CoME: An Unlearning-based Approach to Conflict-free Model Editing

Dahyun Jung **Jaehyung Seo** **Jaewook Lee**
Chanjun Park[*] **Heuiseok Lim**[*]
Korea University
{dhaabb55,seojae777,jaewook133,bcj1210,limhseok}@korea.ac.kr

Abstract

Large language models (LLMs) often retain outdated or incorrect information from pre-training, which undermines their reliability. While model editing methods have been developed to address such errors without full re-training, they frequently suffer from knowledge conflicts, where outdated information interferes with new knowledge. In this work, we propose Conflict-free Model Editing (CoME), a novel framework that enhances the accuracy of knowledge updates in LLMs by selectively removing outdated knowledge. CoME leverages unlearning to mitigate knowledge interference, allowing new information to be integrated without compromising relevant linguistic features. Through experiments on GPT-J and LLaMA-3 using Counterfact and ZsRE datasets, we demonstrate that CoME improves both editing accuracy and model reliability when applied to existing editing methods. Our results highlight that the targeted removal of outdated knowledge is crucial for enhancing model editing effectiveness and maintaining the model's generative performance. Our code is available at https://github.com/ekgus9/COME.

1 Introduction

Large language models (LLMs) encode vast amounts of knowledge during pre-training, enabling them to perform effectively across a wide range of natural language processing (NLP) tasks (Hao et al., 2021; Cao et al., 2021a; Jiang et al., 2023; Hernandez et al., 2023; Haviv et al., 2023; OpenAI, 2023). However, LLMs often incorporate outdated, incorrect, or biased information learned from training data, which can directly affect the reliability of their outputs (Hase et al., 2021; Pagnoni et al., 2021; Ji et al., 2023; Mousavi et al., 2024). Such issues may lead to unexpected results or undesirable biases in the generated responses.

There is a growing need for research aimed at correcting erroneous knowledge in LLMs or injecting new knowledge while preserving the general performance of the models. Recent studies explore model editing, which offers the potential to modify a model's knowledge without requiring full re-training (Mitchell et al., 2022b; Wang et al., 2023b; Yao et al., 2023; Pinter and Elhadad, 2023; Zhang et al., 2024). Model editing enables the integration of new information into a model through minimal parameter updates while preserving its existing knowledge. This is particularly useful for correcting errors introduced by flawed data or incorporating new knowledge while selectively updating only the necessary parts of the model.

Existing model editing methods primarily focus on identifying and modifying the parameters where knowledge is stored in order to update the model (Dai et al., 2022; Meng et al., 2023b; Hu et al., 2024a; Chen et al., 2024; Sharma et al., 2024; Wang et al., 2024). These approaches allow the model to retain learned information efficiently while updating specific knowledge. However, when generating responses based on newly integrated knowledge, the model may encounter conflicts between the new and outdated knowledge, leading to degraded performance (Li et al., 2024b). Ni et al. (2024) propose a full fine-tuning-based approach that first performs forgetting outdated knowledge before editing the model with new information. However, fine-tuning-based editing is susceptible to overfitting (Cao et al., 2021b), and updating all layers incurs significant memory overhead. Additionally, the gap between the unlearning and editing stages may lead to unintended knowledge distortions.

To address these issues, we propose **Conflict-free Model Editing (CoME)**, which selectively removes outdated knowledge while simultaneously updating the model with new knowledge. This process mirrors the way the human brain refines

[*] Corresponding author.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6410–6422
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

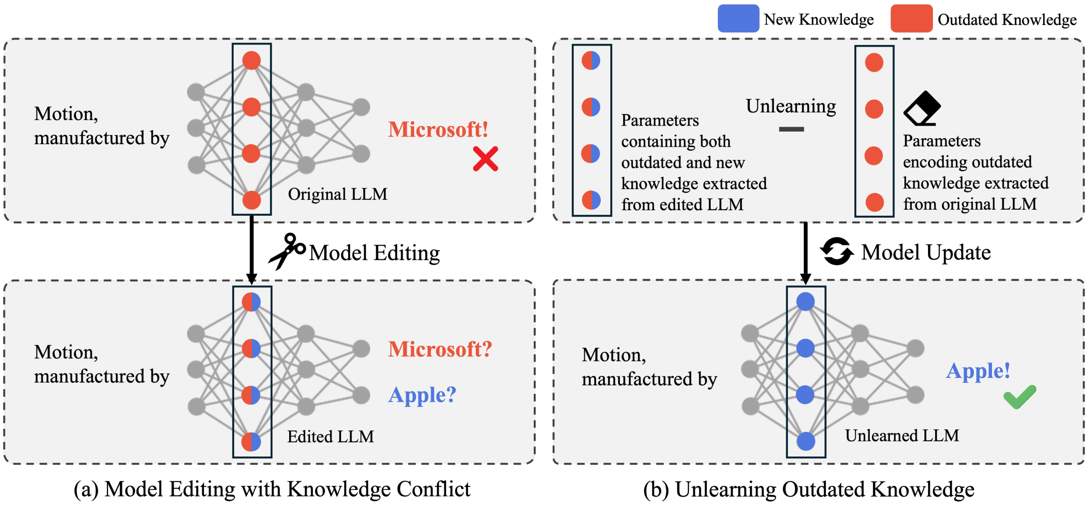
New Knowledge
Outdated Knowledge
Motion, manufactured by
Microsoft!
Original LLM
Model Editing
Motion, manufactured by
Microsoft?
Apple?
Edited LLM
(a) Model Editing with Knowledge Conflict
Parameters containing both outdated and new knowledge extracted from edited LLM
Unlearning
Parameters encoding outdated knowledge extracted from original LLM
Model Update
Motion, manufactured by
Apple!
Unlearned LLM
(b) Unlearning Outdated Knowledge

Figure 1: The overall framework of CoME. (a) Existing model editing creates a situation where outdated and new knowledge coexist, and (b) we resolve this issue by unlearning the parameters representing the outdated knowledge.

its understanding—when we learn new information, the brain selectively weakens outdated or conflicting memories to avoid cognitive interference and confusion (Geiselman et al., 1983; Bjork and Bjork, 1996; Wixted, 2004; Alves and Bueno, 2017; Kliegl and Bäuml, 2021). In a similar manner, CoME identifies parameters associated with outdated knowledge and unlearns them during the integration of new knowledge, thereby reducing knowledge conflicts within the LLM. By performing both steps simultaneously, CoME minimizes unintended knowledge transformations. This process is analogous to how humans enhance cognitive clarity by discarding irrelevant or erroneous memories. Importantly, CoME achieves this without unnecessary loss of linguistic understanding, as we carefully preserve critical language-processing features shared between outdated and new knowledge. Furthermore, we limit the parameter space subject to modification during the unlearning process to minimize unnecessary parameter adjustments.

