COMEM: In-Context Retrieval-Augmented Mass-Editing Memory in Large Language Models

Anonymous ACL submission

Abstract

Noting that world knowledge continuously evolves over time, large language models (LLMs) need to be properly adjusted by performing the "knowledge editing", such as updating outdated information or correcting false information, etc. Pursuing the reliable "massive" editing ability in terms of *generalization* and *specificity*, this paper proposes a unified knowledge editing method referred to by in-**CO**ntext retrieval-augmented **M**ass-**E**diting **M**emory (COMEM), which combines two types of editing approaches – parameter updating and in-context knowledge editing (IKE). In particular, COMEM includes the *retrieval-augmented IKE*, a novel extension of IKE to a massive editing task based on the *updating*-aware demonstration construction. Experimental results on the zsRE and CounterFact datasets show that the proposed COMEM outperforms all existing methods, leading to state-of-the-art performance. Our code is available at https://github.com/xxxx/xxxx.

1 Introduction

Large language models (LLMs), owing to their stored vast amount of world knowledge, have reported the remarkable abilities of understanding and generating natural languages, as well as achieving the state-of-the-art performances in a wide range of natural language processing (NLP) applications (Touvron et al., 2023; OpenAI, 2023; Petroni et al., 2020). Given the demands of enhancing controllability for LLMs in knowledge manipulation (Onoe et al., 2022; Dhingra et al., 2022; Liška et al., 2022) and content generation (Zhao et al., 2023; Ji et al., 2023; Lazaridou et al., 2021; Agarwal and Nenkova, 2022; Gallegos et al., 2023), there has been recently increasing studies on the "knowledge editing" task, which aims to explicitly provide the "editing" mechanism such as revising knowledge or correcting false information in LLMs in a controllable, scaled, and effective manner. In particular, the paper addresses the "massive" editing task as in (Meng et al., 2022b), because LLMs readily face the issue of the massive edits which requires to update much more than hundreds or thousands of facts, given huge knowledge space.

Approaches for knowledge editing in LLMs have been categorized to two main types: *parameter updating* and *in-context knowledge editing* (IKE). Parameter updating adjusts local parameters or specific layers in LLMs in a gradient-based method to likely generate desired targets given edit requests (Cao et al., 2021; Mitchell et al., 2022a; Meng et al., 2022a,b; Li et al., 2023). In the massive editing task, the advantage of parameter updating has inherited from LLMs; the knowledge is stored implicitly in LLM's parameters and the inference step for the knowledge lookup is simply proceeded in a generative manner based on the decoder, without requiring to maintain an external memory or to search over a set of edits. However, parameter updating may lead to *under-editing* problem, because some edits and their relevant facts are interrupted by other edits thereby being stored in somehow blurred manner. Furthermore, as noted by (Zheng et al., 2023), parameter updating may cause side effects like catastrophic forgetting or over-editing on out-of-scope knowledge.

On the other hand, motivated by the ability of in-context learning (ICL) (Brown et al., 2020; Wei et al., 2023), IKE guides LLMs to generate desired targets in a given context by prepending the edit-related specific prompts which consist of relevant demonstrations. IKE has shown to effectively perform the knowledge editing based on demonstration formatting and organization strategies (Zheng et al., 2023), without modifying the model parameters. Under the setting of the massive editing task, however, IKE may require an additional retrieval step which finds relevant facts given the test query, which is not required in parameter updating. In addition, the performance of IKE largely relies on the

demonstration construction, possibly causing the risky situation when the demonstrations are not optimally desired for some test prompts or contexts.

The goal of knowledge editing is to satisfy both *generalization* and *specificity*, however, there are somehow the trade-off-like relationship between these properties. Pursuing the reliable massive editing ability for more stably satisfying *generalization* and *specificity*, this paper proposes a unified knowledge editing method referred to by in-**CO**ntext retrieval-augmented **M**ass-**E**diting **M**emory (COMEM), which combines parametric updating and IKE, specifically consisting of two components:

- **MEMIT for parameter updating**, which takes a set of massive edits and directly applies MEMIT (Meng et al., 2022b) to update the provided knowledge in LLMs.

- **Retrieval-augmented IKE**, which generates IKE (Zheng et al., 2023) to deal with massive edits, by memorizing all edit requests with their relevant demonstrations from the set of training edits. Unlike the original IKE (Zheng et al., 2023), we further propose the *updating-aware demonstration* construction, motivating from that "copy"-type demonstrations are not much necessary because it is expected that the use of MEMIT somehow exhibits the basic editing capability, thus likely obtaining the proper level of generalization and specificity. By removing copy types, we could add other types of demonstrations, which are shown to be helpful to further improve the final editing performances under the combined setting.

Our contributions are summarized as follows: 1) we propose COMEM, a novel knowledge editing approach that combines parameter updating and IKE to guide the model towards stable generalization and specificity for the massive editing, 2) we extensively apply IKE to the massive editing setting and present the retrieval-augmented IKE, further proposing the updating-aware demonstration which is optimal under COMEM, 3) the proposed COMEM shows state-of-the-art performances on zsRE and CounterFact datasets.

2 Related Work

2.1 Parameter Updating for Knowledge Editing

The most of recent knowledge editing works were applied in parameter rewriting manner, and can be further categorized to two types: hyper-networks-based methods and attribution-based methods.

For hyper-network-based method: Knowledge Editor (Cao et al., 2021) trained a hyper-network that predict the parameter changes during inference, updating the target fact and retaining other unrelated knowledge. MEND (Mitchell et al., 2022a) using the hyper-network to convert the initial fine-tuning gradient into a simplified representation using low-rank decomposition and update the gradient. (Mitchell et al., 2022b) offers a higher-capacity solution by incorporating a semi-parametric editing approach with a retrieval augmented counterfactual model. It stores edits in a separate memory and learns to reason with them to influence the predictions of the base model.

For attribution-based methods: (Dai et al., 2022) explores how LLMs store factual knowledge and introduces the concept of knowledge neurons, and utilizing knowledge neurons for precise factual knowledge editing (updates, erasures) without resorting to fine-tuning. ROME (Meng et al., 2022a) is a pioneering that try to locating the model parameter associated with target factual knowledge, and rewriting the key-value pairs in MLP module with computed new vectors. However, all of above methods suffer from significant efficacy and generalization deterioration when increasing the required editing volume. MEMIT (Meng et al., 2022b) further improves ROME to enable massive knowledge editing by spreading the weight changes over multiple model layers.

