FAME: Factual Multi-task Model Editing Benchmark

Anonymous ACL submission

Abstract

Large language models (LLMs) possess the capability to retain a wide range of knowledge, albeit they also show tendencies for factual inaccuracies. To rectify such inaccuracies without the necessity for costly model retraining, a variety of model editing approaches have been proposed, aiming to correct these inaccuracies in a more cost-efficient way. To evaluate these model editing methods, previous work had introduced a series of datasets. However, most of these datasets use fabricated data, rendering them incapable of evaluating or improving the capabilities of models. Additionally, they only include a single task, preventing them from comprehensively simulating the real world. To resolve these challenges and effectively enhance the capabilities of LLMs, we present FAME (**FA**ctual **M**ulti-task model **E**diting), an authentic, comprehensive, and multi-task dataset, which is designed to amplify the practicality of model editing. We then propose SKEME (**S**tructured **K**nowledge retrieved by **E**xact **M**atching and reranking **E**diting), a model editing technique predicated on structured knowledge retrieval. The experiments demonstrate that our method performs excellently across various tasks and scenarios, confirming its practicality.[1]

1 Introduction

Large language models (LLMs) have achieved remarkable capabilities across various domains and are extensively utilized in practical applications (Touvron et al., 2023a,b; Openai, 2023; Geva et al., 2020, 2022). The extensive utilization of LLMs necessitates the provision of precise information by LLMs. However, LLMs may still provide erroneous information due to incorrect, outdated knowledge stored within the model (De Cao et al., 2021; Agarwal and Nenkova, 2022). To avoid costly retraining and to efficiently correct the outputs of LLMs, model editing has been proposed (Mitchell et al., 2022; Sinitsin et al., 2020; De Cao et al., 2021).

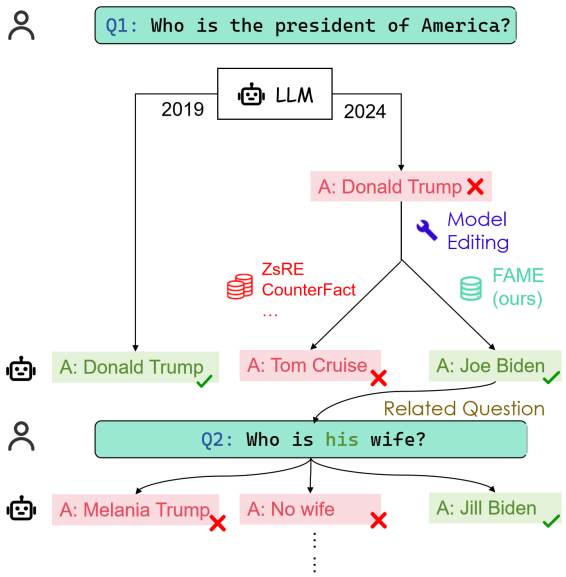

Figure 1: An example of FAME. LLMs may develop factual inaccuracies over time, which can be corrected through model editing. While previous datasets employed fabricated data, FAME utilizes real-world data to improve the performance of LLMs in practical usage.

To evaluate model editing methods, previous works have introduced a series of datasets (De Cao et al., 2021; Meng et al., 2022a; Zhong et al., 2023). Almost all of these datasets modify a portion of the original real facts to obtain the newly constructed facts, which affects the model's practical performance and contradicts the original purpose of model editing. As shown in Figure 1, when user asks "Who is the President of America?", LLMs produce incorrect output due to outdated knowledge. Previous datasets modified them into incorrect targets (for example, Tom Cruise), while our dataset focuses on real-world facts (Joe Biden). Moreover, these datasets are all composed of data in a single format with a single task like QA (Levy

[1]Dataset and codes are publicly available at https://AnonymousLink

et al., 2017) and sentence completion (Meng et al., 2022a), making them unable to achieve a thorough evaluation of the effectiveness of model editing methods.

In the real world, LLMs are expected to possess knowledge about the real world and handle diverse forms of input. Therefore, possessing real facts and data in various tasks is crucial for model editing datasets. However, to the best of our knowledge, previous work falls short in achieving these two aspects.

To address counterfactual data and single objective, we introduce FAME, a factual, extensive model editing benchmark with practicality. FAME comprises 128k real data items, including various tasks with single-hop and multi-hop questions. In response to the drawback of the previous datasets being fabricated, we extract factual data items from Wikidata(Vrandečić and Krötzsch, 2014) and DBpedia (Auer et al., 2007) and employed multiple rounds of manual verification to ensure the accuracy of our benchmark. To overcome the limitation that the previous datasets only comprise data in a single format, we incorporate tasks from existing datasets like QA (Levy et al., 2017), fact-check (Schuster et al., 2021), multi-hop QA (Zhong et al., 2023), and additionally introduced new tasks such as cloze and dialogue, making our evaluation more comprehensive. Our benchmark enhances the model's practical capabilities and enables a complete evaluation of the effectiveness of model editing.

To address the aforementioned challenges and enhance the practicality of LLMs, we propose a new method, SKEME. SKEME achieves precise matching and efficient knowledge application through a structured database, tackling the challenges posed by large-scale data and diverse tasks. Experimental results demonstrate that our method performs well in a range of simulated real-world scenarios, indicating its heightened practicality.

