMs captured in conceptual editing? Considering IC, PO and IL scores of LLaMA-7B in Table 3, it is evidenced that ROME and MEMIT outperform PROMPT in instance-level criteria. To further explore the capabilities that instance-level criteria reflects about edited model, we evaluate the Concept Consistency (Wang et al., 2024) of LLaMA-2-7B with ROME, MEMIT and PROMPT using GPT4 API, and results are shown in Figure 5. It can be observed that ROME and MEMIT achieve great performance in editing conceptual knowledge in a semantic perspective. This evaluation demonstrates that models with strong instance-level scores tend to excel in capturing the semantic modifications from concept edits. Instance-level reflects the ability of LLMs to update knowledge from related in-



(a) LLaMA-2 (Intra setting) (b) LLaMA-2 (Inter setting)

Figure 5: Concept consistency on LLaMA-2-7B with ROME, MEMIT and PROMPT. *Target* means the generated sentence has close semantics to target, vice versa to *Origin*. *Ambiguity* means neither similar to both.



(a) Intra Setting (b) Inter Setting

Figure 6: Augmented massive concept editing of MEMIT on RelEdit with LLaMA-2-7B. *Origin* indicates that MEMIT without augmented edits.

stances of target concepts, equipping them with higher concept consistency.

Q3: Whether massive editing with related concepts facilitate model abilities to conduct relational reasoning regarding conceptual knowledge? To further explore the mechanism of editing models in relational reasoning regarding conceptual knowledge, we conduct experiments by massive editing of MEMIT with augmented concept editing prompts. We employ a few-shot approach (Brown et al., 2020) to prompt LLM to generate the target concept name according to target description d^*, along with instances that belong to target concept. Detailed prompts are provided in Appendix A.1. With obtained augmentation edits, we apply n-batch massive edit of MEMIT on RelEdit, aiming to incorporate related instance-level knowledge into LLM. As shown in Figure 6, we compare the orgin MEMIT to two variants with different number of augmented edits, and results show that simply introduce instance-level augmentations are beneficial for improving instance-level criteria (IC and PO), especially in inter setting. Furthermore, it is evidenced that incorporating more augmentation edits can slightly increase the performance on IC and PO, but leads to a marginal drop on concept-level criteria like AC.

7 Related work

Methods for knowledge editing. Various techniques have been proposed to edit the knowledge stored in a model. Some of these approaches involve identifying and adjusting the model's weights that correspond to specific concepts. Notable examples include KN (Dai et al., 2021), ROME (Meng et al., 2022a), and MEMIT (Meng et al., 2022b). Another line of research adopts meta-learning and utilizes a hyper-network, which is a smaller network responsible for generating the edited gradients. This category includes methods like KE (De Cao et al., 2021), MALMEN (Tan et al., 2023), and MEND (Mitchell et al., 2021). Additionally, there are preservative methods that incorporate explicit memory and prompting techniques to rectify model predictions. Noteworthy examples in this category are SERAC (Mitchell et al., 2022), MemPrompt (Madaan et al., 2022), MeLLo (Zhong et al., 2023), and IKE (Zheng et al., 2023).

Evaluation of conceptual knowledge editing. Existing knowledge editing evaluations mainly focus on factual knowledge. Initially, the assessment limited to verifying whether the target had been successfully modified and ensuring that unrelated details remained unaffected (Yao et al., 2023; Zhong et al., 2023). Recently, several benchmarks argue that the rippling changes in factual knowledge editing should also be evaluated (Cohen et al., 2024; Li et al., 2024; Ma et al., 2024). However, conceptual knowledge differs in form from concrete factual knowledge, and there is no systematic evaluation of conceptual knowledge editing to date. The only existing benchmark, ConceptEdit (Wang et al., 2024), concentrates on measuring whether the edited model can recall newly injected concept definitions. Complementing existing evaluation tools, RelEdit focuses on assessing whether edited models can answer relational reasoning questions.

8 Conclusion

In this work, we introduce a benchmark called RelEdit, which evaluates conceptual knowledge editing of LLMs. We find that existing knowledge editing methods struggle with answering questions related to specific instances and broader concepts. Additionally, we show that a simple in-context editing method with an external memory achieves the best results on RelEdit, highlighting the potential of such editing approaches.