2.2 In-Context Learning for Knowledge Editing

In-Context Learning (Dong et al., 2022) is a technique that emerged with the advent of LLMs, where the model learns by observing and incorporating information from the context (Liu et al., 2022; Brown et al., 2020). It involves temporarily adapting or updating a model's parameters based on the provided prompts or demonstrations (Lu et al., 2022; Rubin et al., 2022) in a run, leading to an improvement in model performance.

(Si et al., 2023) pioneered the exploration of using In-Context Learning to update knowledge

in LLMs. They demonstrated that incorporating various types of demonstrations notably enhances the success rate of knowledge editing. IKE (Zheng et al., 2023) further extended ICL-based knowledge editing in different language model with fewer side effects.

However, both parametric methods and In-Context Learning based methods failed to achieve a significant improvement in the performance of massive knowledge editing task, we integrate both paradigm and make further augmentation to elevate the upper limit of current performance in massive knowledge editing.

3 Task Definition

Suppose that $\mathcal{S}$ is a set of real-world entities or concepts, $\mathcal{M}_\theta$ is an autoregressive language model with the parameter θ and $\mathcal{E} = \{e_i\}_{i=1}^N$ a set of new facts to be injected to $\mathcal{M}_\theta$, where $e_i = (s_i, r_i, o_i)$ is the i-th edit, i.e. a triple that consists of a subject $s_i \in \mathcal{S}$, a relation r_i, an object $o_i \in \mathcal{S}$. For notational convenience, $\mathcal{M}_\theta(x)$ is the generated result given the input prompt x under the language model $\mathcal{M}_\theta$ is defined as follows:

$$\mathcal{M}_\theta(x) = \underset{y \in \mathcal{S}}{\operatorname{argmax}} P_{\mathcal{M}_\theta}(y|x) \tag{1}$$

where $P_{\mathcal{M}_\theta}(y|x)$ is the generative probability of y, given a prefix x.

The goal of the massive knowledge editing is to satisfy efficacy, generalization, and specificity, for "all" edits in $\mathcal{E}$. Formally, for $e_i \in \mathcal{E}$, let $\mathcal{I}(e_i)$ be the *edit scope* of e_i, the set of *in-scope* examples, and $\mathcal{O}(e_i) = \mathcal{U} - \mathcal{I}(e_i)$ be the set of *out-of-scope* examples where

$$\mathcal{U}$$

is a universal set of knowledge [1]. For example, e_i is "*Fox News was created in Canada*," an in-scope example in $\mathcal{I}(e_i)$ is "*Fox Soccer News originated in Canada*," and an out-of-scope example in $\mathcal{O}(e_i)$ is "*iOS 6 was created by Apple.*" Efficacy, generalization, and specificity are defined as follows:

- **Efficacy**, which is satisfied for i-th edit if $o_i = \mathcal{M}_\theta(x_i)$ where x_i is the prefix prompt, roughly defined as $[s_i, r_i]$ for i-th edit [2] .
- **Generalization**, which is satisfied for i-th edit if $o = \mathcal{M}_\theta(x)$ for all in-scope examples $(s, r, o) \in \mathcal{I}(e_i)$ and $x = [s; r]$.
- **Specificity**, which is satisfied for i-th edit if $o = \mathcal{M}_\theta(x)$ for all out-of-scope examples $(s, r, o) \in \mathcal{O}(e_i)$ and $x = [s; r]$.

4 Method

Figure 1 presents the overall structure of our proposed COMEM to inject a set of edits $\mathcal{E}$ to the language model $\mathcal{M}_\theta$, which combines parameter updating method and IKE. Formally, suppose that $\mathcal{T} = \left\{e_j^t\right\}_{j=1}^M$ is a set of "training" edits in a training set, and each training edit e_j^t is pre-associated with $\mathcal{D}^t(e_j^t) = (\mathcal{D}^c(e_j^t), \mathcal{D}^u(e_j^t), \mathcal{D}^r(e_j^t))$ a set of demonstrations of three types including *copy*, *update*, and *retain*, denoted as $\mathcal{D}^c(e_j^t)$, $\mathcal{D}^u(e_j^t)$, and $\mathcal{D}^r(e_j^t)$, respectively [3]. COMEM consists of editing (i.e. training) and inference steps as follows:

- **Editing step**: Given a set of requested edits $\mathcal{E}$, COMEM performs the editing step:

$$\begin{aligned} \mathcal{M}_{\theta^*} &= \mathsf{PU}\left(\mathcal{E}, \mathcal{M}_\theta\right) \\ e_1' \cdots e_k' &= \mathsf{NeighborEdits}\left(e_i, \mathcal{T}\right) \\ \mathcal{D}(e_i) &= \mathsf{ConstructDemo}\left(\mathcal{D}^t(e_j')_{j=1}^k\right) \end{aligned} \tag{2}$$

  where PU is the parameter updating method of knowledge editing that injects a set of edits $\mathcal{E}$ to the language model $\mathcal{M}_\theta$ and returns the language model with the updated parameter θ^*, NeighborEdits is the retrieval function that returns the top-k training edits that are the most similar to the given requested edit e_i, and ConstructDemo is the demonstration construction component that selects a subset of the demonstrations in the top-k training edits, based on the *updating*-aware selection criteria.

- **Inference step**: Given a testing prompt of $x = [s; r]$, COMEM performs the inference step:

$$\begin{aligned} \mathcal{D}(x) &= \mathsf{GetDemo}(x, \mathcal{D}(e_i)_{i=1}^N) \\ y &= \mathcal{M}_{\theta^*}([\mathsf{prompt}\left(\mathcal{D}(x)\right); x]) \end{aligned} \tag{3}$$

  where $\mathcal{D}(e_i)_{i=1}^N$ is a pre-constructed set of demonstrations corresponding to $\mathcal{E}$, and

[1] The terminologies related to the edit scope are based on those in (Mitchell et al., 2022b; Zheng et al., 2023)

[2] Here, $[s_i, r_i]$ refers to a natural language format that consists of s_i and r_i.

[3] Here, the demonstration types of copy, update and retrain correspond to the "requested", "paraphrased", and "neighborhood" prompts in the dataset, respectively.