The main contributions of this paper are as follows:

- To mitigate challenges posed by counterfactual data and single-task limitations, and to support the needs of model editing in the real world, we create FAME, a benchmark that incorporates real-world data and covers various tasks.
- To meet the requirements of model editing in the real world, we propose a practical model editing method called SKEME, which involves the use of a caching mechanism to reflect real-world updates.
- We introduce new metrics and simulate real-world scenarios to evaluate existing model editing methods. Results indicate that previous methods lack practicality and can not handle the diverse scenarios in the real world. Our approach addresses this issue.

2 Related work

2.1 Model Editing Datasets

Model editing datasets serve the purpose of verifying the effectiveness of methods and enhancing the capability of LLMs. Nevertheless, current datasets fall short of directly enhancing the capability of LLMs. The majority of datasets comprised constructed fakedata (Levy et al., 2017; Meng et al., 2022a; Zhong et al., 2023; Gupta et al., 2023), primarily serving to validate effectiveness rather than directly contribute to the enhancement of LLMs' capabilities. MQuAKE-T (Zhong et al., 2023) utilizes modifications in Wikidata, which has the potential to directly enhance LLMs' practical performance. However, due to the limited amount (see Figure 2 for statistics), its direct utility in improving the performance of LLMs is limited, thereby primarily serving to validate effectiveness. In contrast to prior works, our benchmark sets itself apart by featuring a substantial repository of authentic data and integrating multiple diverse tasks. As a result, it holds the potential for direct implementation, exhibiting a heightened level of practical applicability.

2.2 Model Editing Methods

Previous works have introduced various model-editing methods, including methods that modify models' parameters and methods that preserve models' parameters(Yao et al., 2023). The former category includes the locate-then-method (Meng et al., 2022a,b) and meta-learning-based methods (De Cao et al., 2021; Mitchell et al., 2021). The latter category involves adding additional parameters to the model (Dai et al., 2022; Huang et al., 2023) and employing vector databases for knowledge storage (Mitchell et al., 2022; Zhong et al., 2023; Zheng et al., 2023; Cheng et al., 2023; Madaan et al., 2022). Diverging from the previously discussed methods, our approach involves a structured

knowledge base and implements precise matching during the search process.

3 Problems Definition

The objective of model editing is to modify the knowledge contained in a model, allowing the model to engage in reasoning processes based on the edited knowledge, while not affecting the output related to the unedited knowledge. Based on previous work (Wang et al., 2023b; Yao et al., 2023), we define model editing to express the goal as follows.

An input-output pair is defined as (x, y), and a model is represented by a function $f : X \rightarrow Y$, where X represents the input set and Y represents the output set. Let $I(x, y)$ denotes the set of descriptions semantically equivalent to (x, y), and $EX(x, y)$ be the set of input-output pairs that the model can possess with $I(x, y)$ as prior knowledge. Then, let $O(x, y)$ represent the portion outside $I(x, y)$ and $EX(x, y)$.

The present definition of model editing can be summarized as follows: (x, y) denotes the fact that is being edited, while (x_e, y_e) represents the input and output.

$$f'(x_e) = \begin{cases} y_e & (x_e, y_e) \in I(x, y) \\ f(x_e) \mid (x, y) & (x_e, y_e) \in EX(x, y) \\ f(x_e) & (x_e, y_e) \in O(x, y) \end{cases} \tag{1}$$

For detailed formal definitions and examples, please see Appendix B.

4 FAME: Our Benchmark

In this section, we introduce how we constructed our benchmark, FAME. FAME is a benchmark comprising 128k factual data items. We utilized this data to construct both single-hop and multi-hop questions. For single-hop questions, we include five forms: QA, sentence completion, cloze test, multiple-choice questions, and fact check. For multi-hop questions, we include general multi-hop questions and dialogues. The previous work introduced QA, sentence completion, fact check, and multi-hop questions (Wang et al., 2023b), while the remaining tasks were proposed by us. We believe that combining these tasks contributes to a comprehensive assessment of the effectiveness of model editing methods.

To ensure the data quality of FAME and its reflection of the real world, we conducted multiple rounds of manual verification and correction in various aspects.

4.1 Choose Fact Triples

Our dataset is based on Wikidata(Vrandečić and Krötzsch, 2014) and DBpedia (Auer et al., 2007), both of which are knowledge bases comprised of knowledge triplets. We aim to enhance the diversity of our data by collecting knowledge from a variety of knowledge bases.

Specifically, we initially identified equivalent relations in Wikidata and DBpedia, followed by rule-based filtering to eliminate code, numbers, and other irrelevant content. All remaining relations were selected to be included in our dataset.

Then, we collected triplets associated with these relations from Wikidata and DBpedia. After obtaining the triplets, we further filtered them to avoid potential ambiguity issues, see Appendix A.1 for details.

Finally, to ensure the quality of the triplets we obtained, we randomly selected 100 triplets and manually examined their correctness. The results indicate that 96% of the triplets are correct, which shows that our process for obtaining and filtering triplets is acceptable.

