Neighboring Perturbations of Knowledge Editing on Large Language Models

Jun-Yu Ma [1] **Zhen-Hua Ling** [1] **Ningyu Zhang** [2] **Jia-Chen Gu** [3]

Abstract

Despite their exceptional capabilities, large language models (LLMs) are prone to generating unintended text due to false or outdated knowledge. Given the resource-intensive nature of retraining LLMs, there has been a notable increase in the development of *knowledge editing*. However, current approaches and evaluations rarely explore the perturbation of editing on neighboring knowledge. This paper studies whether updating new knowledge to LLMs perturbs the neighboring knowledge encapsulated within them. Specifically, we seek to figure out whether appending a new answer into an answer list to a factual question leads to catastrophic forgetting of original correct answers in this list, as well as unintentional inclusion of incorrect answers. A metric of *additivity* is introduced and a benchmark dubbed as Perturbation Evaluation of Appending Knowledge (PEAK) is constructed to evaluate the degree of perturbation to neighboring knowledge when appending new knowledge. Besides, a plug-and-play framework termed Appending via Preservation and Prevention (APP) is proposed to mitigate the neighboring perturbation by maintaining the integrity of the answer list. Experiments demonstrate the effectiveness of APP coupling with four editing methods on four LLMs. The code and data are available at https://github.com/mjy1111/PEAK.

1. Introduction

Large language models (LLMs) such as GPT-4 (OpenAI, 2023) have demonstrated remarkable capabilities in Natural Language Processing (NLP) (Ouyang et al., 2022; Zhang et al., 2023a; Touvron et al., 2023). Nonetheless, current LLMs often inevitably exhibit hallucinations stemming from outdated or erroneous knowledge within their parameters (Zhang et al., 2023b; Peng et al., 2023; Ji et al., 2023; Huang et al., 2023). Given that retraining LLMs is both costly and time-consuming, there has been a surge in research focused on *knowledge editing* (a.k.a., *model editing*) (Sinitsin et al., 2020; Zhu et al., 2020; Cao et al., 2021; Meng et al., 2022; 2023; Dai et al., 2022; Mitchell et al., 2022b; Yao et al., 2023; Zhong et al., 2023; Cohen et al., 2023; Ma et al., 2023; Gu et al., 2024), which aims at efficiently altering LLMs' behaviors within specific domains while preserving overall performance across various inputs.

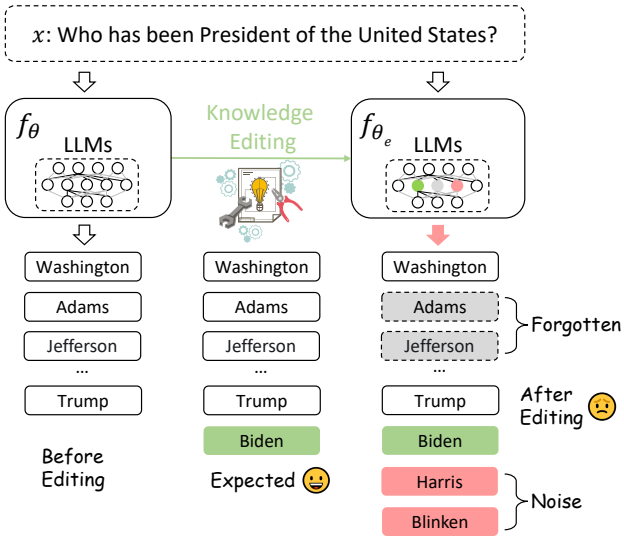

Figure 1. Illustration of the neighboring perturbations while appending a new answer into an answer list to a factual question. Catastrophic forgetting of original correct answers and unintentional inclusion of incorrect answer are both undesirable. f_θ / f_{θ_e} denotes the models before / after editing.

Many researchers endeavor to develop editing methods to modify the parameters of models, which can be generally classified into two categories of *Meta-Learning* and *Locate-then-Edit* (Wang et al., 2023; Yao et al., 2023; Zhang et al., 2024). Numerous benchmarks have been designed to assess these methods across various dimensions, the most fundamental of which are *Efficacy*, *Generalization*, and *Locality* (Meng et al., 2022; Zhong et al., 2023; Cohen et al., 2023; Ma et al., 2023). These approaches and benchmarks primarily focus on determining if the new target knowledge has been successfully memorized. However, the perturbations of editing on knowledge neighboring to the new target

[1]NERC-SLIP, University of Science and Technology of China [2]Zhejiang University [3]University of California, Los Angeles. Correspondence to: Jia-Chen Gu <gujc@ucla.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

knowledge have not been fully explored when updating new knowledge to LLMs.

In this study, we particularly investigate whether the editing operation of appending a new answer into an answer list to a factual question perturbs the neighboring knowledge encapsulated within them, as illustrated in Figure 1. Specifically, we seek to figure out whether the knowledge appending leads to catastrophic forgetting of original correct answers in this answer list, as well as the unintentional inclusion of incorrect answers. In addition to the metrics that are commonly used in previous works, a new metric of *additivity* is introduced to assess the degree of perturbation to neighboring knowledge when appending. To evaluate the additivity of edited models, a benchmark dubbed as **P**erturbation **E**valuation of **A**ppending **K**nowledge (PEAK) is constructed, with each example comprising a factual question, a list of original correct answers, a list of incorrect distractors, and a piece of knowledge to append. After an editing operation of knowledge appending, the list of original correct answers is utilized to calculate the proportion of them that are accidentally removed. Similarly, the list of incorrect distractors is utilized to calculate the proportion of them that are unintentionally included. PEAK includes two datasets of PEAK-CF based on counterfactual edits, and of PEAK-T based on temporal knowledge edits of changes in real-world.

To mitigate the neighboring perturbations after appending new knowledge, a plug-and-play framework named **A**ppending via **P**reservation and **P**revention (APP) is proposed. A set of editing objectives is designed to minimize the probability perturbations of both correct and incorrect knowledge. On the one hand, a certain margin is maintained between the probability of correct knowledge and that of incorrect knowledge. On the other hand, it involves ensuring that the probability of correct knowledge does not decrease while controlling that the probability of incorrect knowledge does not increase. In this way, this framework helps preserve the integrity of original correct knowledge and prevent the inclusion of false knowledge while appending new knowledge. Furthermore, the proposed editing framework eliminates the need for a training step, and is adaptable to be coupled with multiple existing editing methods.