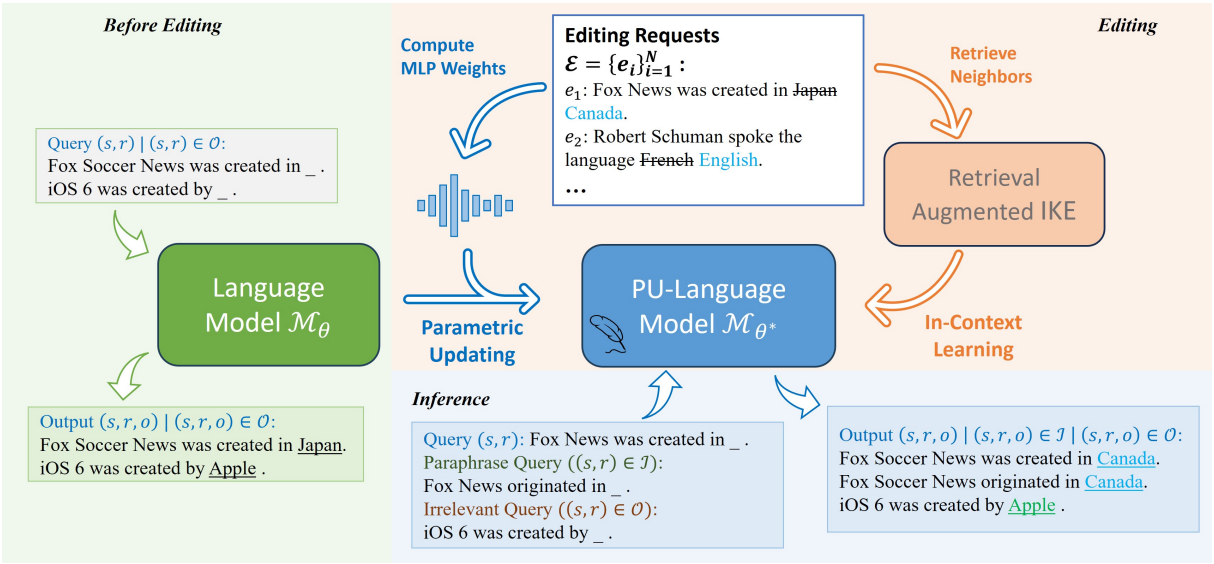

Figure 1: An overall illustration of COMEM: Given a set of massive edits $\mathcal{E}$ parameter-updated (PU) language model is first obtained by using MEMIT (in Section 4.1), and the retrieval-augmented IKE is subsequentially performed (in Section 4.2) to combine the effects of parameter updating and IKE in a complementary manner. During the editing step, the retrieval-augmented IKE constructs updating-aware demonstrations consisting of only update and remain types, based on a set of neighbors in the training edits of each requested edit e_i. During inference step, the test query (s, r) is given, COMEM retrieves the requested edit stored during the editing step by matching with $(s, r$, obtain its associated demonstrations, which are concatenated with the test prompt of (s, r) being fed into $\mathcal{M}^*$, which finally predicts the target objects as required in $\mathcal{E}$, while retaining other non-edited knowledge (i.e., *"iOS 6 was created by Apple"*).

GetDemo returns a set of online few-shot demonstrations for IKE, prompt $(\mathcal{D}(x))$ is the prompting function that linearizes the selected few-shot demonstrations $\mathcal{D}(x)$ via a proper prompting template, $[\text{prompt}\,(\mathcal{D}(x))\,; x]$ is the concatenated prompt that consists of the demonstrations and the testing prompt x, and y is the predicted object returned by COMEM given x.

4.1 Parameter Updating Method: MEMIT

For PU, the parameter updating method, we employ MEMIT proposed by (Meng et al., 2022b), which involves rewriting local model parameters across a range of layers. The detailed description of MEMIT is presented in Appendix A.

4.2 Retrieval Augmented In-Context Knowledge Editing

Given the parametric updated model $\mathcal{M}_{\theta*}$, we perform the retrieval-augmented IKE, which consists of NeighborEdits and ConstructDemo for the editing step, and GetDemo for the inference step.

4.2.1 Updating-aware Demonstration Construction

Unlike IKE that use 32 demonstrations for *"copy"*, *"update"*, and *"retain"* with a ratio of $1:3:4$ (an example shown in Appendix H), we propose the updating-aware demonstration construction for the ConstructDemo, given our COMEM setting where IKE is subsequentially applied on the parameter-updated language model $\mathcal{M}_{\theta*}$, not being used solely as a single editing method.

The underlying assumption is that once parameter updating is applied, $\mathcal{M}_{\theta*}$ is likely equipped with a proper level of editing capabilities, in terms of efficacy, generalization, and specificity. When applying IKE on $\mathcal{M}_{\theta*}$, the language model is updated to somehow handle properly in-scope and out-of-scope examples, unlike the original setting of IKE in (Zheng et al., 2023) based on the fully unedited status. In the preliminary experiment, we found that the use of copy-type demonstrations was not helpful to improve the editing capabilities under COMET setting. Because the effect of ICL is limited by the maximum input length of the language model, we would like construct more impactable demonstrations in a way of adding non-copy-type demonstrations from more training edits which are

similar to the current given edit.

The updating-aware demonstration construction consists of NeighborEdits and ConstructDemo.

Dense Retrieval for Finding Neighbor Edits We use the dense retrieval for NeighborEdits based on the cosine similarity between the training edit e_j^T and the given requested edit e_i. More precisely, suppose that $\mathcal{M}_{sent}$ is an additional sentence encoder, where $\mathcal{M}_{sent}(s) \in \mathbb{R}^d$ is the sentence vector for a given sentence s. For notational convenience, given an edit $e = (s, r, o)$, $\mathcal{M}_{sent}(e) = \mathcal{M}_{sent}([s; r; o])$ where $[s; r; o]$ is the natural language format that concatenates s, r, and o using a proper verbalizing template. The similarity between $e = (s, r, o)$ and $e' = (s, r, o)$ is defined as follows:

$$sim(e, e') = cos(\mathcal{M}_{sent}(e), \mathcal{M}_{sent}(e')) \quad (4)$$

For a given edit $e_i \in \mathcal{E}$, $\textsf{NeighborEdits}\,(e_i, \mathcal{T})$ is defined as follows:

$$top\text{-}k\left\{(e_j^t, sim(e_i, e_j^t))\right\}_{j=1}^{M} \quad (5)$$

where $top\text{-}k$ is the operator for selecting top-k elements given the set of pairs of objects and their associated similarities. For $\mathcal{M}_{sent}$, we deploy the pretrained sentence encoder (Reimers and Gurevych, 2019)

Constructing Demonstration of Update and Retain Types Once we have $\{e'_1 \cdots e'_k\} \in \mathcal{T}$ using NeighborEdits, $\textsf{ConstructDemo}\left(\mathcal{D}^t(e'_j)_{j=1}^k\right)$ construct a set of demonstrations by selecting m update-type and n remain-type demonstrations in $\mathcal{D}^u(e'_j)$ and $\mathcal{D}^r(e'_j)$, respectively for e'_j. As a result, we have $k(m+n)$ demonstrations for each requested edit e_i, and $N \times k(m+n)$ demonstrations in $\mathcal{D}$ *in total* for the massive edit request in $\mathcal{E}$.