Limitations

The limitations of our work are as follows.

- Although RelEdit focuses on the ripple effect of edits, it does not explicitly verify the item-part relationships between concepts. For instance, if we modify the definition of a wheel, it raises the question of how the definition of a car should be adjusted accordingly, as a wheel is a part of a car. Currently, there is no established and universally accepted rule for addressing this issue. We note that building such an evaluation is hard.

- RelEdit does not include concepts with more than one meaning. Existing knowledge editing methods cannot handle concepts with multiple meanings, thus RelEdit does not include polysemous concepts. It is an important future direction for conceptual knowledge editing research.

- In our data generation pipeline, we depend on concepts extracted from an existing knowledge base, specifically DBpedia, which could be incomplete or outdated. These concerns might be an issue when aiming for a comprehensive evaluation. An alternative solution worth exploring is to utilize the internal knowledge of language models instead of relying solely on external knowledge bases.

- Considering the limitations of computational resources, our evaluation of existing knowledge editing methods primarily focuses on limited models. We leave the evaluation on other models as future work.

Ethics Statement

This study adheres strictly to the most rigorous ethical standards and best practices in research. All data utilized are extracted from datasets that are available to the public, thereby ensuring no usage of any proprietary or sensitive information. As a result, this research is free from any ethical concerns. We have implemented measures to reduce the presence of offensive content in our dataset. Throughout the construction process, we utilized rigorous filtering methods to identify and remove material that could be deemed harmful or inappropriate.

In our research, we examine LLMs' sensitivities and preferences regarding concept-related updates and edits through our proposed benchmark and experimental findings. This focus aligns with recent recognition of concept editing as a crucial research direction for language models (Suresh et al., 2023; Wu et al., 2023b). Our work aims to contribute to the development of more robust LLMs, ultimately advancing the field's understanding of model behavior during knowledge updates.

Deliberately editing the concept definitions of LLMs may raise serious ethical issues. If the purpose of editing is to introduce erroneous meanings or misleading reasoning, this may lead to the model generating harmful content. When editing an LLM in research, it is important to update correct knowledge, which is necessary and beneficial. Researchers need to consider whether the output of the model is consistent with social values and ethical standards.

Acknowledgments

This research was supported by Guangdong S&T Program C019.

References

2024. Qwen2 technical report.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *international semantic web conference*, pages 722–735. Springer.

Baichuan. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

Andreas Holzinger, Anna Saranti, Alessa Angerschmid, Bettina Finzel, Ute Schmid, and Heimo Mueller. 2023. Toward human-level concept learning: Pattern benchmarking for ai algorithms. *Patterns*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *CoRR*, abs/2112.09118.

Mohsen Jamali, Ziv M Williams, and Jing Cai. 2023. Unveiling theory of mind in large language models: A parallel to single neurons in the human brain. *arXiv preprint arXiv:2309.01660*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. 2024. Unveiling the pitfalls of knowledge editing for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Michelle Lo, Shay B Cohen, and Fazl Barez. 2024. Large language models relearn removed concepts. *arXiv preprint arXiv:2401.01814*.

Yaojia Lv, Haojie Pan, Ruiji Fu, Ming Liu, Zhongyuan Wang, and Bing Qin. 2024. Coggpt: Unleashing the power of cognitive dynamics on large language models. *arXiv preprint arXiv:2401.08438*.

Jun-Yu Ma, Jia-Chen Gu, Ningyu Zhang, and Zhen-Hua Ling. 2024. Neighboring perturbations of knowledge editing on large language models. *CoRR*, abs/2401.17623.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve gpt-3 after deployment. *arXiv preprint arXiv:2201.06009*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

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

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

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

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. *arXiv preprint arXiv:2004.00345*.

Siddharth Suresh, Kushin Mukherjee, Xizheng Yu, Wei-Chun Huang, Lisa Padua, and Timothy Rogers. 2023. Conceptual structure coheres in human cognition but not in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 722–738.