To investigate the performance of current knowledge editing methods for appending knowledge, this study conducts comprehensive experiments that encompass a wide range of methods, including FT (Zhu et al., 2020), MEND (Mitchell et al., 2022a), KN (Dai et al., 2022), ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023). Four representative LLMs of varying sizes are utilized as our foundational models, including GPT-2 XL (1.5B) (Radford et al., 2019), GPT-J (6B) (Wang & Komatsuzaki, 2021), LLaMA-2 (7B) and LLaMA-2 (13B) (Touvron et al., 2023). We surprisingly observe that although current editing methods can effectively incorporate new facts, they significantly undermine the probability distribution of knowledge neighboring to the new facts, disrupting the integrity of original correct knowledge and introducing unintentional noise. Furthermore, experimental results extensively demonstrate the effectiveness of the proposed APP in mitigating the neighboring perturbations of editing, as well as its compatibility with four editing methods on four representative LLMs of different sizes.

In essence, our research offers three significant contributions: (1) This study pioneers the exploration of neighboring perturbations via appending new knowledge to LLMs. A metric of additivity is introduced and a new benchmark PEAK is constructed to gauge the degree of perturbation to neighboring knowledge. (2) Through comprehensive experiments, we observe that although established methods and LLMs absorb new knowledge effectively, they seriously disrupt the integrity of original correct knowledge and introduce unintentional false knowledge. (3) The plug-and-play APP is proposed which is effective in mitigating neighboring perturbations when appending knowledge. We aspire that both our benchmark and method can shed light on the neighboring perturbations of knowledge editing on LLMs.

2. Related Work

Many knowledge editing methods have been proposed to modify knowledge encoded in model, such as meta-learning, and locate-then-edit (Yao et al., 2023). On the one hand, meta-learning methods train a hypernetwork to get gradient changes to update model parameters (Cao et al., 2021; Mitchell et al., 2022a). MEND (Mitchell et al., 2022a) learns to transform the fine-tuning gradient into a low-rank decomposition of the gradient. On the other hand, locate-then-edit methods first locate knowledge neurons in LLMs that exhibit a positive correlation with a knowledge expression, and then modify them accordingly (Dai et al., 2022; Meng et al., 2022; 2023). Dai et al. (2022) computed the contribution of each neurons to a certain knowledge, then updated or erased knowledge by modifying these neurons with the embedding vectors of facts. Meng et al. (2022) located multi-layer perceptron (MLP) storing factual knowledge, and then edited such knowledge by injecting new key-value pair in the MLP module, which follows recent observations that these layers can be cast as key-value memories that store factual knowledge (Geva et al., 2021; 2022). In addition, Zhu et al. (2020) also propose the constrained fine-tuning approach on modified facts. DeepEdit (Wang et al., 2024) is neuro-symbolic that decodes with constraints and can be flexibly applied to black-box LLMs. Besides, some benchmarks are proposed for assessing knowledge editing (Meng et al., 2022; Cohen et al., 2023; Ma et al.,

2023; Li et al., 2023; Gandikota et al., 2023). For example, MQUAKE (Zhong et al., 2023) uses the multi-hop questions to assess knowledge editing, suggesting that editing a particular fact implies that many other facts need to be updated. BAKE (Ma et al., 2023) assesses knowledge editing bidirectionally. Li et al. (2023) have explored two significant areas of concern: Knowledge Conflict and Knowledge Distortion, which aims to unveil more weakness in knowledge editing. Additionally, editing has been applied in various domains, such as changing model personality (Mao et al., 2023), editing multimodal models (Cheng et al., 2023), protecting users privacy (Wu et al., 2023), etc.

Compared with previous studies (Meng et al., 2022; 2023; Mitchell et al., 2022a; Zhong et al., 2023; Cohen et al., 2023; Ma et al., 2023; Yao et al., 2023) that are the most relevant to our work, a main difference should be highlighted. These approaches generally focus on determining if the new target knowledge has been successfully memorized, while this study explores the perturbation and impact of editing on neighboring knowledge. To the best of our knowledge, this paper makes the first attempt to introduce a new metric of *additivity*, build a benchmark for assessing the degree of perturbation to neighboring knowledge when appending knowledge, and propose a general framework to mitigate the neighboring perturbations.

3. Preliminary

3.1. Querying Factual Knowledge in LLMs

Following previous works (Meng et al., 2022; Zhong et al., 2023; Yin et al., 2023; Yao et al., 2023; Ma et al., 2023), we study factual knowledge of the form (s, r, o), consisting of a subject s, a relation r, and an object o (e.g., s = Eiffel Tower, r = location, o = Paris). Besides, we also follow them to employ discrete prompts to test whether the knowledge is in a language model. Specifically, a natural language template $t_r(\cdot)$ is constructed for each relation r, which is then combined with a subject s as input to generate a prompt (question or cloze-style statement) $t_r(s)$. For instance, given the subject "Eiffel Tower" and the relation "location", we can form a cloze sentence "The Eiffel Tower is located in".

3.2. Knowledge Editing

The goal of knowledge editing is to incorporate new facts into model parameters without retraining. Previous works focus on altering the original knowledge stored in LLMs as $(s, r, o) \rightarrow (s, r, o^*)$, which only cares about the new target knowledge. However, the potential impact of the editing operation of appending new knowledge on its neighboring knowledge remains unclear. Therefore, this paper explores the scenarios where a subject has multiple corresponding objects under a given relation. In detail, appending knowledge aims at incorporating a new object to the set of original objects, retaining original objects, and not introducing false objects: $(s, r, \{o_1, o_2, ..., o_N\}) \rightarrow (s, r, \{o_1, o_2, ..., o_N, o^*\})$. In this paper, for a new piece of knowledge to append (s, r, o^*), its *neighboring knowledge* is defined as (s, r, O) where $O = \{o_1, o_2, ..., o_N\}$. Define an editing fact as $e = (s, r, O, o^*)$ and given an unedited model $\mathcal{F}$, the edited language model is gained by using a model editor K, $\mathcal{F}^* = K(\mathcal{F}, e)$.

To evaluate the performance of editing methods in appending knowledge, the previous three metrics are still utilized.

Efficacy validates whether the edited models could recall the appended fact under the sole editing prompt p. The assessment is based on Efficacy Score **(ES)** representing as: $\mathbb{1}[P_{\mathcal{F}^*}(o^* \mid p) > min\{P_{\mathcal{F}^*}(o \mid p) \mid o \in O\}]$[1].

Generalization verifies whether the edited models could recall the appending fact under the paraphrase prompts $\mathcal{P}^G$ via Generalization Score **(GS)**: $\mathbb{E}_{p \in \mathcal{P}^G}[\mathbb{1}[P_{\mathcal{F}^*}(o^* \mid p) > min\{P_{\mathcal{F}^*}(o \mid p) \mid o \in O\}]]$.