4.2 Generate Data Based on Templates

We employ ChatGPT in the generation process to mitigate expensive labor costs following Petroni et al. (2019). After generating the results, we conduct manual checks to ensure the accuracy and alignment with our intentions.

For single-hop questions, following previous works Yin et al. (2023); Elazar et al. (2021), we prompt ChatGPT to generate question templates based on the relationship and its description, incorporating placeholders. Subsequently, we replace these placeholders with subjects to generate questions from the templates.

For multi-hop questions, following MQuAKE (Zhong et al., 2023), we employed ChatGPT to concatenate multiple consecutive triplets into a single question. Moreover, prior work (Petroni et al., 2019; Zhong et al., 2021) suggests that prompting the model to decompose the multi-hop questions into multiple simple subquestions is beneficial. To distinguish between the differences in model decomposition ability and knowledge it knows, we decompose queries to the model for the subquestions in multi-turn dialogues.

Name	isC.	Tasks						Total	Re.	Source	Hop
		Cho.	FC.	Clo.	Dia.	Com.	QA				
ZSRE	✗	✗	✗	✗	✗	✗	✓	270K	120	WD.	Si.
COUNTERFACT	✗	✗	✗	✗	✗	✓	✗	2.2K	24	WD.	Si.
MQUAKE-CF	✗	✗	✗	✗	✗	✗	✓	9K	37	WD.	Mu.
MQUAKE-T	✓	✗	✗	✗	✗	✗	✓	1.8K	6	WD.	Mu.
FAME	✓	✓	✓	✓	✓	✓	✓	128K	87	WD. & DB.	Si. & Mu.

Table 1: Comparison between our dataset to other model edit datasets, incorrect means if the edit target is the real fact. We believe fabricated facts will decrease the model's performance. Cho stands for choose. FC stands for fact-checking. Clo stands for cloze. Dia. stand for dialogue. Com. stands for completion. isC stands for isCorrect, which means if the edit target is the real fact. Re stands for count of relations included. WD, stands for Wikidata, DB stands for DBpedia. Si. stands for single, Mu. stands for multi. They are used to distinguish whether the data in the dataset involves single-hop or multi-hop scenarios.

When templates were constructed, we incorporated manual verification focusing on ensuring that the templates align with the meaning of the relationships. We found that 97.4% of templates were accurate, we then manually performed multiple rounds of checking, correction, and rechecking, ensuring that we consistently agreed the correctness rate of the templates reached 100%.

Finally, following previous work(Yin et al., 2023), we employed manual sampling and verification techniques to ensure the accuracy of our data. We combine the templates and relation triplets and manually checke the credibility of the generated sentences. The results show that 97.5% of the sentences were credible, demonstrating the reliability of the entire process.

5 Benchmark Analysis

5.1 Comparisons

See Table 1 for a comparison between our benchmark and previous benchmarks. Our benchmark includes all categories seen in previous benchmarks, and we have proposed additional data categories. Moreover, the number of entries far exceeds those in previous benchmarks. Finally, our data originates from two distinct knowledge bases, making it more comprehensive compared to previous datasets.

Similar to MQuAKE-T (Zhong et al., 2023), our data consists of genuine knowledge rather than constructed false information. However, MQuAKE-T is designed for multi-hop questions and both the number of relations and the size are limited, making it challenging to use it to enhance model capabilities. Therefore, we are currently the only benchmark available that can augment these capabilities.

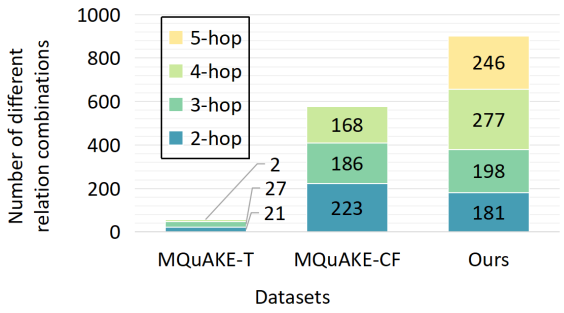

Figure 2: Comparison between multi-hop data in our dataset and MQuAKE. The vertical axis of the graph represents the number of relation combinations. Our dataset encompasses a greater number of combinations, including 5-hop questions, which effectively demonstrates the enhanced diversity of our dataset.

5.2 Analysis

Our data consists of two parts: single-hop data and multi-hop data, both sourced from Wikidata and DBpedia.

For the single-hop data, FAME include 87 relations. We selecte 20,000 data items, each containing 6 questions, resulting in a total of 120,000 distinct data items.

The multi-hop question data includes multi-hop QA and dialogue. Refer to Figure 2 for a comparison between our data and MQuAKE (Zhong et al., 2023). It can be observed that our multi-hop questions cover a higher number of relationships, indicating that our data is more complex.