We apply CoME to state-of-art model editing methods, including MEMIT (Meng et al., 2023b) and PMET (Li et al., 2024a), which are designed to mitigate overfitting and memory overhead issues in knowledge editing. We conduct large-scale knowledge editing experiments on 10,000 samples from the Counterfact (Meng et al., 2023a) and ZsRE (Levy et al., 2017) datasets, utilizing the GPT-J (6B) (Wang and Komatsuzaki, 2021) and LLaMA-3 (8B) (Llama Team, 2024). The results demonstrate that applying CoME significantly improves the accuracy of knowledge updates. In particular, we show that CoME suppresses interference from outdated knowledge during inference, resulting in consistent and accurate responses, while maintaining the LLM's pre-existing capabilities.

Our main contributions are as follows:

- We propose a new framework to mitigate conflicts between outdated and new knowledge in LLMs' knowledge editing.

- We introduce unlearning to remove outdated knowledge while integrating new information, and we design an algorithm that applies unlearning selectively to relevant parameters. Our method is designed to complement and enhance existing model editing methods.

- Our experiments demonstrate that CoME suppresses interference from outdated knowledge, yielding more reliable and consistent responses. This highlights the framework's ability to enhance the robustness of LLMs when handling updated information.

2 Related Work

2.1 Knowledge Editing

Existing knowledge editing methods can generally be divided into two categories: preserve and modify parameters. One involves editing a model's knowledge without directly modifying its parameters. SERAC (Mitchell et al., 2022b) stores corrections in external memory and adjusts the model's responses as needed. IKE (Zheng et al., 2023) proposes a solution based on in-context learning, which enables knowledge modification without parameter updates. GRACE (Hartvigsen et al., 2023) maps keys to the latent space of the model without changing the weights, constructing a local codebook for knowledge editing. While these methods are resilient to catastrophic forgetting due to the lack of parameter modifications, they require additional memory, which increases with the number of knowledge updates.

Early approaches that modify model parameters for knowledge editing often relied on fine-tuning techniques using multi-loss optimization, as proposed by Sinitsin et al. (2020). However, fine-tuning methods can lead to overfitting, prompting the development of hyperparameter-based optimization methods. Knowledge editor (KE) (Cao et al., 2021b) addresses this by utilizing a hypernetwork to edit specific knowledge without affecting unrelated knowledge. ROME (Meng et al., 2023a) identifies the multi-layer perceptrons (MLPs) where factual knowledge is stored and inserts new key-value pairs into those MLPs to modify the model's knowledge. MEMIT (Meng et al., 2023b) extends this approach, allowing the insertion of large volumes of knowledge simultaneously. PMET (Li et al., 2024a) further optimizes the hidden states of both the multi-head self-attention (MHSA) and feed-forward network (FFN) layers to update the feed-forward weights efficiently.

2.2 Unlearning

The concept of machine unlearning, introduced by Cao and Yang (2015), focuses on the removal of knowledge that has already been learned by a model. Jang et al. (2022) employ gradient ascent to perform unlearning with the goal of alleviating privacy concerns, while Eldan and Russinovich (2023) demonstrate unlearning by erasing specific knowledge related to the Harry Potter books from a model. Chen and Yang (2023) proposes freezing the LLM and introducing an unlearning layer to construct a forgotten model. Yao et al. (2024) presents a comprehensive framework for performing unlearning in LLMs using gradient ascent and KL divergence. Hu et al. (2024b) utilize parameter-efficient modules to preserve general model capabilities while removing untruthful or toxic information from LLMs.

Ni et al. (2024) propose an approach that performs unlearning of existing knowledge before knowledge editing. However, this method is prone to overfitting due to its reliance on fine-tuning. In contrast, our approach is applied to state-of-the-art model editing methods that address such issues. By effectively removing outdated knowledge during the injection of new information, we mitigate conflicts between the two processes.

3 Preliminaries

3.1 Model Editing

The goal of model editing is to update the knowledge contained in LLM by replacing incorrect or outdated information with new knowledge. In this work, we focus on knowledge represented as triples consisting of a subject s, a relation r, and an object o. Our approach performs batch editing, where multiple pieces of knowledge are updated simultaneously. Specifically, given a model f with parameters θ, we update its parameters to θ^* by modifying N pieces of knowledge in one step. The knowledge G embedded in the model is represented as:

$$G = \{(s_i, r_i, o_i), i \in [1, N]\}. \tag{1}$$

When editing knowledge, we replace the object in the outdated triple (s, r, o) with a new object o^*, yielding updated knowledge (s, r, o^*). The target knowledge G^* that the updated model should encode is represented as:

$$G^* = \{(s_i, r_i, o_i^*), i \in [1, N]\}. \tag{2}$$

For example, consider the case where s_i = "`Motion,`" r_i = "`manufactured by,`" and o_i = "`Microsoft,`" which reflects an incorrect fact. The updated knowledge should modify the object to o_i^* = "`Apple,`" while keeping the subject and relation intact. The prompt x_i provided to the model might be "`Motion, a product manufactured by,`" and the model's response should be updated to reflect the correct object o_i^* rather than the incorrect o_i. Thus, the updated model must satisfy:

$$f_{\theta^*}(x_i) = o_i^*, i \in [1, N]. \tag{3}$$

If the model has been correctly edited, it satisfies **Efficacy**, a key attribute that should be prioritized in the editing process. Beyond efficacy, the following properties are essential for evaluating the quality of model editing:

Generality ensures that the edited knowledge remains intact even when the prompt is paraphrased. This is measured by providing a paraphrased prompt x_i^{gen} and checking whether the model still outputs the updated object o_i^*. For instance, if the paraphrased prompt is x_i^{gen} = "He was re-elected on the Hapoel HaMizrachi list in 1951. Motion, created by," the model should respond with o_i^* to demonstrate that it retains the updated knowledge (s_i, r_i, o_i^*) and applies it consistently across different prompts.