Locality verifies whether the output of the edited models for inputs out of editing scope remains unchanged under the locality prompts $\mathcal{P}^L$ via Locality Score **(LS)**: $\mathbb{E}_{p_l \in \mathcal{P}^L}[\mathbb{1}[P_{\mathcal{F}^*}(o_l \mid p_l) > P_{\mathcal{F}^*}(o^* \mid p_l)]]$, where o_l was the original answer of p_l. $\mathbb{1}$ is the indicator function.

For example, given an editing fact (s = Apple, r = products, O = {AirPods 3, MacBook Air,..., iPhone 14}, o^*=iPhone 15), the editing prompt p could be "What are the products of Apple?" and a paraphrase prompt could be "What items does Apple produce?" A locality prompt and its original answer could be "Which company developed Windows?" and "Microsoft" respectively.

4. Definition of Additivity

Given a subject s, a relation r, a composed question $t_r(s)$, original objects O, and an object to append o^*, it is unclear whether the original correct knowledge $K^c = (s, r, \{o_1, o_2, ..., o_N\})$ is still retained, and part of the false knowledge $K^f = (s, r, \{o_{f1}, o_{f2}, ..., o_{fM}\})$ is unintentionally included after editing, where $\{o_{f1}, o_{f2}, ..., o_{fM}\} \cap \{o_1, o_2, ..., o_N, o^*\} = \varnothing$. Therefore, a new metric termed *additivity* is designed to evaluate the degree of perturbation to neighboring knowledge, which is calculated in terms of *relative ranking* and *absolute probability change* of objects.

Relative ranking of objects We assume that the minimum probability of correct knowledge K^c should be larger than the maximum probability of false knowledge K^f before and after editing. Otherwise, there is a risk of forgetting original correct knowledge or introducing false knowledge.

[1]Previous works care about whether the probability surpasses the original o. Here O is a set, so we make a slight adaptation.

Intuitively, the maximum probability of false knowledge $P_f^{max} = \max_{i=1}^{M} P_{\mathcal{F}^*}(o_{fi}|t_r(s))$ is used as a threshold to calculate the proportion of the list of original correct objects of K^c whose probabilities are below this threshold. Mathematically, we have *Ranking Forgetting Factor* (RFF):

$$RFF = \frac{\sum_{i=1}^{N}[\mathbb{1}\{P_{\mathcal{F}^*}(o_i|t_r(s)) < P_f^{max}\} * \sigma(P_{\mathcal{F}^*}(o_i|t_r(s)))]}{\sum_{i=1}^{N}\sigma(P_{\mathcal{F}^*}(o_i|t_r(s)))}, \quad (1)$$

σ is the sigmoid function. Here, $\sigma(P)$ of each original correct knowledge is regarded as its weight, since knowledge with larger probability should weigh more and the sigmoid function can also be used for smoothing.

Similarly, the minimum probability of original correct knowledge $P_c^{min} = \min_{i=1}^{N} P_{\mathcal{F}^*}(o_i|t_r(s))$ is used as a threshold to calculate the proportion of the list of sampled false objects of K^f whose probabilities are above this threshold. Then, *Ranking Noise Factor* (RNF) is defined as:

$$RNF = \frac{\sum_{i=1}^{M}[\mathbb{1}\{P_{\mathcal{F}^*}(o_{fi}|t_r(s)) > P_c^{min}\} * \sigma(P_{\mathcal{F}^*}(o_{fi}|t_r(s)))]}{\sum_{i=1}^{M}\sigma(P_{\mathcal{F}^*}(o_{fi}|t_r(s)))}, \quad (2)$$

Absolute probability change of objects In addition to satisfying the assumption of relative ranking of objects, it is also necessary to characterize their absolute probability changes. Even if the relative ranking remains unchanged, substantial harm is inflicted upon the model if the absolute probability changes unexpectedly. Therefore, *Correct Probability Change* (CPC) is introduced to characterize this issue, which is defined as the ratio of the mean probability of correct knowledge after and before editing:

$$CPC = \frac{\sum_{i=1}^{N} P_{\mathcal{F}^*}(o_i|t_r(s))}{\sum_{i=1}^{N} P_{\mathcal{F}}(o_i|t_r(s))}. \quad (3)$$

Similarly, *False Probability Change* (FPC) is defined as the ratio of the mean probability of false knowledge after and before editing:

$$FPC = \frac{\sum_{i=1}^{M} P_{\mathcal{F}^*}(o_{fi}|t_r(s))}{\sum_{i=1}^{M} P_{\mathcal{F}}(o_{fi}|t_r(s))}. \quad (4)$$

Aggregation Finally, we aim to aggregate these two dimensions of relative ranking and absolute probability change into a unified metric, providing a more comprehensive representation of the detrimental impact induced by appending knowledge. The *Additive Forgetting Factor* (AFF) is defined as the degree to which the original correct knowledge is forgotten:

$$AFF = 1 - (1 - RFF) * min\{1, CPC\}. \quad (5)$$

This definition means that if the probability of correct knowledge does not decrease after editing ($CPC >= 1$), AFF equals RFF. Otherwise, the adverse effects of CPC and RFF would combine, resulting in AFF surpassing RFF. The AFF spans from 0 to 1, reflecting the extent to which the original correct knowledge is forgotten amidst neighboring perturbations. A higher AFF value corresponds to a more substantial negative impact. Similarly, the other aggregated metric *Additive Noising Factor* (ANF) is defined as the degree to which false knowledge is introduced:

$$ANF = 1 - (1 - RNF) * min\{1, \frac{1}{FPC}\}. \quad (6)$$

The ANF spans from 0 to 1, reflecting the extent to which the false knowledge is introduced in neighboring perturbations. A higher ANF value corresponds to a more substantial negative impact.

5. PEAK: Perturbation Evaluation of Appending Knowledge

This paper designs the PEAK benchmark to assess the extent of perturbation to neighboring knowledge during editing. It comprises two datasets of PEAK-CF and PEAK-T. The former is designed as a counterfact dataset for the evaluation of knowledge editing methods on counterfactual appending following Meng et al. (2022) and Ma et al. (2023). The latter is based on temporal knowledge edits of changes in the real-world following Zhong et al. (2023).

5.1. Data Construction of PEAK-CF

Aggregating facts This dataset is constructed based on Wikidata (Vrandecic & Krötzsch, 2014), a knowledge base containing millions of fact triples. First, we manually chose a total of 32 relations, where a subject has multiple corresponding objects under each relation. Then we collected all triples for each relation and classified triples of the same subject s and relation r together, denoted as $F_m(r) = \{(s, r, O) \mid O = \{o_1, o_2, ..., o_N\})\}$. For each relation r, ChatGPT (gpt-3.5-turbo) was used to generate some templates $T(r)$ expressing the same semantics. The subject s can be replaced to form the full prompts: $\mathcal{P}(s, r)$ = $\{t_r(s) \mid t_r(\cdot) \in T(r)\}$. For example, a template for (r = product) might be "{} has products like", where "Apple" substitutes "{}".