Chenmien Tan, Ge Zhang, and Jie Fu. 2023. Massive editing for large language model via meta learning. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023a. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Xiaohan Wang, Shengyu Mao, Ningyu Zhang, Shumin Deng, Yunzhi Yao, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024. Editing conceptual knowledge for large language models. *arXiv preprint arXiv:2403.06259*.

Weiqi Wu, Chengyue Jiang, Yong Jiang, Pengjun Xie, and Kewei Tu. 2023a. Do plms know and understand ontological knowledge? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3080–3101.

Weiqi Wu, Chengyue Jiang, Yong Jiang, Pengjun Xie, and Kewei Tu. 2023b. Do plms know and understand ontological knowledge? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 3080–3101.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10222–10240. Association for Computational Linguistics.

Bonan Zhao, Christopher G Lucas, and Neil R Bramley. 2024. A model of conceptual bootstrapping in human cognition. *Nature Human Behaviour*, 8(1):125–136.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

A Benchmark Construction Details

A.1 Templates

Here we numerates a variety of templates employed within our experimental framework.

As mentioned in Question 3 of Section 6.3 *Analysis*, we employ in-context learning approaches to prompt the LLM to generate augmented prompts based on editing requests.

The following Template 1 shows the few-shot prompt used for prompting the LLM itself to generate concept name according to the concept description text.

The following Template 2 shows the few-shot prompt used for prompting the LLM itself to generate instances according the concept name.

The structured template for Concept Consistency, as depicted in Template 3, serves as an input for the GPT-4 evaluator. This template facilitates a qualitative analysis that categorizes the generated sentences into three distinct scores. By employing a relative comparison instead of assigning fixed values, the approach recognizes the evaluator's proficiency, which has been preliminarily confirmed to be more closely aligned with human judgment. This method allows for a more accurate evaluation of the generated content.

Template 1: Concept Name Generation Prompt

Given a description X about a concept, please generate the name Y of the concept according to the description X. The Name Y has to be a named entity and as short as possible.

Description X: company that prints and distributes pressed goods or electronic media
Name Y: publisher

Description X: those who serve as part of an organized armed military force
Name Y: military person

[8 in-context demonstrations abbreviated]

Description X: {new concept description}
Name Y:

Template 2: Instance Generation Prompt

Given a name X of a concept, please generate a list Y containing 5 instances that belong to the concept X. The list Y has to be a list including 10 named entities, divided by identifier ','.

Concept Name X: publisher
Instance List Y: [Virus Music, Famitsu Bunko, BitComposer, Victoria University Press, BBC Audio]

Concept Name X: military person
Instance List Y: [Ronald Reid-Daly, Charles Augustus Hilton, 27th Indiana Infantry Regiment, Spartaco Schergat, Charles Corcoran]

[8 in-context demonstrations abbreviated]

Concept Name X: {new concept name}
Instance List Y:

Template 3: Concept Consistency Evaluation Prompt

Prediction sentence: {Generated sentence}
Sentence A:{Target edit sentence}
Sentence B:{Origin sentence}

Check the prediction sentence and Give a score from -1 to 1:
Score 1: close meaning to sentence A
Score 0: neither relevant to A nor B
Score -1: close meaning to sentence B

Output format is: Score:{}
Only output group name and corresponding score, no other explanation.

A.2 Data Distribution of RelEdit

We further report the data distribution of our proposed RelEdit benchmark in Figure 7, calculated with token length. Figure 7a shows that RelEdit maintain a wide range of description token length from 3 to 45, and most of them are around 10. The wide range of concept descriptions increase the difficulty of conceptual editing. From Figure 7b and

	Annotator-1	Annotator-2	Annotator-3
Agreement Rate	95.8	93.6	94.4

Table 4: Human evaluation results on RelEdit

Figure 7c, we can tell that token length of concept name is mainly around 3-4, and token length of instance name is mainly around 6-7, which bring impact to instance-level evaluation criteria.

A.3 Human Evaluation

We recruited three volunteers to evaluate the entailment between the origin concept description and corresponding concept name. Each volunteer assessed a sample of 300 randomly selected examples to ensure the quality and reliability of our dataset. They were tasked with "Determining whether the origin concept description semantically supports the corresponding concept name." The human evaluation results are listed in Table 4. The high agreement observed further supports our dataset's quality and validity.