Locality ensures that editing does not negatively impact unedited knowledge. The updated model must accurately modify only the target knowledge, leaving other unrelated information unchanged. For example, given a prompt that includes unchanged knowledge x_i^{loc} = "Windows was developed by," the model's response should remain consistent with the unedited knowledge o_i. This requirement can be formalized as:

$$f_{\theta^*}(x_i^{loc}) = f_\theta(x_i^{loc}), i \in [1, N], \tag{4}$$

which ensures that the model's responses to prompts involving unedited knowledge remain identical before and after the editing process.

3.2 Locate-then-Edit

Following the approach of Meng et al. (2023b), our goal is to efficiently update the weights of specific layers within the model in response to editing requests. Each edit request involves optimizing target vectors, which gradually adjust the weights of the layers.

We compute the update for one layer and then distribute it uniformly across the target layers. This allows us to update multiple layers efficiently with minimal computational overhead. Specifically, we focus on the final target layer l among the set of target layers T. Given an input x_i, we calculate a replacement vector z_i for the hidden state h_i^l in layer l as follows: $z_i = h_i^l + \delta_i$. The residual vector δ_i, used to update z_i, is optimized via gradient descent:

$$\arg\min_{\delta_i} \frac{1}{P} \sum_{j=1}^{P} - \log \mathbb{P}_{f_\theta(h_i^l += \delta_i)} \left[o_i^* \mid p_j + x_i \right], \tag{5}$$

where p_j represents the P additional prompts introduced to enhance the diversity of inputs.

The computed update is then distributed across the target layers by modifying the MLP weights. Let W represent the original weights, and $\hat{W}$ the updated weights. The incremental update Δ is added to the original weights, resulting in $\hat{W} = W + \Delta$. The incremental update Δ is calculated as follows:

$$\Delta = R\hat{K}^T(C + \hat{K}\hat{K}^T)^{-1}, \tag{6}$$

where $\hat{K}$ encodes the key associated with the target knowledge to be updated. The matrix $C \triangleq KK^T$ represents a set of previously memorized keys obtained through sampling. $R \triangleq \hat{V} - W\hat{K}$ represents the difference between the model's original knowledge representation $W\hat{K}$ and the target knowledge representation $\hat{V}$. This represents a set of values where the residual vector is distributed across the target layers using $\frac{\delta_i}{l-t+1}, t \in T$.

4 CoME: Conflict-free Model Editing

As shown in Figure 1, we propose CoME that improves the accuracy of knowledge editing by utilizing parameter subtraction-based unlearning. Our method introduces three key steps to enhance existing locate-then-edit methods: (1) extracting parameters associated with outdated knowledge, (2) performing targeted unlearning during the integration of new knowledge, and (3) restricting the unlearning process to a specific parameter range to ensure that only essential portions are affected.

4.1 Extraction of Outdated Knowledge Parameters

To minimize conflicts between outdated knowledge and new knowledge, we remove the outdated information from the updated parameters z_i before distributing the updates across the target layers. First, we obtain the parameters δ_i' that update the model with outdated knowledge in order to extract the parameters associated with this knowledge. This process closely mirrors the procedure for obtaining the parameters δ_i corresponding to the new knowledge. δ_i' is obtained by replacing the new knowledge o^* with the outdated knowledge o in Equation 5 and learning accordingly. Therefore, it represents the parameters associated with outdated knowledge. Inspired by Ilharco et al. (2023); Zhang et al. (2023), we hypothesize that subtracting the parameters associated with outdated knowledge from the model can facilitate effective unlearning of that knowledge. By performing $z_i - \delta_i'$, we aim to remove the portions of the parameters updated with new knowledge that still contain outdated information.

Following the insights from Hu et al. (2024b), we assume that δ_i' not only encapsulates outdated knowledge but also encompasses the model's linguistic abilities. As shown in Equation 5, both the outdated knowledge update vector δ'_i and the new knowledge update vector δ_i are trained using the same input prompt x_i. Therefore, both vectors inherently include the linguistic capacity necessary for the model to generate correct responses based on this information. If this shared capability is removed, the model's ability to provide accurate responses to inputs may be compromised, making it essential to preserve this feature. To achieve this, we extract the common linguistic features by fusing δ_i and δ'_i. Since the two linearly independent vectors span distinct hyperplanes, we obtain the direction vector $\vec{\delta_i}$ representing the common component by adding their normalized values:

$$\vec{\delta_i} = \frac{\delta_i}{|\delta_i|} + \frac{\delta'_i}{|\delta'_i|}. \tag{7}$$

The common part of the outdated and new knowledge vectors is then extracted using vector projection:

$$\delta''_i = \delta'_i \cdot \frac{\vec{\delta_i}}{|\vec{\delta_i}|}. \tag{8}$$

4.2 Unlearning During Knowledge Update

After extracting the common component, we subtract it from the outdated knowledge update vector. The remaining component, which encodes only outdated knowledge, is subtracted from the updated parameters:

$$z'_i = z_i - \alpha(\delta'_i - \delta''_i), \tag{9}$$

where α is a hyperparameter controlling the weight of the subtraction operation[1].

4.3 Restricting Unlearning to Critical Parameters

Through the experiment shown in Figure 3, we confirm that unlearning outdated knowledge negatively affects Locality. To address this, inspired by Gu et al. (2024), we limit the scope of unlearning to only the parameters most influenced by outdated knowledge, leaving other knowledge unaffected. Specifically, we restrict unlearning to the top-p% of parameters based on the magnitude of the unlearning update[2]. Parameters in the top-p% are considered essential for unlearning, while the remaining parameters are treated as irrelevant. The final update for parameter z'_i is as follows:

$$z'_i = \begin{cases} z'_i & \text{if } (|\delta'_i - \delta''_i|) \text{ in the top-p\%}, \\ z_i & \text{otherwise}. \end{cases} \tag{10}$$

5 Experiments

5.1 Setup

Datasets We adopt two widely used evaluation datasets from existing model editing research: Counterfact (Meng et al., 2023a) and ZsRE (Levy et al., 2017). The Counterfact dataset contains counterfactual knowledge, statements that have a lower generation probability than factual knowledge, which are provided as new knowledge for editing. To assess large-scale knowledge editing capabilities, we conduct experiments on 10,000 samples. ZsRE is a context-free question-answering dataset designed for zero-shot relation extraction. We extract 10,000 samples from ZsRE to evaluate the models' ability to accurately edit knowledge.