Constructing counterfactual edits Following previous works (Meng et al., 2022; Yao et al., 2023; Ma et al., 2023), counterfactual edits were built in this dataset. The previous three metrics are used in this dataset. Given a set of triples $(s, r, O) \in F_m(r)$ and an object o^*, where $o^* \notin O$ and (s, r, o^*) is a counterfact, an edit is represented as $\mathcal{E} = \{s, r, O, o^*, p\}$ to test *efficacy*, where the editing prompt $p \in \mathcal{P}(s, r)$. Besides, to test *generalization*, a set of two semantically-equivalent paraphrase prompts $\mathcal{P}^G$ are sampled from $\mathcal{P}(s, r) \setminus \{p\}$. Moreover, to test *local-*

Table 1. An example in the PEAK-T dataset. a and a^* in different rows represent the original answer and desired answer after editing respectively. $\cup$ refers to the set union operation.

$\mathcal{E}$	(s, r): (Olympic Winter Games, host country) O: {France, United States, ..., South Korea} o^*: China p: What are the host countries of Olympic Winter Games? (a: O, a^*: $O \cup \{o^*\}$)
$\mathcal{P}^G$	The host countries of Olympic Winter Games are Which nations have hosted the Winter Olympics? (a: O, a^*: $O \cup \{o^*\}$)
$\mathcal{P}^L$	Big Ben is located in (a & a^*: London) The headquarters of Apple is in (a & a^*: California)
O_h	{Japan, Shanghai, Hong Kong, ..., Russia}
O_r	{Uruguay, Mozambique, Fiji, ..., Tripoli}

ity, a set of triples outside the scope of editing is selected: $\mathcal{S} = \{(s', r', o')\}$. For example, select ($s$= Olympic Games, r= host city, o= Paris), $\mathcal{S}$ might contain triple like (s'= France, r'= shares border, o'= Germany). Then a set of prompts $\{\mathcal{P}(s', r') \mid (s', r', o') \in \mathcal{S}\}$ is constructed to sample the locality prompts $\mathcal{P}^L$.

Sampling false answers For the *additivity* proposed in this paper, given a factual question $t_r(s)$ (e.g., the editing prompt p), both the original correct answers and sampled false answers are utilized. O is the list of original correct answers, while the false answers were sampled in two settings *Hard* and *Random* (sampling details are in Appendix A.1). For the *Hard* setting, some objects that establish direct relations with the new object o^* are selected, represented as $O_h = \{o_{h1}, o_{h2}, ..., o_{hM}\}$, where $O_h \cap O = \varnothing$. Similarly, for the *Random* setting, some objects $O_r = \{o_{r1}, o_{r2}, ..., o_{rM}\}$ that are semantically distant from the new object are sampled. Intuitively, false answers in the *Hard* setting are more easily introduced into the model compared to answers in the *Random* setting.

Filtering correct and false answers Subsequently, according to additivity defined in Section 4, for $t_r(s)$, the minimum probability of correct answers O should be larger than the maximum probability of false answers O_h and O_r before and after editing. Therefore, before editing, we filtered out correct answers with probabilities below a certain threshold in the original model. Then false answers with probabilities greater than the minimum probability of the selected correct answers were also filtered out.

5.2. Data Construction of PEAK-T

This dataset focuses on temporal-based, real-world edits, where each edit is factually correct and occurs after the model is trained. Following Yin et al. (2023), it is built upon YAGO (Mahdisoltani et al., 2015), a knowledge base containing fact triples extracted from Wikipedia, enriched with WordNet, GeoNames, and other data sources, and contains the time when the facts occurred. We sampled facts in YAGO that occurred after 2021, which took place following the release of GPT-2 XL and GPT-J. Due to the limited number of relation types, 9 relations have been selected. Then the construction is similar to PEAK-CF.

Table 2. The statistics of PEAK-CF and PEAK-T datasets.

Type	PEAK-CF	PEAK-T
Editing examples	1,962	993
Correct answers	26,512	7,759
Paraphrase prompts	3,344	1,995
Neighborhood prompts	10,760	6,207
False answers (hard)	24,043	7,879
False answers (random)	19,616	9,929

5.3. Dataset Summary

Dataset Format As shown in Table 1, each instance in the PEAK benchmark is represented as a tuple ($\mathcal{E}$, $\mathcal{P}^G$, $\mathcal{P}^L$, O_h, O_r), where $\mathcal{E}$ comprises a piece of new knowledge and original correct knowledge. $\mathcal{P}^G$ and $\mathcal{P}^L$ are the prompts utilized to validate generalization and locality respectively. O_h and O_r represent the sampled false answer lists for the factual questions p and $\mathcal{P}^G$ under the *Hard* and *Random* settings respectively. a and a^* denote the original answer and desired answer after editing respectively.

Dataset Statistics Table 2 summarizes the statistics of the PEAK-CF and PEAK-T datasets. To verify generalization and locality, there is at least one prompt for each instance. Besides, each fact question has about an average of 10 correct and 20 false answers. Generally speaking, these two datasets contain counterfactual edits and temporal edits respectively, and are used to study neighborhood perturbations of edited models. Readers can refer to Appendix A.2 for the details of relations and prompts.

6. Appending via Preservation and Prevention

The neighboring perturbations in the process of appending knowledge may lead to the forgetting of original correct knowledge, as well as the unintentional inclusion of noise. A plug-and-play framework APP (**A**ppending via **P**reservation and **P**revention) is proposed to improve existing editing methods to mitigate this detriment in editing.

Given the new knowledge to append (s, r, o^*) and an editing prompt p, current editing methods usually define an editing objective $\mathcal{L}_e(o^*, \theta_{\mathcal{F}})$ to incorporate the new knowledge, where $\theta_{\mathcal{F}}$ denotes the parameters to be updated in the unedited model $\mathcal{F}$. APP designs a set of editing objectives which can be coupled with $\mathcal{L}_e(o^*, \theta_{\mathcal{F}})$, to minimize the probability perturbations of both neighboring correct and

Table 3. Evaluation results (%) of the PEAK-CF dataset. "hard" refers to the additivity calculated using false answers in the *Hard* setting, while "ran" refers to that in the *Random* setting. ↑ indicates that a higher value corresponds to better performance, while ↓ is the opposite. Numbers marked with † indicate statistically significant improvements in the method coupled with APP over the original method (t-test with p-value <0.05). Since KN has no loss to optimize, APP is not applied to it. Due to space limitation, the results of LLaMA-2 (13B) were put in Appendix B.4.