A.4 Case of Conceptual Editing Requests

Specifically, we provide a detailed conceptual editing case of RelEdit benchmark in inter setting. As shown in Table 5, the detailed textual information and editing prompts in both instance-level and concept-level elaborate the multi-aspect conceptual knowledge covered in a single editing request.

A.5 SPARQL Protocol and RDF Query Language

SPARQL facilitates the extraction and modification of data that is housed within the Resource Description Framework (RDF), a system adept at representing graph-based data structures. The Wikidata Query Service[1] (WDQS) is an internet-based platform which empowers users to fetch and scrutinize the organized data contained within Wikidata by utilizing SPARQL queries. We employ WDQS to query the description texts for each concept in Section 3.1 *Ontology Building*, and the SPARQL we used is listed in Table 7.

B MICE

MICE explicitly stores all edited concepts in memory, with each concept represented as a pair of concept name and definition. An off-the-shelf retrieval model (Izacard et al., 2021) is utilized to embed the concept names and save them in a retrieval index. This index takes a query as input and returns the edited concept most relevant to the query (i.e., closest in the embedding space). For step-by-step generation and self-checking with LLMs, we follow the approach of previous works (Zhong et al., 2023). Specifically, the model is prompted to verify whether the retrieved fact contradicts the generated answer. If a subquestion is unrelated to any edited concept in memory (because the corresponding concept was not edited), the model is prompted to retain the generated answer, as the retrieved edit does not cause a contradiction. The final answer is then generated based on this process.

C Experimental Details

C.1 Language Models

To adequately evaluate knowledge editing methods on RelEdit, we use following LLMs as the base model: GPT2-XL (Radford et al., 2019b), GPT-J-6B (Wang and Komatsuzaki, 2021), LLaMA2 (7B, 13B) (Touvron et al., 2023b), Mistral-7B-v0.1 (Jiang et al., 2023), Baichuan2 (7B, 13B) (Baichuan, 2023) and Qwen2 (0.5B, 1.5B, 7B) (Qwe, 2024). All models can be reached in Huggingface website [2].

C.2 Metrics

The commonly used metrics are from ROME (Meng et al., 2022a).

Reliability (RE). This metric measures the mean accuracy on a specific collection of predefined input-output pairs $(x_{\mathrm{e}}, y_{\mathrm{e}})$:

$$\mathbb{E}_{x'_{\mathrm{e}},y'_{\mathrm{e}}\sim\{(x_{\mathrm{e}},y_{\mathrm{e}})\}}\mathbb{1}\left\{\operatorname{argmax}_y f_{\theta_e}\left(y \mid x'_{\mathrm{e}}\right)=y'_{\mathrm{e}}\right\} \tag{6}$$

Generalization (GE). Paraphrased sentences should be modified accordingly by editing. This metric gauges the average accuracy on equivalent neighbor $R\left(x_{\mathrm{e}}, y_{\mathrm{e}}\right)$:

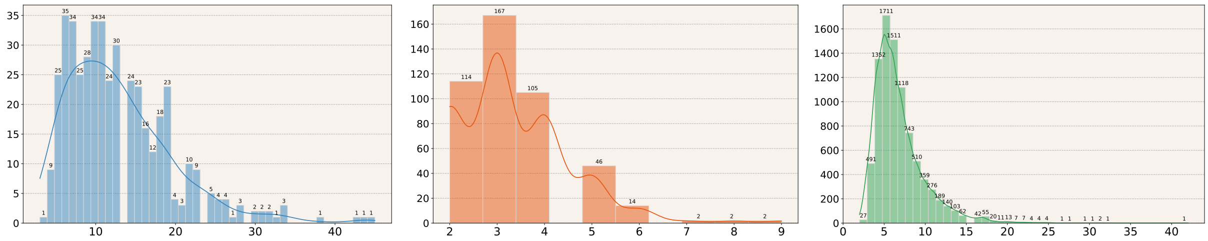
$$\mathbb{E}_{x'_{\mathrm{e}},y'_{\mathrm{e}}\sim R(x_{\mathrm{e}},y_{\mathrm{e}})}\mathbb{1}\left\{\operatorname{argmax}_y f_{\theta_e}\left(y \mid x'_{\mathrm{e}}\right)=y'_{\mathrm{e}}\right\} \tag{7}$$