4.2.2 Retrieval-augmented Inference Step

Retrieval-augmented Inference Step Given a test prompting $q = (s, r)$, we match the subject and relation part in $\mathcal{E}$ in the dataset and obtain $e_q = (s, r, o) \in \mathcal{E}$. GetDemo returns the set of the associated $k(m+n)$ demonstrations for e_q, defined as follows:

$$\textsf{GetDemo}(x, \mathcal{D}(e_i)_{i=1}^N) = \mathcal{D}(e_q) \quad (6)$$

The resulting demonstrations are further concatenated with the test prompt q for finally predicting an output by COMEM.

5 Experiments

5.1 Dataset and Metrics

We first evaluated COMEM on **Zero-Shot Relation Extraction** (zsRE, Levy et al. (2017)) dataset with 10,000 knowledge edits following (Cao et al., 2021; Mitchell et al., 2022a; Meng et al., 2022b). After process, each evaluate sample has one factual statement and its paraphrase, and one natural question that irrelevant to the factual statement, see example in Appendix G.

In this dataset, the metric **Efficacy** measures the editing accuracy:

$$\mathbb{E}[o^* = \text{argmax}\mathcal{P}_{\mathcal{M}^*}((s, r))]. \quad (7)$$

Paraphrase measures the same accuracy on paraphrase prompt:

$$\mathbb{E}[o^* = \text{argmax}\mathcal{P}_{\mathcal{M}^*}((s, r))], \quad (8)$$

where $p(\cdot)$ denote the paraphrase of prompt. And **Specificity** is the model's maximum probability accuracy on unrelated questions that should not be edited:

$$\mathbb{E}[o = \text{argmax}\mathcal{P}_{\mathcal{M}^*}((s, r))], \quad (9)$$

where $u(\cdot)$ denote the editing-irrelevant statement. The **Score** is the harmonic mean of above three metrics that reflects the integrated performance of the model.

We also test our method on **CounterFact** dataset (Meng et al., 2022a) following (Meng et al., 2022a,b; Zheng et al., 2023), which contains 21,919 samples, each sample contains a factual statements, 2 paraphrase of the statements and 10 neighbor prompts that irrelevant to the fact, detailed format can be seen in Appendix G. Similar to the three aforementioned metrics (Efficacy, Paraphrase and Specificity) on zsRE, the **Efficacy Score (ES)**, **Paraphrase Score (PS)**, and **Neighborhood Score (NS)** will be computed to represent the accuracy terms. We also report their mean difference (magnitude) terms: **Efficacy Magnitude (EM)**, **Paraphrase Magnitude (PM)**, and **Neighborhood Magnitude (NM)** that measure the *significance* of editing, detailed definition can be seen in Appendix E. And the aggregated **Score (S)** is the harmonic mean of **ES**, **PS**, and **NS**.

Our implementation details are provided in Appendix D.

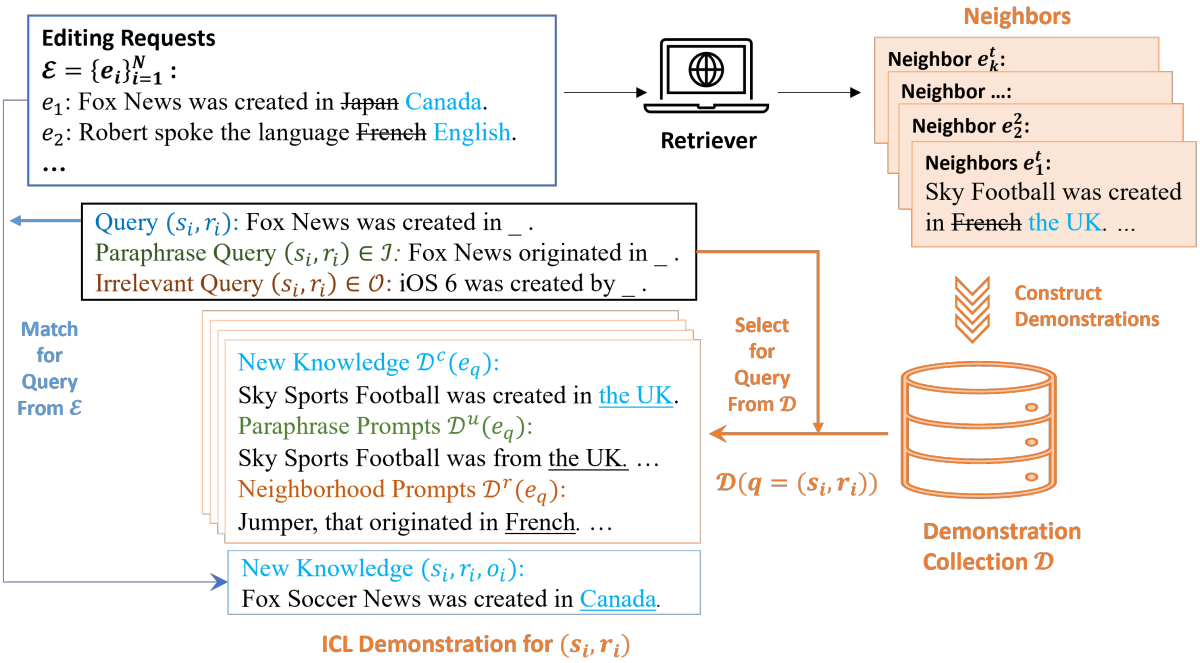

Figure 2: An illustration of retrieval augmented IKE to construct updating-aware demonstration. Given the requested edit e_i, the dense retrieval is first performed to find the top-k neighbor edits in the training sets, which are most similar to e_i. Then, the demonstrations of m update and n retrain-types for each neighbor edit are selected to create $k(m+n)$ demonstrations for $\mathcal{D}(e_i)$. During the inference step, a new query (s, r) is provided and the retrieval is performed by selecting $e_q = (s, r, o)$ where the subject and relation elements are matched. $\mathcal{D}(e_q)$ are finally provided as online demonstrations for a query (s, r).