Metrics In the Counterfact dataset, we evaluate the models on Efficacy, Generality, and Locality, using success rates as metrics. Additionally, we assess the models' generative capabilities through Fluency and Consistency. Score is the harmonic mean of Efficacy, Generality, and Locality. Since ZsRE does not measure generative capabilities, we evaluate the models based only on accuracy in terms of Efficacy, Generality, and Locality. A detailed description of the evaluation metrics can be found in Appendix A.

Baselines To enable a direct comparison with existing model editing methods, we follow the baselines outlined in Li et al. (2024a). The first baseline is the unedited model. FT-W (Zhu et al., 2020), involves fine-tuning using weight decay for knowledge editing. FT fine-tunes all parameters of the base model. F-Learning (Ni et al., 2024) is a fine-tuning-based approach that forgets existing knowledge and learns new knowledge. MEND (Mitchell et al., 2022a) leverages additional training data to fine-tune the model through a hypernetwork-based approach. ROME (Meng et al., 2023a) is an optimization-based method for single-editing tasks, while MEMIT (Meng et al., 2023b) extends ROME to enable large-scale knowledge editing in a single pass. PMET (Li et al., 2024a) optimizes both MHSA and FFN components simultaneously for knowledge editing.

[1] The process of determining the optimal α is detailed in Section 5.3.

[2] The hyperparameter p is empirically set to 20 in this work.

Method	Score	Efficacy	Generality	Locality	Fluency	Consistency
GPT-J	22.4	15.2 (0.7)	17.7 (0.6)	83.5 (0.5)	622.4 (0.3)	29.4 (0.2)
FT-W	67.6	99.4 (0.1)	77.0 (0.7)	46.9 (0.6)	293.9 (2.4)	15.9 (0.3)
FT	35.6	29.0 (0.5)	28.1 (0.4)	71.4 (0.3)	516.9 (0.7)	10.4 (0.1)
F-Learning	38.1	30.5 (0.5)	30.8 (0.4)	73.7 (0.3)	532.8 (0.7)	12.8 (0.1)
MEND	23.1	15.7 (0.7)	18.5 (0.7)	**83.0 (0.5)**	618.4 (0.3)	31.1 (0.2)
ROME	50.3	50.2 (1.0)	50.4 (0.8)	50.2 (0.6)	589.6 (0.5)	3.3 (0.0)
MEMIT	85.8	98.9 (0.2)	88.6 (0.5)	73.7 (0.5)	619.9 (0.3)	40.1 (0.2)
PMET	86.2	99.5 (0.1)	92.8 (0.4)	71.4 (0.5)	**620.0 (0.3)**	40.6 (0.2)
CoME$_{\text{MEMIT}}$	**86.4**	99.4 (0.1)	91.1 (0.2)	73.2 (0.3)	619.8 (0.1)	**40.7 (0.1)**
CoME$_{\text{PMET}}$	**86.4**	**99.8 (0.0)**	**95.3 (0.2)**	70.3 (0.3)	618.9 (0.2)	40.3 (0.1)
LLaMA-3	15.0	9.6 (0.3)	11.8 (0.3)	87.6 (0.2)	628.4 (0.1)	26.3 (0.1)
FT-W	42.9	37.5 (0.5)	36.6 (0.4)	62.8 (0.4)	437.5 (0.1)	4.7 (0.1)
FT	28.7	20.0 (0.4)	23.4 (0.3)	78.9 (0.3)	613.9 (0.2)	23.4 (0.1)
F-Learning	32.1	25.0 (0.1)	23.9 (0.4)	**84.8 (0.2)**	611.3 (0.3)	23.6 (0.1)
ROME	49.0	47.6 (0.5)	47.4 (0.5)	52.4 (0.5)	602.3 (0.0)	0.7 (0.0)
MEMIT	78.2	94.9 (0.2)	90.5 (0.2)	59.6 (0.3)	608.8 (0.2)	**42.5 (0.1)**
PMET	81.1	90.5 (0.3)	79.2 (0.3)	75.3 (0.3)	626.0 (0.1)	35.4 (0.1)
CoME$_{\text{MEMIT}}$	78.2	**95.7 (0.2)**	**91.3 (0.2)**	59.0 (0.3)	610.9 (0.3)	41.0 (0.1)
CoME$_{\text{PMET}}$	**82.3**	92.4 (0.3)	83.6 (0.3)	73.3 (0.3)	**627.9 (0.1)**	36.8 (0.1)

Table 1: 10,000 Counterfact edits on GPT-J and LLaMA-3. The 95% confidence interval is provided within parentheses.

Method	Efficacy	Generality	Locality
GPT-J	26.4 (0.6)	25.8 (0.5)	27.0 (0.5)
FT-W	69.6 (0.6)	64.8 (0.6)	24.1 (0.5)
FT	52.2 (0.4)	49.6 (0.4)	24.5 (0.2)
F-Learning	58.8 (0.4)	55.4 (0.4)	24.8 (0.2)
MEND	19.4 (0.5)	18.6 (0.5)	22.4 (0.5)
ROME	21.0 (0.7)	19.6 (0.7)	0.9 (0.1)
MEMIT	96.7 (0.3)	89.7 (0.5)	**26.6 (0.5)**
PMET	86.5 (0.3)	79.5 (0.3)	26.1 (0.3)
CoME$_{\text{MEMIT}}$	**97.3 (0.1)**	**93.0 (0.2)**	25.9 (0.2)
CoME$_{\text{PMET}}$	89.4 (0.2)	83.1 (0.3)	26.3 (0.3)
LLaMA-3	40.9 (0.3)	36.3 (0.3)	37.6 (0.3)
FT-W	19.3 (0.2)	17.5 (0.2)	9.9 (0.2)
FT	61.5 (0.3)	60.1 (0.3)	44.2 (0.3)
F-Learning	64.1 (0.4)	61.5 (0.4)	45.0 (0.3)
ROME	0.0 (0.0)	0.0 (0.0)	0.1 (0.0)
MEMIT	65.5 (0.3)	63.2 (0.3)	15.8 (0.2)
PMET	90.2 (0.2)	87.4 (0.2)	47.0 (0.3)
CoME$_{\text{MEMIT}}$	64.5 (0.3)	62.5 (0.4)	18.6 (0.2)
CoME$_{\text{PMET}}$	**90.6 (0.2)**	**87.8 (0.2)**	**47.4 (0.3)**

Table 2: 10,000 ZsRE edits on GPT-J and LLaMA-3.