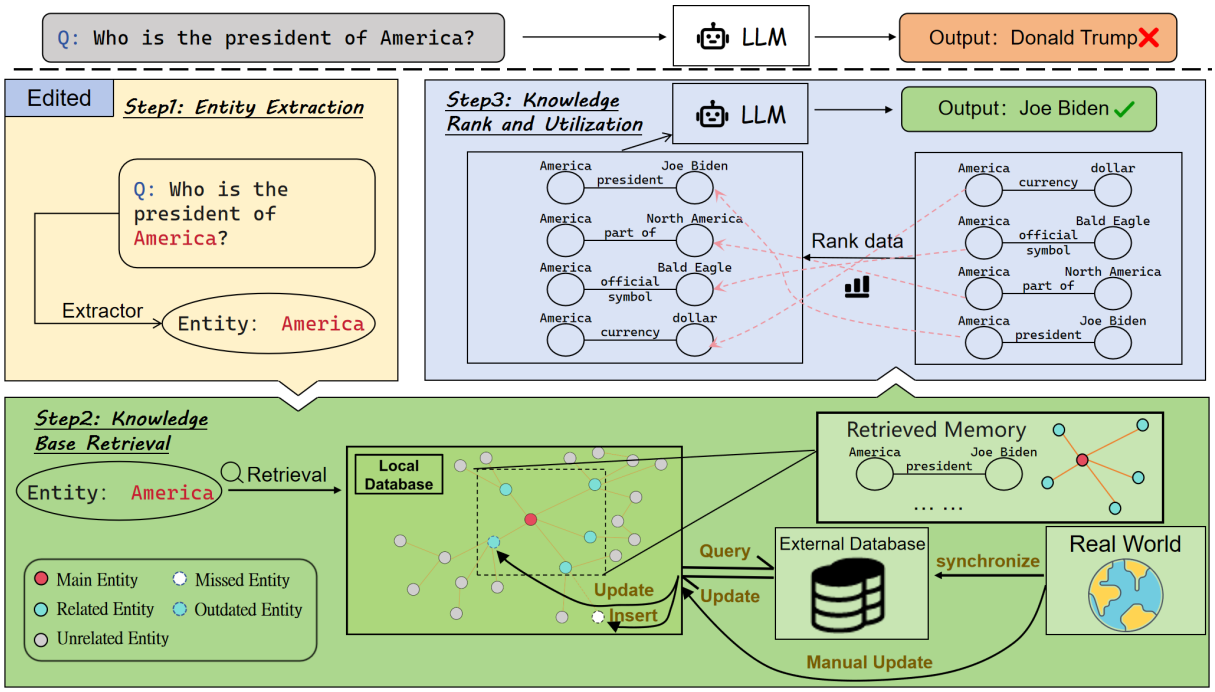

Figure 3: An overview of our method SKEME. SKEME initially extracts key entities from the question. Subsequently, it searches the local knowledge base and ranks applicable knowledge items. Ultimately, it utilizes in-context learning to modify the model's output. Additionally, we update knowledge from external databases and the real world to ensure that the local knowledge base reflects real-world changes.

6 SKEME: Our Model Edit Method

6.1 Overview

The overview of SKEME is shown in Figure 3. SKEME consists of three main components: Entity Extraction is used to discern key entities from the input. This is followed by Knowledge Base Retrieval, which is implemented for the precise retrieval of relevant facts and updating the local database. Lastly, Knowledge Rank and Utilization are applied to employ knowledge, thereby enhancing the outputs of extensive models.

For implementation details, please refer to Appendix A.2.

6.2 Entity Extraction

Due to the vast volume and frequent updates of real-world knowledge, we contend that structured database outperforms vector database. Entity extraction, which aims to extract important entities from the provided input, is employed as the initial stage for the effective utilization of structured data.

6.3 Knowledge Base Retrieval

Handling a large number of rapidly updating facts in the real world poses a challenge for model editing. Structured databases are crucial in tackling these challenges (Zhang et al., 2023). For the first challenge, structured databases facilitate precise searches over a wide range. For the latter, structured databases make it possible to modify knowledge, which is challenging for previous methods.

To follow the swift adapt of real-world changes, inspired by the caching mechanism in operating systems, we propose a method for dynamically constructing a local database. During the process of querying, the local knowledge base is responsible for updating outdated knowledge or incorporating new knowledge from the external database. Although external databases like Wikidata are continuously updated, there might be delays in reflecting real-world changes. Therefore, we also provide manual updates to ensure our knowledge base can better reflect real-world developments.

6.4 Knowledge Rank and Utilization

To achieve a more efficient and precise utilization of facts acquired in the preceding stage, these facts were subsequently ranked based on their relevance to the input sentence.

To address two additional challenges in practical scenarios: how to handle diverse forms of input and how to ensure efficacy across models of varying sizes. Inspired by Zheng et al. (2023), we insert the

facts into the model's input to facilitate in-context learning by the model.

7 Experiments

7.1 Metrics

We use the following metrics to evaluate whether editing has achieved our goal in Section 3.

Accuracy To calculate accuracy, We instruct the model to generate responses for tasks and evaluate whether they match the gold answers exactly. The resulting average accuracy is then recorded as **exact match (EM)**.

Locality Locality measures whether an editing method will influence irrelevant knowledge. We utilize **drawdown (DD)** (Mitchell et al., 2021, 2022) to compute performance degradation and employ **Neighborhood KL divergence (NKL)** (Hoelscher-Obermaier et al., 2023) to measure whether the model is significantly affected.