5.2 Baselines

We choose the GPT-J (6B) model (Wang and Komatsuzaki, 2021) which wildly used by related works as backbone and compare COMEM with existing knowledge-editing works:

- **FT** The naive GPT-J model fine-tuned on the edit facts, using early stop to prevent overfitting and weight decay to prevent forgetfulness following (Meng et al., 2022b).
- **MEND** (Mitchell et al., 2022a), a learning based method that predict weight changes using hyper-networks.
- **ROME** (Meng et al., 2022a), a direct parametric updating method that rewrite key-value pairs in MLP layers, it edit single knowledge at a time, and need to perform iteratively for multiple editing.
- **MEMIT** (Meng et al., 2022b), parametric updating method that can edits massive knowledge at one time, it can scale up to thousands of knowledge edits for GPT-J (6B) or larger models.
- **IKE** (Zheng et al., 2023), a pure In-Context Learning based method that use three kinds of designed demonstrations (*"copy"*, *"update"* and *"retain"*) as prompt to steer the language models prediction.
- **PMET** (Li et al., 2023) is an optimized parametric multiple knowledge editing work that simultaneously optimizes the hidden states of Multi-Head Self Attention (MHSA) and Feed-Forward Network (FFN) layers and precisely update the FFN weights.

6 Results

In this section, we present experimental results for massive knowledge editing task on zsRE (Levy et al., 2017) and CounterFact (Meng et al., 2022a) datasets, comparing them with recently proposed baselines. Additionally, we conduct discussions and analyses based on ablation studies.

6.1 Main Results

Results on zsRE. We compared the *Efficacy*, *Paraphrase*, and *Specificity* metrics with baselines on zsRE, the results are listed in Table 1. COMEM

Method	Score ↑	Efficacy ↑	Paraphrase ↑	Specificity ↑
GPT-J	26.4	26.4	25.8	27.0
FT	42.1	69.6	64.8	24.1
MEND	20.0	19.4	18.6	22.4
ROME	2.6	21.0	19.6	0.9
MEMIT	50.7	96.7	89.7	26.6
IKE	35.3	**100**	**100**	15.4
PMET	<u>51.0</u>	<u>96.9</u>	<u>90.6</u>	<u>26.7</u>
COMEM	**61.7**	**100**	**100**	**34.9**

Table 1: Performance comparison of GPT-J (6B) with 10,000 knowledge edits on zsRE dataset. Column-wise best are in bold, second best are underlined.

achieved best results on all metrics and showed a significant boost in the aggregated *Score* (harmonic mean of *Efficacy*, *Paraphrase* and *Specificity*). MEND (Mitchell et al., 2022a) and ROME (Meng et al., 2022a) are baselines that are incapable of massive knowledge editing, consequently, they performed even worse than fine-tuning. Massive knowledge editing method such as MEMIT (Meng et al., 2022b) and PMET (Li et al., 2023) can provide strong editing efficacy with good generalization (in *Paraphrase*) and *Specificity*, while leaving room for further improvements. The pure In-Context Learning method IKE (Zheng et al., 2023) can also achieve the best scores for *Efficacy* and *Paraphrase*, but it does not exhibit ideal *Specificity* in such massive editing context.

Results on CounterFact Table 2 shows the comparison results on CounterFact. We report the accuracy terms **ES, PS, NS** and the magnitude terms **EM, PM, NM** on this dataset, and the *Score* **S** is the harmonic mean of accuracy terms. It can be seen that our proposed COMEM can achieve the best overall performance. Similar to the results on zsRE, MEND (Mitchell et al., 2022a) and ROME (Meng et al., 2022a) losing *Efficacy* and *Generalization* on massive knowledge editing, while MEND achieved best Neighborhood Score. Interestingly, fine-tuned model performs well on *Efficacy* and *Generalization*, but also deteriorated severely in *Specificity*. There are no significant difference between parametric massive knowledge updating methods (Meng et al., 2022b; Li et al., 2023) and IKE (Zheng et al., 2023) in terms of *Efficacy*, including our method. However, there are still considerable gaps in *Generalization* for parametric methods when compared with IKE and COMEM. For IKE, it achieved high *Efficacy* and *Generalization* performance, while underwhelming in *Specificity*.

From the above main results, intensive knowledge editing task beyond the capability of methods that designed for single or few knowledge editing (Mitchell et al., 2022a; Meng et al., 2022a). For models that are capable for massive editing, parametric methods (Meng et al., 2022b; Li et al., 2023) performed well while still leaving room for further optimization. Pure In-Context Learning method exhibits a drop in *Specificity* with massive editing compared to fewer edits as it achieves better *Neighborhood Score* on 2,000 edits test (77.0 on original CounterFact in IKE's (Zheng et al., 2023) paper and 67.6 on filtered CounterFact[4] in our test). Besides the significant impact brought by unfiltered conflicting samples, this drop is primarily caused by the shrink of the retrieval corpus size as more data samples are allocated to the test set, resulting in smaller retrieval searching space, and In-Context Learning method in this task heavily relies on the quality of demonstrations constructed from the retrieved neighbors.

COMEM leverages both the strong foundational efficacy of parametric updating and the augmentation capabilities of In-Context Learning, achieving state-of-the-art performance in massive knowledge editing. We show some output examples in Appendix I.

6.2 Ablation on zsRE

We conducted ablation experiments on zsRE to demonstrate the necessity of using *Parametric Updating* in advance and *In-Context Learning* augmentation, as well as the impact of using different numbers of neighbors (i.e., the quantity of demon-

[4]We use the CounterFact dataset which filtered to remove the samples that violate multiple knowledge editing paradigm as described in **Section 4.3**, the filtered dataset is also referred to as multi-CounterFact.

Method	Score	Efficacy		Generalization		Specificity	
	S ↑	ES ↑	EM ↑	PS ↑	PM ↑	NS ↑	NM ↑
GPT-J	20.47	14.66	-7.40	15.06	-7.50	83.97	7.65
FT	63.54	<u>99.91</u>	**98.24**	88.14	48.65	38.67	-8.22
MEND	25.23	17.61	-12.19	20.10	-11.34	**80.83**	12.55
ROME	49.92	49.36	-0.03	49.51	-0.09	50.92	0.09
MEMIT	85.71	99.10	87.85	88.33	38.02	<u>73.59</u>	4.64
IKE	84.88	**99.98**	92.86	<u>96.29</u>	<u>67.37</u>	66.88	<u>25.19</u>
PMET	<u>86.20</u>	99.50	-	92.80	-	71.40	-
COMEM	**88.09**	99.87	<u>94.88</u>	**96.42**	**71.00**	73.14	**35.87**

Table 2: Performance comparison of GPT-J (6B) with 10,000 knowledge edits on CounterFact dataset. Column-wise best are in bold, second best are underlined.

strations) on performance.