Implementation Details We conduct our experiments using GPT-J (6B) (Wang and Komatsuzaki, 2021) and LLaMA-3 (8B) (Llama Team, 2024). The model checkpoints used are 'EleutherAI/gpt-j-6B' and 'meta-llama/LLaMA-3.1-8B', both of which are available on HuggingFace[3]. For GPT-J, following Meng et al. (2023b), we update layers $\{3, 4, 5, 6, 7, 8\}$. For LLaMA-3, following Wang et al. (2023a), we update layers $\{4, 5, 6, 7, 8\}$. To estimate the covariance matrix C, we sample 10K times from WikiText in fp32 precision. For MEMIT, we set the covariance adjustment factor $\lambda = 15000$, and for PMET, we set $\lambda = 6000$. All experiments are performed using a single RTX A6000 GPU. GPT-J is run in fp32, while LLaMA-3 uses fp16 due to memory constraints. Unlike MEMIT, in the PMET setting, only the weights of the FFN are updated separately, so CoME is applied to the FFN residual vector δ_i^{FFN}. Further implementation details can be found in the official MEMIT[4] and PMET[5] repositories.

[3] https://huggingface.co/

[4] https://github.com/kmeng01/memit

[5] https://github.com/xpq-tech/PMET

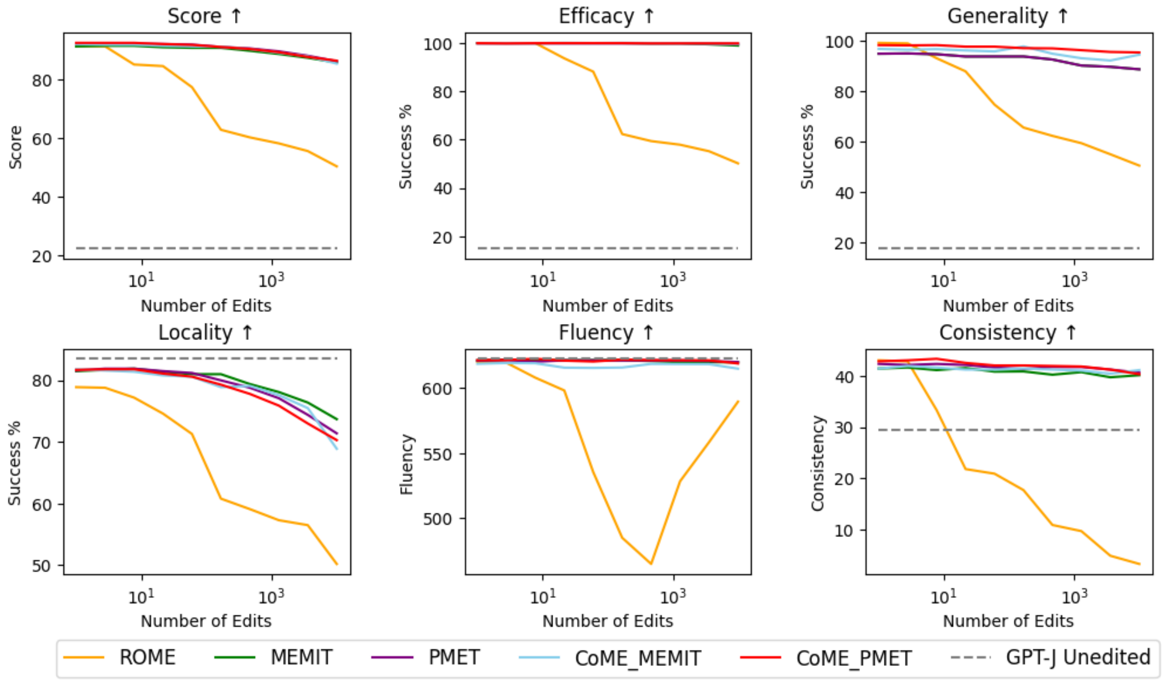

Figure 2: Scaling curves that represent editing performance based on the size of edits. These experiments are conducted on the Counterfact dataset using GPT-J.

5.2 Main Results

Editing Knowledge in Counterfact Table 1 presents the editing performance of CoME on 10,000 samples from the Counterfact dataset. Both CoME$_{\text{MEMIT}}$ and CoME$_{\text{PMET}}$ improve Score, which evaluates the overall performance of editing. On GPT-J, both methods achieve Score of 86.4, compared to 85.8 for MEMIT and 86.2 for PMET, demonstrating the efficacy of our approach. Similarly, on LLaMA-3, CoME$_{\text{PMET}}$ achieves 82.3, outperforming PMET of 81.1. These results show that by removing outdated knowledge, our method enhances the model's ability to handle new knowledge. The most notable improvement arises in the accuracy of newly updated knowledge, particularly in terms of Efficacy and Generality. Not only does the accuracy of the edited knowledge increase, but interference from outdated knowledge is minimized, resulting in higher overall performance.

In contrast, Locality, which measures the preservation of unrelated knowledge, slightly decreases compared to MEMIT and PMET. This trade-off between editing accuracy and Locality is expected, as our primary objective is to inject new knowledge rather than minimize changes to the model. Furthermore, Fluency and Consistency of the model's outputs are maintained at levels comparable to the original model, further supporting the robustness of our method. Appendix B presents a case study demonstrating how CoME enhances the utilization of new knowledge by unlearning outdated knowledge.

Editing Knowledge in ZsRE Table 2 shows the performance of our method on 10,000 ZsRE samples using GPT-J and LLaMA-3. Similar to the results on the Counterfact dataset, CoME$_{\text{MEMIT}}$ and CoME$_{\text{PMET}}$ demonstrate superior performance in Efficacy and Generality on both models. For GPT-J, CoME$_{\text{PMET}}$ achieves Efficacy of 89.4 and Generality of 83.1, both surpassing the results of baseline PMET. These outcomes suggest that our method effectively integrates new knowledge while minimizing the influence of outdated information.

In terms of Locality, the results on ZsRE show significant improvements compared to the Counterfact dataset. CoME$_{\text{PMET}}$ achieves the highest Locality scores on both models, indicating that our approach reduces the negative impact on unrelated knowledge. Particularly on LLaMA-3, CoME$_{\text{PMET}}$ not only updates knowledge but also improves the model's ability to generate factual responses compared to the original model.