SURE To comprehensively evaluate and compare the practical effectiveness of methods, we integrate both Accuracy and Locality, and then propose the metric **SURE** (**S**tatistical and **U**nbiased **R**eal-world **E**valuation) to estimate the performance of edited models in real-world scenarios. We define SURE as follows:

$$SURE = ax^{\alpha} - by^{\beta} \tag{2}$$

The x and y represent EM and DD. The parameters a and b denote the ratio of the data used to evaluate the two metrics. The weights α and β are used to characterize the importance of EM and DD, which represent Accuracy and Locality. In our evaluation, we considered an equal amount of data for Accuracy and Locality and treated them as equally important. Therefore, we set the parameters as: $a = b = 1$, $\alpha = \beta = 1$. The task of determining parameters with greater precision is deferred to future research.

Efficiency We aim to find an editing method that is fast and has low memory consumption. Following (Yao et al., 2023), we measure efficiency in both **time consumption (Ti)** and **memory requirements (Me)**.

7.2 Baselines

We compare our method with FT, MEMIT (Meng et al., 2022b), MeLLo (Zhong et al., 2023), and IKE (Zheng et al., 2023). FT is the most classic and straightforward model-editing method. MEMIT is currently considered a state-of-the-art method among parameter modification methods. IKE and MeLLo, much like our approach, leverage a knowledge base and in-context learning. Implementation details can be found in appendix A.3.

7.3 Main Results

Table 2 shows results on FAME. We experiment with all methods on GPT2-XL (Solaiman et al., 2019), GPT-J(6B) (Wang and Komatsuzaki, 2021), Llama2 (Touvron et al., 2023b), and utilized in-context learning based methods on GPT-3.5-turbo (Ouyang et al., 2022).

As articulated in Section 7.1, SURE evaluate the practical effectiveness of different methods comprehensively, thereby emulating the performance of edited model in an actual environment. Firstly, we scrutinize the results on Llama2, which is the largest model achievable and on which all model editing techniques can be employed. **FT** demonstrates a somewhat insignificant enhancement in model performance. **MEMIT**, did not perform as expected in our experiments. We hypothesize that this may be due to the editing process not specifically targeting the model's generative capability. **MeLLo** has a higher EM score, but its DD is also the highest, which indicates pronounced side effects, which leads to a low SURE. Both **IKE** and **SKEME** obtained an EM above 0.9. However, IKE also has presented adverse effects that consequently decreased its SURE. SKEME uniquely maintains a high EM and simultaneously ensures a low DD, thus demonstrating superior practicality compared to other methods.

To test the impact of model size, we experimented with models of various sizes. The results indicate that our method exhibits advantages across all models, whereas some methods fail to work on extremely small models, such as MeLLo on GPT2-XL. These model editing methods also require diverse amounts of time and GPU space. MeLLo, due to its long in-context learning process, consumes the most time. On the other hand, MEMIT had a shorter time consumption but might still be challenging to accept for large-scale data. On the contrary, SKEME proves effective across model sizes while consuming less additional time and GPU space.

It appears that all methods performed poorly on certain tasks. This further validates the meaningfulness of constructing data in various forms.

Model	Method	SURE↑	Accuracy						Locality		Efficiency	
			EM.↑	QA.↑	Com.↑	Clo.↑	Cho.↑	FC.↑	**DD.↓**	**NKL.↓**	**Ti.↓**	**Me.↓**
GPT2-XL	Base	-	19.83	8.00	7.11	3.63	34.25	46.16	-	-	0.18	9.12
	FT	12.75	22.72	11.82	10.26	9.96	33.58	47.96	9.97	1.33	2.12	12.43
	MEMIT	20.87	20.87	7.31	7.14	6.67	34.22	49.04	**0.00**	1.29	13.6	11.85
	MeLLo	−53.67	30.90	71.42	0.24	0.09	33.72	49.01	84.57	1.32	1.43	17.43
	IKE	37.32	50.51	62.05	54.82	48.96	36.09	50.64	13.19	1.25	0.75	14.26
	SKEME	**65.80**	**65.80**	**85.12**	**70.60**	**78.45**	**38.33**	**56.51**	**0.00**	**1.09**	**0.23**	**11.52**
GPT-J	Base	-	23.36	11.86	12.02	11.52	35.34	46.08	-	-	0.35	26.57
	FT	25.21	26.59	13.69	13.38	13.39	40.74	51.72	1.38	1.76	3.27	34.81
	MEMIT	45.85	45.85	49.51	41.14	43.51	**46.62**	48.49	**0.00**	1.86	13.8	29.84
	MeLLo	28.42	55.74	72.20	48.35	72.95	21.81	**63.41**	27.33	1.54	2.42	33.38
	IKE	58.62	70.04	87.00	**82.35**	82.27	46.32	52.26	11.42	**1.28**	0.97	31.53
	SKEME	**73.93**	**73.93**	**97.03**	79.63	**87.02**	46.01	59.97	**0.00**	1.39	**0.49**	**28.17**
Llama2	Base	-	32.20	15.82	15.78	16.02	48.91	64.45	-	-	0.33	30.04
	FT	34.31	41.80	30.05	29.08	29.22	60.57	60.07	7.49	2.55	5.18	38.92
	MEMIT	48.03	48.39	41.16	40.26	41.47	61.00	58.08	0.36	2.83	13.2	33.52
	MeLLo	36.38	66.26	68.56	36.95	69.26	78.17	78.35	29.88	2.72	2.45	38.66
	IKE	71.38	**91.42**	97.72	**90.11**	**95.76**	**95.10**	78.42	20.04	2.48	1.08	35.17
	SKEME	**90.54**	90.54	**98.61**	83.04	90.27	93.73	**87.07**	**0.00**	**2.12**	**0.45**	**31.83**
GPT-3.5-turbo	Base	-	40.11	18.76	19.65	17.17	73.73	71.22	-	*	0.81	*
	MeLLo	56.58	73.75	70.51	57.16	76.37	82.78	81.92	17.16	*	2.92	*
	IKE	76.45	89.53	92.81	**89.72**	**90.88**	90.41	83.85	13.08	*	1.47	*
	SKEME	**91.76**	**91.76**	**98.07**	84.78	89.45	**99.04**	**87.40**	**0.00**	*	**1.03**	*