It can be seen from Table 3 that when without using parametric updating, the model struggles to achieve optimal *Generalization* and exhibits poor *Specificity*. On the other hand, solely relying on parametric updating (i.e., without ICL) leads to an overall performance decline, particularly with a notable deterioration in *Generalization*.

Increasing the number of nearest neighbors to construct ICL demonstrations can enhance performance, although *Efficacy* and *Generalization* reach their optimum with fewer demonstrations ($k = 8$).

Method	S ↑	ES ↑	PS ↑	NS ↑
- w/o PU	46.8	100	99.9	22.7
- w/o ICL	50.7	96.7	89.7	26.6
- $k = 8$	57.6	100	100	31.2
- $k = 16$	59.7	100	100	33.1
- $k = 24$	61.7	100	100	34.9

Table 3: Ablation study on zsRE. k denotes the number of retrieved nearest neighbors. w/o PU and w/o ICL denote *without Parametric Updating* and *without In-Context Learning* respectively, and w/o PE was conducted under the setting of $k = 24$. **ES**, **PS**, and **NS** reflect the model's *Efficacy*, *Generalization*, and *Specificity*.

6.3 Ablation on CounterFact

The ablation study on CounterFact in Table 4 shows that both solely parametric updating and solely In-Context Learning Editing can achieve good *Specificity*, but suffer from a significant drop in *Generalization*. More precisely, parametric updating can provide slightly stronger *Efficacy* and *Generalization* than pure In-Context Learning.

There is a difference between the results on CounterFact and zsRE that solely use In-Context Learning can achieve best **Neighborhood Score** on CounterFact, mainly because CounterFact dataset can provide us more *neighborhood prompts* to strength *Specificity*.

Integrate parametric updating with In-Context Learning significantly enhances *Generalization* and slightly strengthens *Efficacy*, but leads to a loss in *Specificity*. However, increasing the number of demonstrations can help regain *Specificity* without compromising other aspects.

Method	S ↑	ES ↑	PS ↑	NS ↑
- w/o PU	85.29	99.60	85.64	74.31
- w/o ICL	85.71	99.10	88.33	73.59
- $k = 3$	85.72	99.95	96.43	68.39
- $k = 4$	87.08	99.93	96.43	71.05
- $k = 5$	88.09	99.87	96.42	73.14

Table 4: Ablation study on CounterFact. Similar to on the zsRE dataset, k is the number of nearest neighbors used for ICL demonstration construction, the test of without introducing parametric updating (w/o PE) was conducted under the setting of $k = 5$.

7 Conclusion

In this paper, we proposed COMEM, the unified framework of parameter updating and IKE for massive knowledge editing task. Extensive experiments on zsRE and CounterFact datasets showed that COMEM leaded to the state-of-the-art overall performances, outperforming most existing knowledge editing methods.

In future work, we would like to explore how to parameterize the in-context learning demonstrations into the language model, to avoid the inference efficiency decrease caused by lengthy input prompts, ultimately striving for more concise and efficient knowledge editing.

Limitations

Our work optimizes based on In-Context Learning after parametric rewriting, yet ICL cannot achieve permanent or long-term model knowledge updates. This means that currently the optimized part cannot avoid lengthy demonstration inputs, and concatenating such demonstrations every time the model restarts is inefficient. Therefore, achieving permanent or long-term optimal knowledge editing performance requires exploring methods to parameterize the ICL demonstrations. This would also allow the final model to operate without lengthy input prompts, thereby enhancing inference efficiency, which is one of our future directions. Additionally, current models primarily operate on data samples in tuple form like $(subject, relation, object)$, whereas real-world natural language comes in more diverse and complex forms. Exploring weather the current work can generalize to universal text formats is also an important future task.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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*, volume 33, pages 1877–1901. Curran Associates, Inc.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models.

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.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2023. Bias and fairness in large language models: A survey.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. 2021. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023. Pmet: Precise model editing in a transformer. *arXiv preprint arXiv:2308.08742*.

Adam Liška, Tomáš Kočiský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. *arXiv preprint arXiv:2205.11388*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

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

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. 2022a. Fast model editing at scale. In *International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

Yasumasa Onoe, Michael Zhang, Eunsol Choi, and Greg Durrett. 2022. Entity cloze by date: What LMs know about unseen entities. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 693–702, Seattle, United States. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. *arXiv preprint arXiv:2005.04611*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2023. Prompting gpt-3 to be reliable. In *International Conference on Learning Representations (ICLR)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

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*.

A Detailed Process of MEMIT

Note the MLP weights in a Transformer (Vaswani et al., 2017) as W that can be operated as a key-value store, where $WK \approx V$, $K = [k_1|k_2|...]$ and $V = [v_1|v_2|...]$. Given requested edits $mathcalE = \{(s_i, r_i, o_i)\}$, language model $\mathcal{M}_\theta$, layers to edit $\mathcal{L} = \{L_1, L_2, ..., L_l\}$, and pre-cached covariance constant C^L of k computed from Wikipedia samples (Meng et al., 2022a). For each $(s_i, r_i, o_i) \in \mathcal{E}$, a target vector z_i will be computed:

$$z_i \leftarrow h_i^{L_l} + \delta_i, \tag{10}$$

where δ_i is optimized by:

$$\delta_i \leftarrow \arg\min_{\delta_i} \frac{1}{P} \sum_{j=1}^{P} \xi_i$$

$$\xi_i = -\log \mathbb{P}_{\mathcal{M}(h_i^{L_l} += \delta_i)}[o_i | x_j \oplus (s_i, r_i)] \tag{11}$$

Then for each editing layer $L \in \mathcal{L}$, the hidden state is updated by:

$$h_i^L \leftarrow h_i^{L-1} + a_i^L + m_i^L \tag{12}$$

where a and m denote the *"attention"* and *"MLP"* contributions computed from previous layers in Transformer (Vaswani et al., 2017) model. On the current layer, for each $(s_i, r_i, o_i) \in \mathcal{E}$, the MLP key updated as:

$$k_i^L \leftarrow k_i^L = \frac{1}{P} \sum_{j=1}^{P} k(x_j + s_i) \tag{13}$$

where x_j are random prefixes that aid generalization across contexts. The distributed residual ϕ over remaining layers is computed as:

$$\phi_i^L \leftarrow \frac{z_i - h_i^{L_l}}{l - idx(L) + 1} \tag{14}$$

where $idx(L)$ denote the number index of L. Thus in this layer $k^L = \{k_i^L\}$ and $\phi^L = \{\phi_i^L\}$.