Locality (LO). Locality is assessed based on the frequency at which the predictions of the post-edit model remain unchanged in out-scope neighbor $O\left(x_{\mathrm{e}}, y_{\mathrm{e}}\right)$:

$$\mathbb{E}_{x'_{\mathrm{e}},y'_{\mathrm{e}}\sim O(x_{\mathrm{e}},y_{\mathrm{e}})}\mathbb{1}\left\{f_{\theta_e}\left(y \mid x'_{\mathrm{e}}\right)=f_{\theta}\left(y \mid x'_{\mathrm{e}}\right)\right\} \tag{8}$$

[1] https://query.wikidata.org

[2] https://huggingface.co



(a) Distribution of Concept Descriptions

(b) Distribution of Concept Names

(c) Distribution of Instance Names

Figure 7: Token length distribution of concept descriptions(a), concept names(b) and instance names(c) in RelEdit benchmark for conceptual editing. Texts above are tokenized by the LLaMA-2-7B tokenizer.

C.3 Baselines

We evaluate the following state-of-the-art knowledge editing approaches on our datasets: **Fine-tuning (FT)** (Zhu et al., 2020), **MEND** (Mitchell et al., 2021), **ROME** (Meng et al., 2022a), **MEMIT** (Meng et al., 2022b) and **PROMPT** (Wang et al., 2024).

- **Fine-tuning (FT)** approach uses gradient descent to update model parameters based on the edits. In our study, we adopt the method (Zhu et al., 2020), where we fine-tune one layer with a norm constraint on weight changes.

- **MEND** (Mitchell et al., 2021) trains a hypernetwork to produce weight updates by transforming the raw fine-tuning gradients.

- **ROME** (Meng et al., 2022a) first identifying the knowledge within a specific layer of the Transformer architecture. It then updates the feedforward network within that layer.

- **MEMIT** (Meng et al., 2022b) is an extension of ROME that allows for batch editing. It updates the feedforward networks across multiple layers to encode all the edited facts.

- **PROMPT** (Wang et al., 2024) serves as the prefix is concatenated to the beginning of the evaluation input to elicit the desired modification in outputs.

C.4 Implementation Details

For the code implementation, we adopt EasyEdit[3] repository to reimplement all baselines including FT (FT-L implementation), ROME, MEND, MEMIT and PROMPT. The editing procedure is conducted in a independent manner, focusing exclusively on a single specified concept with each edit instead of sequentially, ensuring that each modification is made in isolation. Once the evaluation for a given sample is finished, the LLM is reverted to its original state, prior to any edits being applied. This approach guarantees that no subsequent edits are influenced by the previous ones.

For the settings of all baselines, we adopt their default configurations. All experiments are carried out on a single A800 GPU. To be specific, we list important hyper-parameters of baselines as follows:

FT: FT executes a gradient descent operation on the modifications to refine the model parameters. Following Zhu et al. (2021), FT uses a norm constraint on weight changes with a coefficient 5×10^{-5} in our implementation. We finetune layer 0 of GPT2-xl, layer 21 of GPT-J-6B, LLaMA-2-7B and Mistral-7B-v0.1.

MEND: Since MEND requires a trained hypernetwork to update parameters, we construct an extra dataset with train/validation/test sets to train a hypernetwork specifical to conceptual editing task. During training, we set the max iteration step step 10000. During inference, the learning rate scale is set to be 1.0.

ROME: We use the default hyper-parameters of ROME and the pre-computed covariance statistics released by Meng et al. (2022a). The edit layer in GPT2-xl is set to 17, in GPT-J-6B, LLaMA-2-7B, Mistral-7B-v0.1 and Baichuan2-7B is set to 5.