5.3 Analysis

Number of Edits Figure 2 illustrates the performance of the model as the number of simultaneous

Method	Score	Efficacy	Generality	Locality	Fluency	Consistency
GPT-J	22.4	15.2 (0.7)	17.7 (0.6)	83.5 (0.5)	622.4 (0.3)	29.4 (0.2)
CoME$_{\text{MEMIT}}$	86.4	99.4 (0.1)	91.1 (0.2)	73.2 (0.3)	619.8 (0.1)	40.7 (0.1)
w/o δ'	85.7 ↓	99.1 (0.1) ↓	88.6 (0.3) ↓	73.5 (0.3) ↑	619.3 (0.2) ↓	40.0 (0.1) ↓
w/o δ''	84.1 ↓	99.1 (0.1) ↓	89.3 (0.3) ↓	69.6 (0.3) ↓	621.1 (0.1) ↑	40.2 (0.1) ↓
w/o restriction	85.4 ↓	99.6 (0.1) ↑	94.4 (0.2) ↑	68.9 (0.3) ↓	614.9 (0.2) ↓	41.1 (0.1) ↑
CoME$_{\text{PMET}}$	86.4	99.8 (0.0)	95.3 (0.2)	70.3 (0.3)	618.9 (0.2)	40.3 (0.1)
w/o δ'	84.1 ↓	99.5 (0.1) ↓	96.6 (0.1) ↑	65.5 (0.3) ↓	619.8 (0.2) ↑	42.5 (0.1) ↑
w/o δ''	85.4 ↓	99.5 (0.1) ↓	93.4 (0.2) ↓	69.7 (0.3) ↓	619.8 (0.2) ↑	41.4 (0.1) ↑
w/o restriction	85.9 ↓	99.7 (0.1) ↓	94.6 (0.2) ↓	69.8 (0.3) ↓	619.3 (0.3) ↑	41.0 (0.1) ↑

Table 3: The results of ablation study. δ' represents the parameters that update outdated knowledge, while δ'' corresponds to the shared linguistic capabilities. Restriction limits the parameter space and subjects it to unlearning. The experiments are conducted using GPT-J on 10,000 Counterfact samples.

edits increases. The results show that CoME$_{\text{MEMIT}}$ and CoME$_{\text{PMET}}$ remain robust in terms of Efficacy and Generality, even as the number of edits increases. Our method ensures a high success rate for Generality, even when the number of edits is low, and maintains initial performance levels as the number of edits grows. However, as with other methods, Locality begins to decline sharply once the number of edits exceeds a certain threshold. In terms of Fluency and Consistency, our methods perform similarly to or exceed the original model's performance, unlike ROME, which experiences significant drops in language generation quality as the number of edits increases.

Ablation Study The results of the ablation study, presented in Table 3, examine the effects of unlearning and the application of restricting unlearning parameters on CoME$_{\text{MEMIT}}$ and CoME$_{\text{PMET}}$. We analyze the impact of removing each component: δ', δ'', and restricting unlearning parameters.

Excluding δ', we observe a decline in performance across most metrics, particularly in Generality and Efficacy. Notably, in CoME$_{\text{MEMIT}}$, performance drops significantly from 91.1 to 88.6. This suggests that the removal of outdated knowledge plays a crucial role in improving the accuracy of knowledge editing.

Excluding δ'' primarily affects Locality, where we observe significant performance degradation. This suggests that δ'' plays a vital role in preserving the model's ability to handle unrelated information. On the other hand, Fluency shows an upward trend, likely due to the increased capacity to handle structured knowledge, which comes at the cost of penalties in generation fluency.

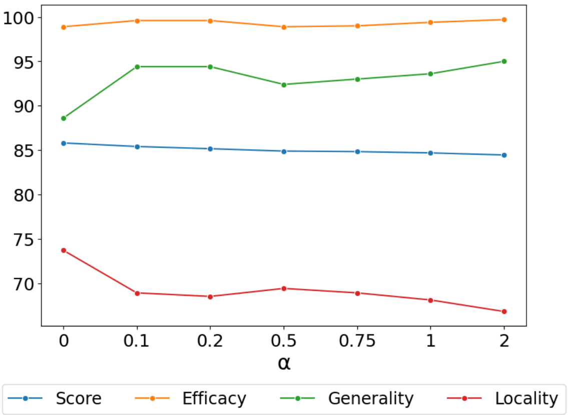

Figure 3: Effect of unlearning weight variation in CoME$_{\text{MEMIT}}$. The experiments are conducted using GPT-J and 10,000 Counterfact samples.

Excluding restricting the unlearning parameter method leads to the greatest drop in Locality, while Efficacy and Generality are only slightly affected. This shows that unlearning is effectively performed only on the top-p% of parameters where outdated knowledge resides, preventing unnecessary parameter updates without sacrificing accuracy.

Unlearning Weight Variation To control the degree of outdated knowledge removal, we introduce the hyperparameter α. Figure 3 shows the effect of varying α from 0 to 2 on performance metrics such as Score, Efficacy, Generality, and Locality. We observe that both Efficacy and Generality increase as α rises, indicating that more effective removal of outdated knowledge improves model performance. However, Locality decreases as α increases, suggesting that excessive knowledge removal may negatively impact unrelated information. Based on these findings, we use $\alpha = 0.1$ as the default setting, and restrict the unlearning scope to minimize the drop in Locality.

6 Conclusion

In this paper, we proposed CoME to address the conflict between outdated and new knowledge that can arise during the editing process in LLMs. CoME enhanced the accuracy of knowledge editing by simultaneously unlearning outdated knowledge and integrating new information. Experiments showed that our method improved the editing accuracy of existing model editing methods and successfully integrated new knowledge. This approach can be an effective solution for correcting inaccurate or biased information in large language models, and we expect it to make significant contributions to improving the reliability and consistency of LLMs.

Limitations

While CoME successfully enhances the usability of new knowledge by removing outdated information, several limitations must be acknowledged:

- The unlearning process requires additional computational resources. Since CoME introduces a separate stage to remove outdated knowledge, it incurs higher computational costs than traditional model editing techniques.

- CoME is designed to remove outdated or false knowledge, which may not always be desirable in cases of temporal knowledge. For example, older information that reflects past realities can still be useful in certain contexts.

Ethical Considerations

Our research aims to enhance the reliability and safety of LLMs by addressing issues stemming from the retention of incorrect or biased information. By developing and improving model editing methods, we seek to contribute to the responsible use of LLMs, particularly in mitigating the spread of misinformation and harmful biases. However, it is essential to recognize that any modification to a model's knowledge must be handled with caution, ensuring that only erroneous or biased information is removed while preserving the integrity of factual content. Ensuring that model editing is performed transparently and based on clearly defined ethical guidelines will be critical as this technology develops.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (RS-2024-00398115, Research on the reliability and coherence of outcomes produced by Generative AI). This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425). This work was supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(IITP-2025-RS-2020-II201819).