To update the MLP weights in the editing layers, for each layer $L \in \mathcal{L}$, the adding weight is computed as:

$$\Delta^L \leftarrow \phi^L k^{L\top}(C^L + k^L k^{L\top})^{-1}, \tag{15}$$

finally in current layer L the MLP weights updated as:

$$W^L \leftarrow W^L + \Delta^L, \tag{16}$$

after the above updating performed on all the editing layers, we can obtain the parametric updated model $\mathcal{M}_{\theta^*}$.

B Demonstration Analysis for Parametric Updated Model

We demonstrate that a parametric updated model does not need additional *"Efficacy Demonstrations"* but requires more for *Specificity* in In-Context Learning stage.

In IKE's work (Zheng et al., 2023), three kinds of demonstrations (*"copy"*, *"update"*, *"retain"*) were designed for In-Context Learning knowledge editing, where *"update"* demonstrations contribute to *Generalization* and *"retain"* demonstrations improve *Specificity*. Removing *"copy"* demonstrations also lead to performance degradation as there was a drop in *Specificity*. We tested weather the *"copy"* demonstrations still have significance for the parametric updated models, as strong *Efficacy* and *Specificity* have been pre-provided by parameter rewriting.

We keep the number of total demonstrations fixed and redistribute *"copy"* to *"retain"* demonstrations, since IKE losing some *Specificity* when expanding the number of edits from 2,000 to 10,000. Table 5 demonstrates that this redistribution improved the **Neighborhood Score** without compromising *Efficacy* and *Generalization*, leading to an overall promotion in performance.

C Demonstration Analysis on Query Prompt

Table 6 indicates that pre-appending the new knowledge demonstration before the query for the parametric updated model resulted in a significant drop in *Specificity*, as the model was given prompts that excessively biased its predictions towards new facts.

C/U/R	S ↑	ES ↑	PS ↑	NS ↑
4/12/16	83.53	99.99	98.64	63.43
2/12/18	84.25	99.99	98.51	64.70
0/12/20	84.80	99.98	98.53	65.70

Table 5: Various demonstration distributions applied for parametric updated model. The demonstration format used in this test is adopted from IKE, where **C**, **U**, and **R** denote the number of *"copy"*, *"update"*, and *"retain"* demonstrations.

Method	S ↑	ES ↑	PS ↑	NS ↑
zsRE				
- $w/$ Pre	56.3	100	100	30.0
- w/o Pre	61.7	100	100	34.9
CounterFact				
- $w/$ Pre	86.28	99.28	97.03	69.48
- w/o Pre	88.09	99.87	96.42	73.14

Table 6: Comparison of pre-appending the new knowledge demonstration before the query prompt or not.

D Implementation Details

We use GPT-J (6B) (Wang and Komatsuzaki, 2021) as backbone language model to enabling maximum number of comparable cases with related works.

For zsRE (Levy et al., 2017) dataset, we extract 10,000 samples as test set to perform massive knowledge editing, and use *sentence-transformer* toolkit to retrieve k nearest neighbors from the rest set (172,282 samples) of data. Our best result was tested on $k = 24$ setting, resulting in 48 demonstrations for each test sample, wherein each neighbor provides one paraphrase prompt ($m = 1$) and one irrelevant prompt ($n = 1$) used for demonstration construction. In our experiments, larger k values will result in the input sequence length of most samples exceeding the maximum input length of GPT-J (6B). We first run parametric updating on the test set following (Meng et al., 2022b), then using the edited model to perform In-context Learning.

We tested IKE (Zheng et al., 2023) on zsRE with the same demonstration setting as in their paper: retrieve top 32 nearest neighbors and assign the usage of *factual statement*, *paraphrase prompt*, *neighborhood prompt* for *"copy"*, *"update"*, *"retain"* demonstration with the ratio of 1:3:4. Other

baselines were tested by previous works on this dataset, and we adopted the statistic from their paper (Meng et al., 2022b; Li et al., 2023).

For CounterFact (Meng et al., 2022a) dataset, the original dataset contains 21,919 samples, but some of the samples may entail the same prefix (s, r) editing to different new facts, which conflicting with multiple knowledge editing, therefore need to be filtered out. We use the filtered set following (Meng et al., 2022b) that contains 20,877 in total. Given that each data sample in this dataset has 2 paraphrase prompts ($m = 2$) and 10 neighborhood (irrelevant) prompts ($n = 10$), the In-Context Learning prompt consists of $k * 12$ demonstrations. Hence, in our optimal setting, the number of demonstrations for each test sample is 60.

To get the precise results and the *Magnitude* term of baselines, we rerun IKE (Zheng et al., 2023) on the filtered dataset for 10,000 samples under the same setting in their paper, and retested other baselines based on (Meng et al., 2022b)'s repository. But for PMET (Li et al., 2023), we failed to reproduce the experiment due to GPU limitation, thus we directly adopted their results in the paper.

We also observed that pre-appending new knowledge demonstration before query sequence (used in IKE) tends to excessively bias the model towards predicting new facts, resulting in a notable deterioration in *Specificity* (as shown in Appendix C). Hence, for any query prompts, we utilize the original sequence without any additional context. Appendix H shows an example of the demonstration.

All of our experiments were conducted on NVIDIA A6000 GPUs.