Editor	GPT-2 XL (1.5B)							LLaMA-2 (7B)						
	Previous			Additivity (hard)		Additivity (ran)		Previous			Additivity (hard)		Additivity (ran)	
	ES ↑	GS ↑	LS ↑	AFF ↓	ANF ↓	AFF ↓	ANF ↓	ES ↑	GS ↑	LS ↑	AFF ↓	ANF ↓	AFF ↓	ANF ↓
FT	90.99	64.68	87.71	78.86	59.04	69.29	37.12	98.27	87.72	68.48	74.51	60.62	64.22	38.72
KN	26.48	26.08	89.08	44.71	29.47	41.23	22.96	46.67	46.45	66.16	45.63	40.51	43.72	35.64
MEND	97.46	84.87	42.46	30.15	33.97	12.86	11.62	95.22	86.02	47.47	35.39	33.51	21.16	14.61
MEMIT	72.90	63.08	98.44	51.49	52.11	31.71	17.22	99.95	**99.11**	92.12	94.26	84.14	86.03	53.61
ROME	**99.27**	**94.05**	90.57	87.44	69.80	75.16	35.58	99.69	97.74	94.40	93.05	82.47	83.80	52.25
FT+APP	87.89	59.30	89.21	70.48†	49.73†	59.85†	29.76†	98.01	85.24	73.55†	68.44†	49.62†	57.93†	33.34†
MEND+APP	94.25	81.02	45.51†	**26.27**†	**28.81**†	**12.21**	**11.07**	92.86	82.46	51.43†	**32.55**†	29.73†	**20.44**	13.99
MEMIT+APP	69.75	59.26	**98.66**	39.89†	43.41†	22.14†	13.58†	99.42	96.51	94.81†	38.83†	**17.41**†	32.65†	**11.69**†
ROME+APP	96.15	87.17	93.27†	40.45†	31.60†	25.92†	12.61†	**100.00**	94.24	**96.22**†	43.11†	20.14†	36.86†	13.74†

incorrect knowledge. On the one hand, the editing objective $\mathcal{L}_1(O, O_h, \theta_{\mathcal{F}})$ is designed to maintain a certain margin between the probabilities of original correct answers O and those of false answers O_h for the question $t_r(s)$ via the Hinge Loss (Gentile & Warmuth, 1998; Wu & Liu, 2007) as follows:[2]

$$\mathcal{L}_1(O, O_h, \theta_{\mathcal{F}}) = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} max\{0,\ \mathcal{M} - \log P_{\mathcal{F}'}(o_i \,|\, p) + \log P_{\mathcal{F}'}(o_{hj} \,|\, p)\}, \tag{7}$$

where $\mathcal{F}'$ is the intermediate model of the editing process. N and M represent the number of elements in O and O_h respectively. This equation implies that the log probabilities of correct answers are encouraged to be larger than those of false answers by at least a certain margin $\mathcal{M}$.

On the other hand, it involves ensuring that the absolute probabilities of correct answers do not decrease while those of false answers do not increase during editing, which can be conceptualized as two objectives $\mathcal{L}_2(O, \theta_{\mathcal{F}})$ and $\mathcal{L}_3(O_h, \theta_{\mathcal{F}})$ respectively as:

$$\mathcal{L}_2(O, \theta_{\mathcal{F}}) = \frac{1}{N}\sum_{i=1}^{N} max\{0, \log P_{\mathcal{F}}(o_i \,|\, p) - \log P_{\mathcal{F}'}(o_i \,|\, p)\}, \tag{8}$$

$$\mathcal{L}_3(O_h, \theta_{\mathcal{F}}) = \frac{1}{M}\sum_{i=1}^{M} max\{0, \log P_{\mathcal{F}'}(o_{hi} \,|\, p) - \log P_{\mathcal{F}}(o_{hi} \,|\, p)\}, \tag{9}$$

$\mathcal{L}_2(O, \theta_{\mathcal{F}})$ means that if the probability of the correct answer decreases during editing, the loss is equal to the log probability decrease value, otherwise it is 0. $\mathcal{L}_3(O_h, \theta_{\mathcal{F}})$ is designed similarly. Finally, these proposed objectives are jointly optimized with the original editing objective in each method $\mathcal{L}_e(o^*, \theta_{\mathcal{F}})$ as:

$$\mathcal{L} = \min_{\theta_{\mathcal{F}}} \mathcal{L}_e(o^*, \theta_{\mathcal{F}}) + \alpha\mathcal{L}_1(O, O_h, \theta_{\mathcal{F}}) + \beta\mathcal{L}_2(O, \theta_{\mathcal{F}}) + \gamma\mathcal{L}_3(O_h, \theta_{\mathcal{F}}). \tag{10}$$

Here $\alpha, \beta, \gamma >= 0$ are hyperparameters. Appendix C provides an example illustrating how the proposed APP method is coupled with a selected editing method for clarity.

7. Experiments

In this section, experiments were conducted to demonstrate the neighborhood perturbations of existing editing methods based on four representative LLMs. Furthermore, the proposed APP method was coupled with these editing methods to mitigate the neighborhood perturbations. Finally, comprehensive analyses were carried out to further verify the effectiveness of the proposed framework.

7.1. Base LLMs and Editing Methods

Four popular LLMs were used for experiments. Considering limited computing resources, the PEAK-CF dataset was conducted on **GPT-2 XL** (1.5B) (Radford et al., 2019), **LLaMA-2** (7B) and **LLaMA-2** (13B) (Touvron et al., 2023). Since there is little data in YAGO after LLaMA-2 release date (2023-7), The PEAK-T was conducted on **GPT-2 XL** and **GPT-J** (6B) (Wang & Komatsuzaki, 2021). Five popular knowledge editing methods were selected as the baselines including **FT** (Zhu et al., 2020), **KN** (Dai et al., 2022), **MEND** (Mitchell et al., 2022a), **ROME** (Meng et al., 2022), and **MEMIT** (Meng et al., 2023). Readers can refer to Appendix B.1 for the details of these editing methods.

[2] O_h is adopted due to hard negative sampling.

Table 4. Evaluation results (%) of the PEAK-T dataset.