MEMIT: To apply MEMIT on all LLM backbones, we compute the covariance statistics for Vicuna-7B on Wikitext using a sample size of 100,000. For GPT2-xl, we updated model weights at layer $\{13, 14, 15, 16, 17\}$. For GPT-J-6B, we updated model weights at layer $\{3, 4, 5, 6, 7, 8\}$. For LLaMA-2-7B, Mistral-7B-v0.1 and Baichuan2-7B, we updated model weights at layer $\{4, 5, 6, 7, 8\}$.

[3] https://github.com/zjunlp/EasyEdit

Concept Infomation	
Subject Concept	Island
Subject Superclass	Place
Subject Description	Piece of sub-continental land completely surrounded by water
Subject Instances	Saba Island, United States Virgin Islands Liard Island Shaw Islan
Target Concept	American football player
Target Superclass	Species
Target Description	Athlete who plays American football
Target Instances	Adam Bisnowaty Marvin Hall Siran Stacy
Conceptual editing Prompts	
Base Editing	The definition of island is athlete who plays American football
Phrase	To put it simply, island refers to athlete who plays American football
Locality	The definition of galaxy is large gravitationally bound system of stars and interstellar matter
Instance-level editing Prompts	
Instance Change	Whether Saba Island, United States Virgin Islands belongs to category island?
Portability	Whether Adam Bisnowaty belongs to category island?
Instance Locality	Whether Salyut 6 belong to category space station?
Concept-level editing Prompts	
Alignment Compare	Do island and american football player belong to the same superclass?
Alignment Belong	Whether island belongs to superclass species?

Table 5: **An example of conceptual editing requests in RelEdit (under inter setting).** The subject conceptual knowledge and target conceptual knowledge required in a single editing procedure are highlighted in blue and red, respectively. Due to the limitation of space, some instances of concept are omitted.

PROMPT: We employ In-context Learning to prompt LLMs to elicit the desired modification. The detailed prompt we use during editing is to include a definition about new concept before exact editing request, which is formulated as "Definition of [subject concept]: [target description]".

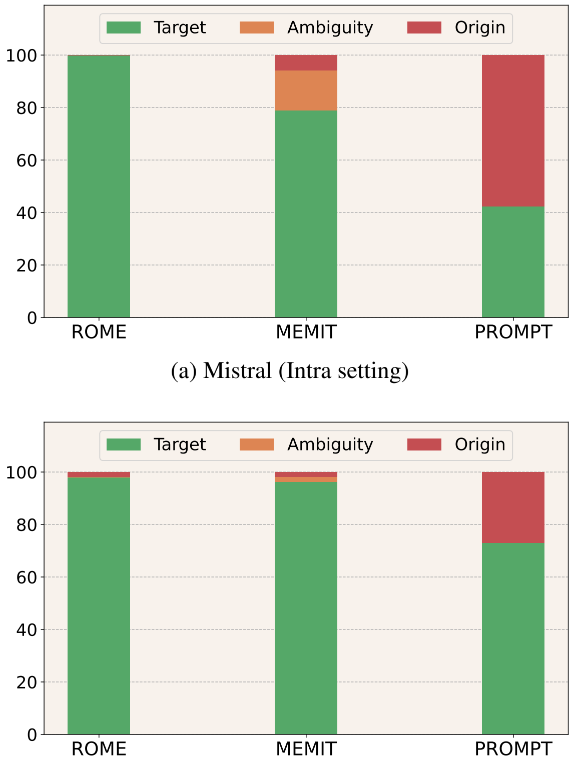
D Additional Experiments

D.1 More results about concept consistency

Additional to Section 6.3 *Analysis* Q2, we provide more results evaluated on concept consistency of Mistral-7B with ROME, MEMIT and PROMPT using GPT-4 API, as shown in Fig 8.