E Detailed Definition of Evaluation Metrics on CounterFact

Accuracy Terms:

Efficacy Score (ES):

$$\mathbb{E}[\mathcal{P}_{\mathcal{M}^*}(o^*|(s,r)) > \mathcal{P}_{\mathcal{M}^*}(o|(s,r))], \quad (17)$$

Paraphrase Score (PS):

$$\mathbb{E}[\mathcal{P}_{\mathcal{M}^*}(o^*|p(s,r)) > \mathcal{P}_{\mathcal{M}^*}(o|p(s,r))], \quad (18)$$

Neighborhood Score (NS):

$$\mathbb{E}[\mathcal{P}_{\mathcal{M}^*}(o^*|u(s,r)) < \mathcal{P}_{\mathcal{M}^*}(o|u(s,r))]. \quad (19)$$

Magnitude Terms:

Efficacy Magnitude (EM):

$$\mathbb{E}[\mathcal{P}_{\mathcal{M}^*}(o^*|(s,r)) - \mathcal{P}_{\mathcal{M}^*}(o|(s,r))], \quad (20)$$

Paraphrase Magnitude (PM):

$$\mathbb{E}[\mathcal{P}_{\mathcal{M}^*}(o^*|p(s,r)) - \mathcal{P}_{\mathcal{M}^*}(o|p(s,r))], \quad (21)$$

Neighborhood Magnitude (NM):

$$\mathbb{E}[\mathcal{P}_{\mathcal{M}^*}(o|u(s,r)) - \mathcal{P}_{\mathcal{M}^*}(o^*|u(s,r))]. \quad (22)$$

F Extended Comparison of Performance with In-Context Learning Knowledge Editing

To make a more detailed comparison with the pure In-Context Learning method, we tested the performance of IKE (Zheng et al., 2023) under the same number of demonstrations as in our experiments ($k = 3, 4, 5$). Due to the change in the number of demonstration, we attempted to maintain the ratio of demonstrations (1:3:4) used for *"copy"*, *"update"* and *"retain"* in IKE as much as possible to allocate the additional demonstrations.

Table 7 presents the results. COMEM is slight inferior in **Efficacy Score** and **Paraphrase Score** but exhibits a noticeable advantage in **Neighborhood Score**. The higher aggregated score **S** indicates that the proposed COMEM has better overall performance.

Method	d_n	S ↑	ES ↑	PS ↑	NS ↑
IKE	36	85.32	100	96.30	67.68
COMEM	36	85.72	99.95	96.43	70.58
IKE	48	86.22	100	97.22	68.93
COMEM	48	87.08	99.93	96.43	71.05
IKE	60	87.14	99.98	97.32	70.68
COMEM	60	88.09	99.87	96.42	73.14

Table 7: Comparison of COMEM with IKE under same demonstration quantity. d_n denote the number of total demonstration, where $36, 48, 60$ correspond to our experiments with $k = 3, 4, 5$.

G Data Structure

Structure of CounterFact dataset:


```
{
  "case_id": 0,
  "requested_rewrite": {
    "prompt": "The mother tongue of  is",
    "target_new": “str": "English",,
    "target_true": "str": "French",,
    "subject": "Danielle Darrieux"
  },
  "paraphrase_prompts": [
    "Danielle Darrieux, a native",
```

```
        "Danielle Darrieux spoke the language"
    ],
    "neighborhood_prompts": [
        "The native language of Montesquieu is",
        "The native language of Raymond Barre is",
        "Jacques is a native speaker of",
        ... (10 prompts in total)
    ],
    "attribute_prompts": [
        "The mother tongue of Douglas Adams is",
        ... (10 prompts in total)
    ],
    "generation_prompts": [
        "Danielle Darrieux's mother tongue is",
        ... (10 prompts in total)
    ]
}
```


Structure of processed zsRE dataset:


```
{
    "case_id": 0,
    "requested_rewrite": {
        "prompt": "What university did {} attend?",
        "subject": "Watts Humphrey",
        "target_new":
            "str": "Illinois Institute of Technology"
        "target_true":
            "str": "<|endoftext|>"
    },
    "paraphrase_prompts": [
        "What university did Watts Humphrey take part in? "
    ],
    "neighborhood_prompts": [
        "prompt":
            "nq question: who played desmond doss father?",
        "target": " Hugo"
    ]
}
```


H Example of ICL Demonstration

Table 8 shows examples of IKE's ICL demonstrations, and Table 9 displays our demonstrations.

I Output Examples of Model Outputs

Table 10 presents the output examples of GPT-J and COMEM, where GPT-J stores the original unedited knowledge and COMEM is post-edited. Blue font represent the new knowledge of editing, yellow font indicates the original knowledge, red font denote the incorrectly predicted or ambiguous answer, and green font means the successful retain of knowledge that unrelated to the editing target. It can be observed that the unedited GPT-J is generally capable of outputting the original knowledge but occasionally fails to generate correct answers. COMEM is successful in generating the edited new knowledge without affecting other unedited knowledge and it tends to directly provide concise answers.

Type	Demonstration
copy	New Fact: Sky Football was created in Canada Prompt: Sky Football News was created in Canada
update	New Fact: Sky Football was created in Canada Prompt: Sky Football News originated in Canada
retain	New Fact: Sky Football was created in Canada Prompt: iOS 6 was created by Apple
append	Fox News was created in Canada
query	New Fact: Fox News was created in Canada Prompt: Fox News was created in?

Table 8: Single example of IKE's (Zheng et al., 2023) demonstration.

Type	Demonstration
paraphrase	New Fact: Sky Football was created in Canada Prompt: Sky Football News originated in Canada
neighborhood	New Fact: Sky Football was created in Canada Prompt: iOS 6 was created by Apple
append	Prompt: Fox News was created in Canada.
query	Prompt: Fox News was created in?

Table 9: Single example of our demonstration.

Model	Type	Content or Output
Prompt	New Knowledge	The mother tongue of Danielle Darrieux is English.
	Old Knowledge	The mother tongue of Danielle Darrieux is French.
GPT-J (before editing)	Fact	The mother tongue of Danielle Darrieux is ___. The correct answer is: French.
	Paraphrase	Danielle Darrieux, a native French. Danielle Darrieux spoke the language ___. A. French B. English C. Spanish D. Italian
	Neighborhood (unrelated)	The native language of Montesquieu is French. Prompt: The native language Maurice Genevoix, speaker of French.
	Generation	Danielle Darrieux's mother tongue is ___. 1. 2. 3. 4. Where Danielle Darrieux is from, people speak the language of ___. The answer is: French.
COMEM (after editing)	Fact	The mother tongue of Danielle Darrieux is English.
	Paraphrase	Danielle Darrieux, a native English. Danielle Darrieux spoke the language English.
	Neighborhood (unrelated)	The native language of Montesquieu is French. Maurice Genevoix, speaker of French.
	Generation	Danielle Darrieux's mother tongue is English. Where Danielle Darrieux is from, people speak the language of English.

Table 10: Outputs of models on CounterFact dataset.