Table 2: Main result on our dataset. Com. stands for completion. Clo stands for cloze. Cho stands for choose. FC stands for fact-checking. TI(s) includes both editing and generating time in Wall clock time and Me(GB) is calculated by measuring the maximum required GPU VRAM. To maintain brevity, the multiplier of $\times 10^{-4}$ has been excluded for the NKL metric. Since DD and NKL are calculated relative to the unedited model, the unedited model does not have these metrics. $*$: The computation of NKL and ME metrics for GPT-3.5-turbo is impractical due to its utilization via API calls.

On the completion task, although the base model performed similarly to QA and Cloze, the edited model's accuracy was significantly lower than QA and Cloze. We believe that it indicates that the method's generalization performance still needs to improve.

8 Analysis

As we mentioned above, we found that certain methods have already reached a commendable level. However, we cannot ensure how these methods perform in the real world. To simulate the performance of edited models in the real world, we proposed a series of research questions (RQs) as follows.

8.1 RQ1: Whether the Method Can Handle Iterative Editing?

One possible situation is the iterative editing of a particular fact (Xu et al., 2023). For example, if we want LLMs to tell us today's date, it would require the model to change its output every day, making it necessary to edit the model continuously. The results show that even if a fact is edited only

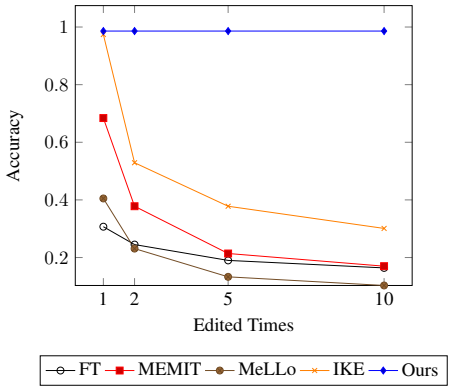

Figure 4: Result of RQ1. We select a varying of answers for each question and edited them into the model in turn. The graph shows the trend of accuracy for methods as the number of edits increases.

twice, the accuracy of other methods has declined significantly. For retrieval-based models, previous methods could not update knowledge, highlighting the necessity of using structured knowledge.

8.2 RQ2: How Many Facts Can We Edit Simultaneously?

To simulate a large number of facts in the real world, we used more triples to test the capabilities of methods.

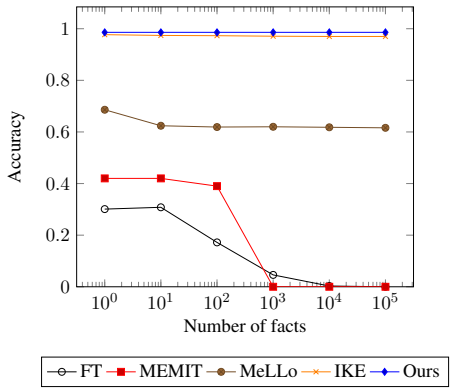

Figure 5: Result of RQ2. We edit diverse quantities of facts at once, the graph shows the trend of accuracy as the number of facts edited changes.

The results indicate that the performance of FT and MEMIT decreases quickly. IKE, MeLLo, and our method all perform well in scenarios with a larger number of facts to be edited.

8.3 RQ3: Whether Methods are Effective in Other Benchmarks?

To comprehensively evaluate model editing methods, we assess these methods on a range of general datasets. See Appendix A.4 for dataset details.

Method	TQA	NQ	FEVER	Vi
Base	0.698	0.191	0.792	0.397
FT	0.362	0.274	0.646	0.228
MEMIT	0.449	0.632	0.724	0.461
MeLLo	0.811	0.633	0.872	0.720
IKE	0.962	**0.980**	0.954	**0.964**
SKEME	**0.984**	0.964	**0.987**	0.956

Table 3: Result of RQ3. TQA stands for triviaQA, NQ stands for Natural Questions, Vi stands for VitaminC. All accuracies are calculate based on exact match rates.