D.2 Impact of LLM Size on Conceptual Editing

We analyze how conceptual editing performance on RelEdit is affected by the LLM size. Specifically, we conduct conceptual editing on LLaMA2, Baichuan2 and Qwen2 with different amount of parameters, and detailed results are shown in Table 6. From this table, we can observe that: the editing performance generally improves with the increase in the number of LLM parameters, especially on factual criteria. LLMs with larger amount parameters tend to perform better on instance-level criteria, but facing a performance drop on concept-level metric like AB. These results indicate that parameter volume of LLMs positively affects the knowledge editing performance in the factual level



(a) Mistral (Intra setting)

(b) Mistral (Inter setting)

Figure 8: The results of concept consistency on Mistral-7B with ROME, MEMIT and PROMPT in both intra and inter settings. *Target* means the generated sentence has close semantics to target, vice versa to *Origin*. *Ambiguity* means neither similar to both.

to a certain extent. However, they also face the challenge of updating knowledge in the ontological level, as models with a larger number of parameters involve more types of relational reasoning challenges, making it more difficult to thoroughly update ontological knowledge from the root.

D.3 Conceptual Patterns across Editing Methods

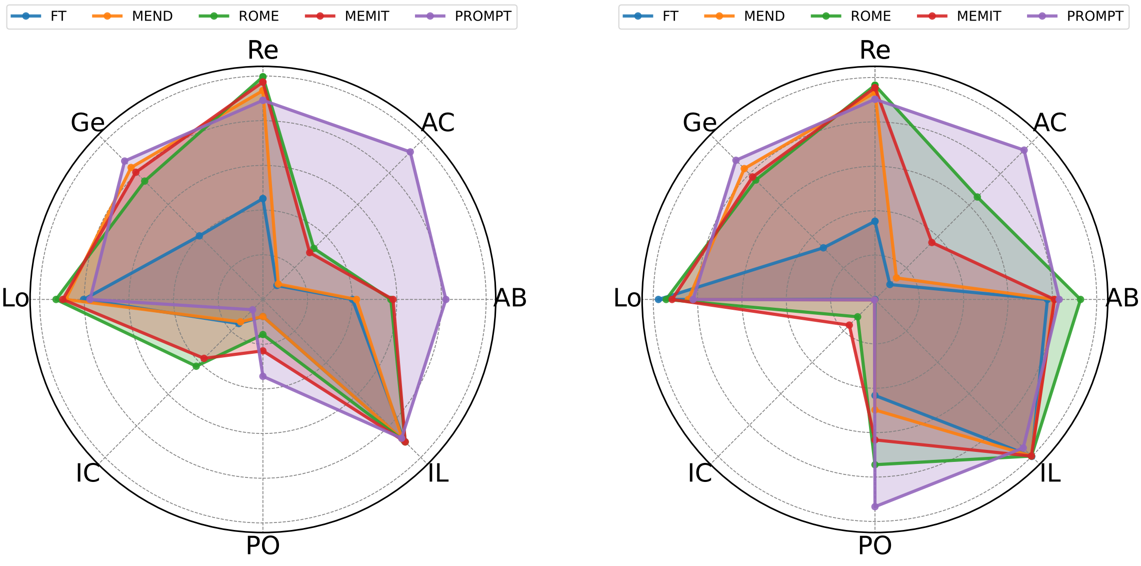
According to the main results in Table 3, we further conduct a thorough analysis on conceptual patterns that different methods captured during editing procedure. From Figure 9, we can observe that among five editing methods, PROMPT yields leading performance on instance-level and concept-level criteria like IC, PO, IL, AB and AC. On the contrary, locate-and-edit methods like ROME and MEMIT achieve strong performance on metrics Re, Ge and Lo.

D.4 Case of Editing Results

Table 8 presents several editing results with ROME on RelEdit benchmark in inter setting, listing three cases of ROME method in varying degrees of editing effectiveness. Case 1 is an ideally successful editing, as the generated sentence from post-edit model is the same as editing target in token-level. Case 2 reflects a partial successful editing, since there is an overlap between target sentence and output sentence. Case 3 shows an unsuccessful editing on concept cyclist, because the output sentence from post-edit model exactly inherited the origin conceptual knowledge.