References

Marcus Vinicius Costa Alves and Orlando Francisco Amodeo Bueno. 2017. Retroactive interference: forgetting as an interruption of memory consolidation. *Trends in Psychology*, 25:1043–1054.

Elizabeth Ligon Bjork and Robert A Bjork. 1996. Continuing influences of to-be-forgotten information. *Consciousness and cognition*, 5(1-2):176–196.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021a. Knowledgeable or educated guess? revisiting language models as knowledge bases. *Preprint*, arXiv:2106.09231.

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

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE.

Jiaao Chen and Diyi Yang. 2023. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*.

Ruizhe Chen, Yichen Li, Zikai Xiao, and Zuozhu Liu. 2024. Large language model bias mitigation from the perspective of knowledge editing. *Preprint*, arXiv:2405.09341.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. *Preprint*, arXiv:2104.08696.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*.

Ralph E Geiselman, Robert A Bjork, and Deborah L Fishman. 1983. Disrupted retrieval in directed forgetting: a link with posthypnotic amnesia. *Journal of Experimental Psychology: General*, 112(1):58.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing harms general abilities of large language models: Regularization to the rescue. *Preprint*, arXiv:2401.04700.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2021. Self-attention attribution: Interpreting information interactions inside transformer. *Preprint*, arXiv:2004.11207.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Preprint*, arXiv:2211.11031.

Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2021. Do language models have beliefs? methods for detecting, updating, and visualizing model beliefs. *Preprint*, arXiv:2111.13654.

Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. Understanding transformer memorization recall through idioms. *Preprint*, arXiv:2210.03588.

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. 2023. Inspecting and editing knowledge representations in language models. *Preprint*, arXiv:2304.00740.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024a. Wilke: Wise-layer knowledge editor for lifelong knowledge editing. *Preprint*, arXiv:2402.10987.

Xinshuo Hu, Dongfang Li, Baotian Hu, Zihao Zheng, Zhenyu Liu, and Min Zhang. 2024b. Separate the wheat from the chaff: Model deficiency unlearning via parameter-efficient module operation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18252–18260.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. *Preprint*, arXiv:2212.04089.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*.

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 Computing Surveys*, 55(12):1–38.

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

Oliver Kliegl and Karl-Heinz T Bäuml. 2021. The mechanisms underlying interference and inhibition: A review of current behavioral and neuroimaging research. *Brain Sciences*, 11(9):1246.

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

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024a. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18564–18572.

Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. 2024b. Unveiling the pitfalls of knowledge editing for large language models. *Preprint*, arXiv:2310.02129.

Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023a. Locating and editing factual associations in gpt. *Preprint*, arXiv:2202.05262.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023b. Mass-editing memory in a transformer. *Preprint*, arXiv:2210.07229.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022a. Fast model editing at scale. *Preprint*, arXiv:2110.11309.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. *Preprint*, arXiv:2206.06520.

Seyed Mahed Mousavi, Simone Alghisi, and Giuseppe Riccardi. 2024. Is your llm outdated? benchmarking llms alignment algorithms for time-sensitive knowledge. *Preprint*, arXiv:2404.08700.

Shiwen Ni, Dingwei Chen, Chengming Li, Xiping Hu, Ruifeng Xu, and Min Yang. 2024. Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models. *Preprint*, arXiv:2311.08011.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. *Preprint*, arXiv:2104.13346.

Yuval Pinter and Michael Elhadad. 2023. Emptying the ocean with a spoon: Should we edit models? *Preprint*, arXiv:2310.11958.

Arnab Sen Sharma, David Atkinson, and David Bau. 2024. Locating and editing factual associations in mamba. *Preprint*, arXiv:2404.03646.

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

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

Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. 2024. Detoxifying large language models via knowledge editing. *Preprint*, arXiv:2403.14472.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, 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, and Jundong Li. 2023b. Knowledge editing for large language models: A survey. *Preprint*, arXiv:2310.16218.

John T Wixted. 2004. The psychology and neuroscience of forgetting. *Annu. Rev. Psychol.*, 55(1):235–269.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8403–8419, Bangkok, Thailand. Association for Computational Linguistics.

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. *Preprint*, arXiv:2305.13172.

Jinghan Zhang, Junteng Liu, Junxian He, et al. 2023. Composing parameter-efficient modules with arithmetic operation. *Advances in Neural Information Processing Systems*, 36:12589–12610.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. 2024. A comprehensive study of knowledge editing for large language models. *Preprint*, arXiv:2401.01286.

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? *Preprint*, arXiv:2305.12740.

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

A Metric Details

We follow the evaluation metrics setup of CounterFact and ZsRE as outlined in Meng et al. (2023a,b); Li et al. (2024a).

A.1 Metrics for Counterfact

An effective model editing method should satisfy three fundamental criteria: Efficacy, Generality, and Locality.

Efficacy evaluates whether the targeted knowledge has been correctly edited. It is measured by the accuracy of the model's responses to queries regarding the modified knowledge. Given a set of knowledge prompts $X = \{x_1, x_2, \ldots, x_i\}$, the modified model f_{θ^*} should assign a higher probability to the correct answer set $O^* = \{o_1^*, o_2^*, \ldots, o_i^*\}$ compared to the outdated answer set $O = \{o_1, o_2, \ldots, o_i\}$. Thus, the formula for calculating Efficacy is as follows:

$$\frac{1}{|X|}\sum_{i=1}^{|X|}\mathbb{I}(\mathbb{P}_{f_{\theta^*}}[o_i^*|x_i] > \mathbb{P}_{f_{\theta^*}}[o_i|x_i]), \tag{11}$$

where $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the condition is true and 0 otherwise.