It can be observed that our method consistently improves the model's performance irrespective of the benchmarks, demonstrating the robust versatility and scalability of our approach. Other methods show less stable improvements in model performance.

8.4 RQ4: Can Model Make Further Inference Based on Edited Facts?

When discussing model editing, beyond modifying the model's responses to specific questions, we also aim at modifying the model to make further reasoning based on the edited facts.

Method	MultihupQA			
	k=2	k=3	k=4	k=5
Base	0.145	0.135	0.112	0.079
FT	0.223	0.362	0.231	0.128
MEMIT	0.176	0.247	0.136	0.060
MeLLo	0.270	0.227	0.167	0.073
IKE	0.332	0.237	0.220	0.159
SKEME	**0.960**	**0.786**	**0.427**	**0.167**
Method	**Dialogue**			
	k=2	k=3	k=4	k=5
Base	0.119	0.118	0.116	0.082
FT	0.190	0.216	0.152	0.133
MEMIT	0.238	0.220	0.148	0.126
MeLLo	0.353	0.295	0.193	0.111
IKE	0.229	0.235	0.207	**0.188**
SKEME	**0.946**	**0.757**	**0.390**	0.181

Table 4: Result of RQ4. SKEME manifests significant improvements compared to previous approaches, however, it still fails to address the issue when $k \geq 4$. A conceivable explanation could be the limited inferential capabilities of the model.

Table 4 presents the results for this task. We can observe that all methods, except for SKEME, performed poorly. Traditional retrieval-based models struggle to find answers to multi-hop questions, and other methods do not enable the model to infer based on edited facts.

9 Conclusion

We introduce the practicality requirement for model editing and created a dataset FAME, which embodies practicality with factual data and diverse tasks. We propose a model editing method, SKEME, that proves effective across various models and tasks. The experiments demonstrate that previous model editing methods have difficulties dealing with real-world complexities, while our approach successfully addresses these challenges. We hope that our work will advance the field of model editing and inspire further research in this area.

Limitations

The data in FAME is limited to a monolingual scope, and we did not multilingual data. We posit that the inclusion of multilingual data can further align with the real world, and we leave this as a potential area for future work.

Ethics Statement

We ensure that the collection of FAME is done in a manner consistent with the terms of use stipulated by its sources and the intellectual property rights of the original authors. We make sure that individuals involved in the collection process are treated fairly, including ensuring their voluntary participation and informed consent. Due to the dynamic nature of the real world, certain knowledge contained in FAME may become outdated, rendering it no longer reflective of the latest world conditions.

References

Oshin Agarwal and Ani Nenkova. 2022. Temporal effects on pre-trained models for language processing tasks. *Transactions of the Association for Computational Linguistics*, 10:904–921.

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.

Xin Cheng, Yankai Lin, Xiuying Chen, Dongyan Zhao, and Rui Yan. 2023. Decouple knowledge from paramters for plug-and-play language modeling. *arXiv preprint arXiv:2305.11564*.

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 (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

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*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Anshita Gupta, Debanjan Mondal, Akshay Sheshadri, Wenlong Zhao, Xiang Li, Sarah Wiegreffe, and Niket Tandon. 2023. Editing common sense in transformers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8214–8232.

Jason Hoelscher-Obermaier, Julia Persson, Esben Kran, Ioannis Konstas, and Fazl Barez. 2023. Detecting edit failures in large language models: An improved specificity benchmark. *arXiv preprint arXiv:2305.17553*.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

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 Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

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.

Openai. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin c! robust fact verification with contrastive evidence. *arXiv preprint arXiv:2103.08541*.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. In *International Conference on Learning Representations*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

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.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. 2023a. Easyedit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, et al. 2023b. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Yan Xu, Mahdi Namazifar, Devamanyu Hazarika, Aishwarya Padmakumar, Yang Liu, and Dilek Hakkani-Tür. 2023. Kilm: Knowledge injection into encoder-decoder language models. *arXiv preprint arXiv:2302.09170*.

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. *arXiv preprint arXiv:2305.13172*.

Xunjian Yin, Baizhou Huang, and Xiaojun Wan. 2023. Alcuna: Large language models meet new knowledge. *arXiv preprint arXiv:2310.14820*.

Yichi Zhang, Zhuo Chen, Yin Fang, Lei Cheng, Yanxi Lu, Fangming Li, Wen Zhang, and Huajun Chen. 2023. Knowledgeable preference alignment for llms in domain-specific question answering. *arXiv preprint arXiv:2311.06503*.

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, Dan Friedman, and Danqi Chen. 2021. Factual probing is [mask]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

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*.

A Implementation Details

A.1 Data Filter

Ambiguity issues involve two aspects: different entities sharing the same name and a specific entity's relation corresponding to multiple objects.

For the former scenario, one example is: *Hope Springs* could refer to a movie from 2012 (Q327214 in Wikidata)[2], but can be a movie from 2003 as well (Q596646 in Wikidata)[3]. So when asking *Who is the director of Hope Springs?*, there are multiple correct options.

An example of the latter scenario is: a person may have multiple children, so there are multiple correct answers when asking for their children's names.