Base Model	**Method**	**Intra Setting**							**Inter Setting**							
		Re	Ge	Lo	IC	PO	IL	AB	Re	Ge	Lo	IC	PO	IL	AB	AC
LLaMA-2-7B	PROMPT	**89.35**	**87.28**	76.84	6.86	38.50	**89.82**	**84.51**	**88.88**	87.89	78.06	6.19	30.31	**85.84**	**79.42**	**93.36**
LLaMA-2-13B	PROMPT	88.43	87.08	**81.13**	**8.19**	**41.81**	84.29	58.63	88.57	**87.93**	**81.58**	**7.74**	**36.73**	**85.84**	49.56	80.09
Baichuan2-7B	PROMPT	**89.41**	87.89	77.36	4.87	**72.79**	**90.71**	**96.90**	**89.05**	87.70	78.08	4.87	**70.80**	**93.14**	**90.71**	**98.23**
Baichuan2-13B	PROMPT	89.17	**88.08**	**80.58**	**34.29**	30.97	65.27	37.61	88.41	**87.85**	**80.64**	**35.62**	24.78	68.36	28.10	73.01
Qwen2-0.5B	PROMPT	88.32	86.69	72.24	1.11	**96.02**	58.63	**95.13**	88.48	86.67	73.08	0.88	**95.58**	56.86	**94.03**	46.68
Qwen2-1.5B	PROMPT	88.62	86.56	78.61	**5.75**	63.72	92.92	90.71	88.30	87.12	79.43	**8.41**	56.42	91.15	81.42	**92.04**
Qwen2-7B	PROMPT	**88.85**	**87.56**	**80.53**	1.33	68.36	**94.03**	56.19	**88.89**	**87.67**	**80.83**	1.11	58.63	**93.36**	40.49	79.87
Qwen2-0.5B	ROME	98.95	72.84	85.32	30.31	55.09	98.01	22.79	99.02	71.03	85.84	33.41	51.33	98.23	20.80	6.86
Qwen2-1.5B	ROME	99.07	**80.77**	**92.64**	**21.46**	**61.95**	**98.67**	**73.89**	99.65	**81.03**	**92.81**	**32.30**	**57.52**	98.89	**70.13**	**30.97**
Qwen2-7B	ROME	**99.85**	75.34	92.41	18.81	49.56	98.45	59.96	**99.88**	73.84	92.60	22.79	45.58	**99.34**	53.76	60.84

Table 6: Main results of PROMPT and ROME across base models LLaMA-2, Baichuan2 and Qwen2 with varying number of parameters. The best results are in **bold**. **Re**, **Ge** and **Lo** are the abbreviation of metric Reliability, Generalization and Locality.



(a) LLaMA-2-7B

(b) Mistral-7B-v0.1

Figure 9: Conceptual editing performance of FT, MEND, ROME, MEMIT and PROMPT on backbones LLaMA-2-7B(a) and Mistral-7B-v0.1(b). The results reported are averaged between intra and inter setting.

SPARQL for Extracting Concept Description

```
PREFIX bd: <http://www.bigdata.com/rdf#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX schema: <http://schema.org/>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wikibase: <http://wikiba.se/ontology#>

SELECT ?conceptLabel ?conceptDesc
WHERE {
  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "en" .
    wd:<QID> rdfs:label ?conceptLabel .
    wd:<QID> schema:description ?conceptDesc .
  }
}
```

Table 7: SPARQL Query for extracting concept description based on concept entity id (denoted by "<QID>").

CASE 1	
Concept	noble
[Origin]	member of the nobility
[Target]	group of players organized to compete as a side in baseball
[Generated]	group of players organized to compete as a side in baseball
Rewrite Accuracy	1.000
CASE 2	
Concept	continent
[Origin]	large landmass identified by convention
[Target]	an athlete who participates in traditional Irish sports such as Gaelic football, hurling, camogie, or handball
[Generated]	a person who participates in a sport that involves a combination of rugby, rugby league, or American football
Rewrite Confidence	0.7826
CASE 3	
Concept	cyclist
[Origin]	person who rides a bike
[Target]	set of episodes produced for a television series
[Generated]	a person who rides a bicycle
Rewrite Accuracy	0.375

Table 8: **Examples of conceptual editing results with ROME in inter setting. [Origin]** refers to the initial concept recognition before editing, **[Target]** denotes the ideal concept description after editing and **[Generated]** represents the generated sentence from post-edit model.