Editor	GPT-2 XL (1.5B)							GPT-J (6B)						
	Previous			Additivity (hard)		Additivity (ran)		Previous			Additivity (hard)		Additivity (ran)	
	ES ↑	GS ↑	LS ↑	AFF ↓	ANF ↓	AFF ↓	ANF ↓	ES ↑	GS ↑	LS ↑	AFF ↓	ANF ↓	AFF ↓	ANF ↓
FT	83.31	59.89	61.67	80.44	38.10	72.24	18.76	95.21	80.68	42.74	87.25	40.14	77.09	14.97
KN	28.39	27.38	83.81	50.85	24.41	47.29	18.73	45.49	43.11	67.05	93.84	46.47	93.29	40.23
MEND	99.00	89.90	8.78	24.02	19.78	12.91	8.18	99.72	87.55	10.05	24.73	16.82	13.66	5.12
MEMIT	57.92	46.46	97.71	25.92	31.45	7.31	4.19	100.00	85.42	94.00	50.18	36.35	22.85	2.95
ROME	**100.00**	86.77	92.95	57.65	33.84	36.05	5.35	100.00	**92.57**	89.15	60.48	38.66	33.87	4.02
FT+APP	79.03	53.19	67.29†	75.24†	31.64†	67.47†	17.75†	90.85	74.50	49.53†	82.75†	31.06†	75.02†	13.41†
MEND+APP	99.29	**90.68**	8.37	26.27	28.81	12.21	11.07	99.72	87.49	9.93	26.38	17.05	14.85	5.12
MEMIT+APP	59.06	47.68	**97.92**	**23.30**†	29.00†	**6.86**	4.20	100.00	83.23	**94.36**	**21.40**†	**16.78**†	**7.54**†	**1.93**†
ROME+APP	99.86	86.70	93.65	28.81†	**14.84**†	16.79†	**3.16**†	**100.00**	90.83	89.90	31.56†	17.15†	16.17†	2.17†

7.2. Evaluation Metrics

Three basic metrics of *Efficacy* (ES), *Generalization* (GS) and *Locality* (LS) (Meng et al., 2022; 2023) defined in Section 3.2 were still adopted to measure each editing method. Besides, the proposed *Additivity* was to evaluate the degree of perturbation to neighboring knowledge when appending knowledge. **AFF** and **ANF** designed in Section 4 were employed to quantify the extent of forgetting original correct answers and the inclusion of noise and incorrect answers respectively. When assessing the additivity, both the editing and paraphrase prompts $\{p\} \cup \mathcal{P}^G$ were utilized without loss of generality. For all the above metrics, the average results across all edits in each dataset were reported.

7.3. Results of Existing Editing Methods

The top five rows of Table 3 and Table 4 reported the knowledge editing results of current editing methods on PEAK-CF and PEAK-T respectively. These results were analyzed from the following perspectives.

The performance of editing the new target knowledge. Except for KN, editing methods performed well in efficacy (ES) and generalization (GS), showing that most of existing editing methods are capable of effectively appending new target knowledge. For locality, locate-then-edit methods (KN, ROME, MEMIT) significantly outperformed other methods, demonstrating that they have little interference with irrelevant knowledge. Moreover, as the model size increased, the performance of a particular editing method exhibited continuous improvement in appending new facts.

The perturbations of editing on neighboring knowledge. We were surprised to observe that existing editing methods significantly perturbed the knowledge neighboring to the target knowledge in LLMs after editing, compromising the integrity of original correct knowledge and introducing unintentional noise. Taking LLaMA-2 edited by ROME on PEAK-CF as an example, despite its superior performance evaluated by previous metrics, it achieved remarkably poor performance of 93.05% AFF and 82.47% ANF respectively in the *Hard* negative setting. Even in the *Random* setting, they still show poor performance, indicating severe damage to neighboring knowledge. Meanwhile, edited models generally showed worse performance under the *Hard* setting than those under the *Random* setting in terms of AFF and ANF, demonstrating that the closer the semantic relationship between the false answer and the newly appended answer, the more susceptible it is to perturbation. Remarkably, while MEND performed the best in terms of additivity, its relatively low LS indicates a significant disruption to irrelevant knowledge. Our findings suggest that current editing methods primarily care about the editing performance of new target knowledge, while neglecting the negative perturbations on its neighboring knowledge.

Comparison between datasets. Comparing the results of each method on the two datasets, two conclusions were drawn. (1) The ES and GS of each method on PEAK-T were lower than those on PEAK-CF, showing that the PEAK-T is more challenging for existing methods to append knowledge without neighboring perturbations. This could be attributed to the fact that, for PEAK-T, new knowledge originates from the external world and has never been seen by the model. In contrast, PEAK-CF already contains some weak counterfactual associations, making it easier for the model to integrate counterfacts. (2) Both AFF and ANF on PEAK-T were lower than those on PEAK-CF, suggesting that PEAK-T suffers fewer neighboring perturbations during editing than PEAK-CF. This is possibly due to the weaker correlation between the sampled false answers and the new external knowledge compared to the original counterfactual knowledge in the model. Consequently, the perturbations when appending new knowledge in PEAK-T are smaller.

7.4. Results of APP

As shown in the bottom four rows in Table 3 and Table 4, APP was coupled with four editing methods. The false answers utilized in APP were from the *Hard* setting. De-

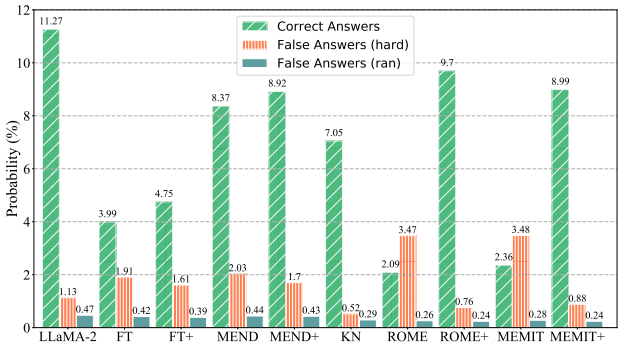

Figure 2. Average probability of the correct answers O and the false answers O_h (*Hard*) and O_r (*Random*) of LLaMA-2 after editing with different editing methods. LLaMA-2 refers to the unedited model. "+" means this method was coupled with APP.

tailed hyperparameters of each method can be referred to in Appendix B.2. In general, APP almost maintained the performance of appending new knowledge in terms of previous editing metrics, and greatly mitigated the neighborhood perturbations in terms of AFF and ANF under both the *Hard* and *Random* settings. Particularly, ROME+APP and MEMIT+APP still performed well in appending new knowledge, with the most substantial reduction in perturbations compared to the original editing methods. Besides, we also found that APP has also improved the performance of LS, preserving irrelevant knowledge. These results help conclude that APP effectively preserves the integrity of original correct knowledge and prevents the inclusion of false knowledge while appending new knowledge. Despite the notable improvement in additivity, it remains considerably below a satisfactory level, highlighting the severity and complexity of the proposed neighboring perturbations. Addressing this challenge will necessitate collaborative efforts from the community.