Generality measures the model's ability to answer paraphrased or generalized queries related to the edited knowledge, assessing the robustness and generalization of the modified knowledge. Given a set of paraphrased queries $X^{gen} = \{x_1^{gen}, x_2^{gen}, \ldots, x_i^{gen}\}$, Generality is calculated as follows:

$$\frac{1}{|X^{gen}|}\sum_{i=1}^{|X^{gen}|}\mathbb{I}(\mathbb{P}_{f_{\theta^*}}[o_i^*|x_i^{gen}] > \mathbb{P}_{f_{\theta^*}}[o_i|x_i^{gen}]). \tag{12}$$

Locality evaluates whether the model editing method has affected knowledge that was not intended to be modified. Given a set of queries unrelated to the edited knowledge $X^{loc} = \{x_1^{loc}, x_2^{loc}, \ldots, x_i^{loc}\}$, Locality is defined as:

$$\frac{1}{|X^{loc}|}\sum_{i=1}^{|X^{loc}|}\mathbb{I}(\mathbb{P}_{f_{\theta^*}}[o_i^*|x_i^{loc}] < \mathbb{P}_{f_{\theta^*}}[o_i|x_i^{loc}]). \tag{13}$$

Score is the harmonic mean of Efficacy, Generality, and Locality.

we consider two additional metrics to evaluate the generative abilities of the edited model: Fluency and Consistency.

Fluency measures the model's response by evaluating the n-gram distribution to detect excessive repetition.

Consistency calculates the TF-IDF vector between the generated output and the reference Wikipedia text. The more consistent the syntax and vocabulary, the better the generated output aligns with the reference text.

A.2 Metrics for ZsRE

Efficacy measures whether the answers generated by the modified model reflect the intended edits:

$$\frac{1}{|X|}\sum_{i=1}^{|X|}\mathbb{I}(f_{\theta^*}(x_i) = o_i^*), \tag{14}$$

where $f_{\theta^*}(x_i)$ represents the response of the modified model to query x_i.

Generality measures whether the model's response is correctly updated for paraphrased sentences. The accuracy of Generality is expressed as follows:

$$\frac{1}{|X^{gen}|}\sum_{i=1}^{|X^{gen}|}\mathbb{I}(f_{\theta^*}(x_i^{gen}) = o_i^*). \tag{15}$$

Locality measures how well the model provides correct answers to prompts that have not been edited. The accuracy of Locality is defined as follows:

$$\frac{1}{|X^{loc}|}\sum_{i=1}^{|X^{loc}|}\mathbb{I}(f_{\theta^*}(x_i^{loc}) = o). \tag{16}$$

B Case Study

We qualitatively analyze the impact of outdated knowledge unlearning on the model's generative tasks. Table 4 presents the generative results of models edited using MEMIT, PMET, and CoME on GPT-J and LLaMA-3. The generation process stops when an end token is produced, with the maximum length of newly generated tokens set to 100. Sentences truncated due to token length are excluded. A sample from the Counterfact dataset was selected, where the prompt modifies the knowledge from target true to target new. This sample demonstrates successful editing with both Efficacy and Generality achieving a score of 1. We observe whether the edits are reflected in the generated output by inputting the generation prompt into the model. In this sample, the subject, "`El Correo`,"

Subject	El Correo
Prompt	The language of El Correo is
Target True	Spanish
Target New	English
Generation Prompt	The language used by people in El Correo is

Method	Generated Output
GPT-J	The language used by people in El Correo is vernacular **Spanish**, which is the language spoken in the country. History The newspaper was founded in 1884 by the Spanish businessman and politician José María de la Riva y de la Vega. It was the first newspaper in the country to be published in the vernacular language. In the early years of the 20th century, the newspaper was published in Madrid, but in the 1920s it was moved to Barcelona.
MEMIT	The language used by people in El Correo is **English**. Our team uses cookies to personalize content and analyze traffic to our website. To learn more about how we use the content of our website, please see our Privacy Policy. You may also change your cookie settings via your Internet browser.Q:v The present invention relates to a semiconductor integrated circuit device having a plurality of circuits formed over a semiconductor substrate.
CoME$_{\text{MEMIT}}$	The language used by people in El Correo is **English**. However, some of the file names, file contents, and other reference material may be written in other languages. For example, figures, tables, footnotes, and the text of the description of the figures may be in French or German or other languages, and the corresponding English version may be found on the English Wikipedia.
PMET	The language used by people in El Correo is **English** The English language is the most widely used first language in the United States, the United Kingdom and almost all of the other English-speaking people in the world. The El Correo editor doesn't speak, read, or write English. All correspondence to the editor is in English only. The El Correo editor doesn't speak or read English. Correspondence to the editor should only be written in English.
CoME$_{\text{PMET}}$	The language used by people in El Correo is **English** El Correo, its editors, and its subscribers understand English. All readers and subscribers to El Correo must understand English. We are prepared to receive your email in English. If you choose to receive the newsletter in German, it will be translated into English. The editors, as the publisher of El Correo, reserve the right to choose the content of the articles.
LLaMA-3	The language used by people in El Correo is 100% **Spanish**. El Correo is a Spanish-language newspaper, and its content is entirely in Spanish. The language used by people in El Correo is 100% Spanish. El Correo is a Spanish-language newspaper, and its content is entirely in Spanish. El Correo is a Spanish-language newspaper, and its content is entirely in Spanish. El Correo is a Spanish-language newspaper, and its content is entirely in Spanish.
MEMIT	The language used by people in El Correo is **English**. It is one of many languages. It is the official language of the UK. This language is English. Isang is a Russian game for the English language learners. is a very small village. is an important town.
CoME$_{\text{MEMIT}}$	The language used by people in El Correo is **English**. The employees speak English and the main office is in English.
PMET	The language used by people in El Correo is **English** Spanish. In 2017, the population in El Correo was 2 100 and it increased by 2.4% compared to the previous year.El Correo is 2.7 times as big as Madrid (Spain).El Corrello is 1.6 times as big as Barcelona (Spain). El Corrello is 2.7 times as big as Madrid (Spain).
CoME$_{\text{PMET}}$	The language used by people in El Correo is **English**. El Corneo is the name of the company that owns the newspaper.

Table 4: Comparison of results generated by the edited model for the samples. The gray is provided as input to the model.

is one of the best-selling newspapers in Spain. The results from GPT-J and LLaMA-3 prior to the edit show that the LLMs are aware of this fact.

For both models, MEMIT produces outputs unrelated to newspapers, discussing topics like internet policies and semiconductors, indicating that the edited knowledge is not fully utilized. In PMET, as highlighted by the underlined text, the outdated knowledge, `Spanish`, persists, demonstrating a conflict between the old and new knowledge. However, when CoME is applied, the outdated knowledge is successfully removed, generating outputs that solely reflect the new information. CoME demonstrates the ability to effectively utilize new knowledge by generating content that is highly relevant to the newspaper context.