We believe that the above two scenarios are simpler compared to questions with only one answer. Therefore, for easier implementation and to focus on more fundamental phenomena, we excluded data in the dataset containing instances of the above situations.

A.2 SKEME Details

A.2.1 Entity extraction

Entity Extraction aims to extract important entities from the provided input, aligning with the subject of the sentence. Previous research has extensively explored methods such as NER or entity linking (Wu et al., 2019). Results indicate that this specific subtask can easily attain an accuracy rate exceeding 97% on our dataset. The accuracy statistics of entity extraction on our dataset are depicted in the table 5.

Method	accuracy
GPT-3.5-turbo	98.1
Llama2	97.3
T5	99.8

Table 5: Accuracy for entity extraction, when using GPT-3.5-turbo and Llama2, we employed few-shot. When using T5, we finetune on FAME items for 5 epochs.

A.2.2 Knowledge Base Retrieval

The local knowledge base is stored in the form of a knowledge graph. When updating the local knowledge base, it can be automatically updated from the external database or manually injected with certain facts to reflect real-world changes. Such updates may require a considerable amount of time, but they can be done in parallel in arbitrary quantities and during idle times. Consequently, we did not explicitly evaluate the duration dedicated to this aspect.

[2] https://www.Wikidata.org/wiki/Q327214

[3] https://www.Wikidata.org/wiki/Q596646

A.2.3 Knowledge Rank and Utilization

Following previous works (Zhong et al., 2023; Zheng et al., 2023), we ranked the retrieved knowledge based on similarity to the input and selected the top-k knowledge. In our experiments, we set $k = 1$. We prompt the model to use the retrieved knowledge for updating its output.

We utilized an off-the-shelf retrieval model (Izacard et al., 2021) to identify and rank the fact triplets, which allows us to avoid the training process.

A.3 Implementation Details for Baselines

For FT, MEMIT, and IKE, we use the framework provided by Wang et al. (2023a).[4].

FT Following previous works (Meng et al., 2022b), We applied Fine-Tuning (FT) to the given layer of the model. For GPT2-XL, we select layer 0, and for GPT-J and Llama2, we choose layer 21.

MEMIT For GPT2-XL and GPT-J, we employed default hyperparameters. For Llama2, we updated the parameters of layers $\{4, 5, 6, 7, 8\}$. Across all models, we calculated covariance statistics using 50,000 instances from Wikitext.

MeLLo The original method was designed for multi-hop questions. We redesigned the prompt for each task while keeping the knowledge retrieval part unchanged.

IKE In the original paper, relevant facts were directly added to the prompt. To make a fair comparison, we removed this part and ensured that all facts were retrieved[5]. Our retrieval settings remained consistent with the original paper.

A.4 Other Benchmarks

To comprehensively evaluate model editing methods, we tested these methods on triviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), FEVER (Thorne et al., 2018) and VitaminC (Schuster et al., 2021). TriviaQA and Natural Questions are commonly employed to assess the capabilities of LLMs (Touvron et al., 2023a). FEVER serves as a classic dataset for fact-checking, and VitaminC has been utilized in prior works to evaluate the effectiveness of model editing (Mitchell et al., 2022).

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

[5] The author's response to the issue: https://github.com/Zce1112zslx/IKE/issues/3

B Problems Definition Details

B.1 Precise Definition

Let (subject, relation, object) be a factual triple, denoted as (s, r, o). Consider an input-output pair as (x, y), where x is effectively a combination of s and r. A model is represented by a function $f : X \to Y$, where X represents the input set and Y represents the output set.

For any t in the set $\{s, r, o, x, y\}$, we use the notation T' to represent all description that is semantically equivalent to t, and t' represents any element within the set T'. Notice that $t \in T'$. Then, we can define $I(x, y)$ as

$$I(x, y) = \{(x', y') | x' \in X' \text{ and } y' \in Y'\}. \quad (3)$$

To define $EX(x, y)$, let's define a fact triple as $tr(s, r, o)$, and S is the set of all fact triples. Also, define the multiplication operation $*$ for two sets of fact triples A and B as the join operation:

$$A * B = A \underset{o=s}{\bowtie} B \quad (4)$$

Then, define

$$N_0(tr) = \{(s', r', o') \mid s' \in S', r' \in R', o' \in O'\} \quad (5)$$

and

$$N_i(tr) = N_{i-1}(tr) * S \qquad (i \geq 1) \quad (6)$$

.

Ultimately, we define $EX(tr)$ as

$$EX(tr) = \bigcup_{i=0}^{\infty} N_i \quad (7)$$

.

By incorporating s and r into the x, we derive the expression $EX(x, y)$.

After defining $I(x, y)$ and $EX(x, y)$, we can define $O(x, y)$ as

$$\complement_S(I \bigcup EX) \quad (8)$$

.

B.2 Example of Definition

Symbol	Example
(x, y)	(Who is the current head of government for America?, Joe Biden)
$I(x, y)$	(The head of government for America is __, Joe Biden)
$EX(x, y)$	(Who is the spouse of the President of the United States?, Jill Biden)
$O(x, y)$	(What color is the Sky?, Blue)