7.5. Probability Change of Answers

To further analyze the neighboring perturbations in edited models and the mechanism of our proposed APP, the average probability of both correct and false answers in LLaMA-2 models edited by different methods on PEAK-CF, is illustrated in Figure 2. Two conclusions can be drawn here.

Existing editing methods severely perturb probabilities. Compared with the original LLaMA-2, the probabilities of correct answers dropped significantly, while the probabilities of false answers (*Hard*) increased a lot, especially for MEMIT and ROME. MEND has shown minimal disturbance, aligning with its superior additivity performance. Compared to the false answers (*Hard*), the probabilities of false answers (*Random*) undergo a notably smaller shift after editing, indicating that the false answers more closely tied to the new appended answers warrant greater attention.

APP effectively mitigates the probability perturbations.

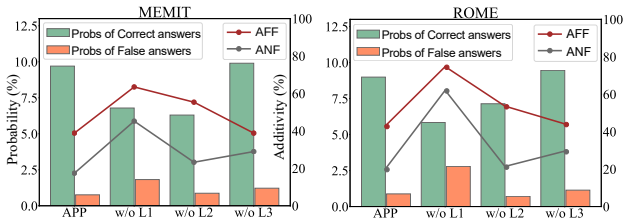

Figure 3. Ablation analysis of probability and additivity for APP. Results were conducted with LLaMA-2 on PEAK-CF dataset. Due to page limit, results for other methods are put in Appendix B.5.

After coupling existing editing methods with the proposed APP, the probability perturbations for both correct and false answers were significantly mitigated. These results vividly explain why APP can effectively mitigate perturbations, thereby preserving the correct knowledge and preventing the inclusion of false knowledge.

7.6. Ablation study

Ablation tests of removing each editing objective $\mathcal{L}_1$, $\mathcal{L}_2$ or $\mathcal{L}_3$ in APP were conducted to validate their effectiveness. The AFF and ANF of additivity, as well as the probabilities of correct and false (*Hard*) answers were illustrated in Figure 3. We have two findings. First, removing any editing objective of the APP led to performance degradation in terms of additivity and probability perturbations, showing their effectiveness. Second, removing $\mathcal{L}_1$ resulted in the most significant performance degradation. It demonstrated the most importance of $\mathcal{L}_1$ in APP, which is designed to maintain a certain margin between the probabilities of correct answers and those of false answers.

7.7. Effect of Access to All Correct Answers

The proposed method has access to all the correct answers O during editing, whereas the baseline methods can only have access to the new correct answer o^*. To explore the effect of access to all the correct answers, we let these baseline methods also directly use all correct answers as editing targets when editing. Specifically, all baseline methods have been extended to utilize the entire set O when editing LLMs to ensure a fair comparison. The results are shown in Table 5. Comparing these results with those of the proposed APP in Table 3, we observe that, Despite similar performance in appending new knowledge, the additivity of the new baselines is still significantly higher than that of the proposed APP method, suggesting that APP performs better at maintaining the integrity of the answer list.

7.8. Effect of Number of Neighboring Answers

Figure 4 illustrates how the performance of APP changed with respect to different numbers of neighboring answers on PEAK-CF with LLaMA-2 edited by different meth-

Table 5. Evaluation results (%) of the baselines using LLaMA-2 (7B) to access all correct answers on the PEAK-CF dataset. "+" refers to the baseline coupled with all correct answers O. Here only false answers in the *Hard* setting were used.

Editor	LLaMA-2 (7B)				
	Previous			Additivity (hard)	
	ES ↑	GS ↑	LS ↑	AFF ↓	ANF ↓
FT+	97.54	85.93	71.68	74.35	61.83
MEND+	93.26	79.90	50.91	33.92	32.03
MEMIT+	98.29	93.07	94.20	48.10	66.72
ROME+	100.00	93.13	95.48	50.27	64.91

ods. To this end, we extended our evaluation and considered utilizing k correct and false answers in APP, where $k \in [0, 1, 3, 5, all]$. It can be seen from these results that the performance of all editing methods coupled with APP was significantly improved as the number of neighboring answers increased. This trend shows the effectiveness of considering more neighboring answers to comprehensively characterize the neighboring perturbations. It also shows that both AFF and ANF can be significantly improved even with fewer answers, showcasing the practicality of the proposed APP method.

8. Conclusion & Limitation

This study investigates the neighboring perturbations of knowledge editing on LLMs. A metric of *additivity* is introduced and a benchmark dubbed as PEAK is constructed for assessing the degree of perturbations in neighboring knowledge. A plug-and-play framework APP is proposed to mitigate perturbations by minimizing the probability disruptions during knowledge appending. Comprehensive experiments on various knowledge editing methods and LLMs reveal that they inevitably perturb neighboring knowledge during editing, and the proposed APP method demonstrates its effectiveness in mitigating this perturbations to a certain extent. In the future, we will explore expanding the scope of neighbor knowledge to more comprehensively characterize the neighboring perturbations of knowledge editing.

Besides, there are several limitations to our work. First, regarding the experimental settings, the editing methods employed in this paper involve modifying the model parameters. But some editing methods that preserve model parameters remain to be explored. Second, this paper focuses on factual knowledge, and it would be beneficial to extend this approach to different types of knowledge, such as commonsense, logical, or spatial knowledge. Third, this paper designs the proposed additivity metric based on probability. However, the results generated by the actual model do not entirely depend on probability. Therefore, while this evaluation roughly reflects the degree of perturbation, it may deviate from real-world scenarios to some extent.

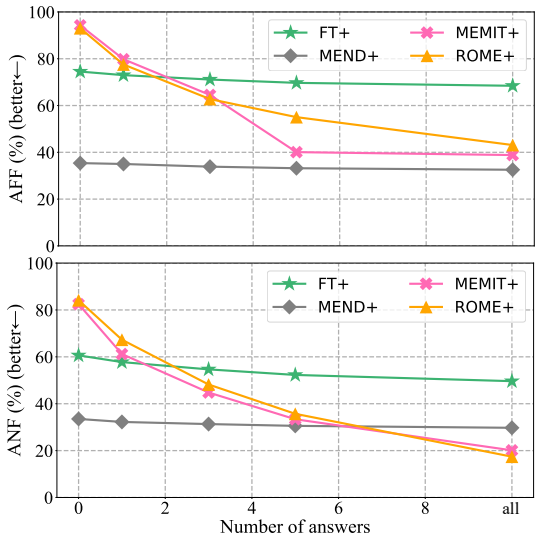

Figure 4. Both AFF and ANF in *Hard setting* on PEAK-CF with LLaMA-2 across four editing methods equipped with APP where k original correct and false answers were used. $k \in [0, 1, 3, 5, all]$.

Acknowledgements

We would like to express gratitude to the anonymous reviewers for their kind comments. This work is funded by the National Science and Technology Major Project (No. 2023ZD0121103). Ningyu Zhang is supported by the National Natural Science Foundation of China (No. 62206246), Zhejiang Provincial Natural Science Foundation of China (No. LGG22F030011), Yongjiang Talent Introduction Programme (2021A-156-G), CCF-Tencent Rhino-Bird Open Research Fund.

Impact Statement

The research presented in this paper focuses on a critical aspect of LLMs – their susceptibility to unintended text generation due to false or outdated knowledge. In light of the resource-intensive nature of retraining LLMs, the study focuses on the emerging field of knowledge editing. It develops technologies to reduce perturbations to preserve factuality. In terms of future societal consequences, this work contributes to mitigating the need for full retraining, reducing computational resources and energy consumption. This aligns with sustainable AI and makes it more feasible to deploy LLMs in resource-constrained settings. Besides, APP's ability to perform targeted edits and mitigate neighboring perturbations can enhance LLMs' effectiveness in specialized domains, such as medical diagnostics, and legal analysis, where precise and updated knowledge is essential. Additionally, editing methods for appending knowledge to LLMs could be misused to propagate misinformation. Furthermore, it could pose privacy risks if sensitive information is appended without proper safeguards. In the future, some strategies should be implemented to reduce these risks.

References

Bau, D., Liu, S., Wang, T., Zhu, J., and Torralba, A. Rewriting a deep generative model. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J. (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pp. 351–369. Springer, 2020. doi: 10.1007/978-3-030-58452-8_21. URL https://doi.org/10.1007/978-3-030-58452-8_21.

Cao, N. D., Aziz, W., and Titov, I. Editing factual knowledge in language models. In Moens, M., Huang, X., Specia, L., and Yih, S. W. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 6491–6506. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.522. URL https://doi.org/10.18653/v1/2021.emnlp-main.522.

Cheng, S., Tian, B., Liu, Q., Chen, X., Wang, Y., Chen, H., and Zhang, N. Can we edit multimodal large language models? In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 13877–13888. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.856.

Cohen, R., Biran, E., Yoran, O., Globerson, A., and Geva, M. Evaluating the ripple effects of knowledge editing in language models. *CoRR*, abs/2307.12976, 2023. doi: 10.48550/ARXIV.2307.12976. URL https://doi.org/10.48550/arXiv.2307.12976.

Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. Knowledge neurons in pretrained transformers. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 8493–8502. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.581. URL https://doi.org/10.18653/v1/2022.acl-long.581.

Gandikota, R., Materzynska, J., Fiotto-Kaufman, J., and Bau, D. Erasing concepts from diffusion models. *CoRR*, abs/2303.07345, 2023. doi: 10.48550/ARXIV.2303.07345. URL https://doi.org/10.48550/arXiv.2303.07345.

Gentile, C. and Warmuth, M. K. Linear hinge loss and average margin. In Kearns, M. J., Solla, S. A., and Cohn, D. A. (eds.), *Advances in Neural Information Processing Systems 11, [NIPS Conference, Denver, Colorado, USA, November 30 - December 5, 1998]*, pp. 225–231. The MIT Press, 1998.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. In Moens, M., Huang, X., Specia, L., and Yih, S. W. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 5484–5495. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.446. URL https://doi.org/10.18653/v1/2021.emnlp-main.446.

Geva, M., Caciularu, A., Wang, K. R., and Goldberg, Y. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 30–45. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.3. URL https://doi.org/10.18653/v1/2022.emnlp-main.3.

Gu, J., Xu, H., Ma, J., Lu, P., Ling, Z., Chang, K., and Peng, N. Model editing can hurt general abilities of large language models. *CoRR*, abs/2401.04700, 2024. doi: 10.48550/ARXIV.2401.04700. URL https://doi.org/10.48550/arXiv.2401.04700.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., and Liu, T. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *CoRR*, abs/2311.05232, 2023. doi: 10.48550/ARXIV.2311.05232. URL https://doi.org/10.48550/arXiv.2311.05232.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A., and Fung, P. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38, 2023. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

Li, Z., Zhang, N., Yao, Y., Wang, M., Chen, X., and Chen, H. Unveiling the pitfalls of knowledge editing for large language models. *CoRR*, abs/2310.02129, 2023. doi: 10.48550/ARXIV.2310.02129. URL https://doi.org/10.48550/arXiv.2310.02129.

Ma, J., Chen, B., Gu, J., Ling, Z., Guo, W., Liu, Q., Chen, Z., and Liu, C. Wider & closer: Mixture of short-channel distillers for zero-shot cross-lingual named entity recognition. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference*

on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pp. 5171–5183. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.345. URL https://doi.org/10.18653/v1/2022.emnlp-main.345.

Ma, J., Gu, J., Ling, Z., Liu, Q., and Liu, C. Untying the reversal curse via bidirectional language model editing. *CoRR*, abs/2310.10322, 2023. doi: 10.48550/ARXIV.2310.10322. URL https://doi.org/10.48550/arXiv.2310.10322.

Mahdisoltani, F., Biega, J., and Suchanek, F. M. YAGO3: A knowledge base from multilingual wikipedias. In *Seventh Biennial Conference on Innovative Data Systems Research, CIDR 2015, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*. www.cidrdb.org, 2015. URL http://cidrdb.org/cidr2015/Papers/CIDR15_Paper1.pdf.

Mao, S., Zhang, N., Wang, X., Wang, M., Yao, Y., Jiang, Y., Xie, P., Huang, F., and Chen, H. Editing personality for llms. *CoRR*, abs/2310.02168, 2023. doi: 10.48550/ARXIV.2310.02168. URL https://doi.org/10.48550/arXiv.2310.02168.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in GPT. In *NeurIPS*, 2022. URL https://arxiv.org/abs/2202.05262.

Meng, K., Sharma, A. S., Andonian, A. J., Belinkov, Y., and Bau, D. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=MkbcAHIYgyS.

Mitchell, E., Lin, C., Bosselut, A., Finn, C., and Manning, C. D. Fast model editing at scale. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022a. URL https://openreview.net/forum?id=0DcZxeWfOPt.

Mitchell, E., Lin, C., Bosselut, A., Manning, C. D., and Finn, C. Memory-based model editing at scale. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15817–15831. PMLR, 2022b. URL https://proceedings.mlr.press/v162/mitchell22a.html.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. URL https://arxiv.org/abs/2203.02155.

Pearl, J. Direct and indirect effects. In Breese, J. S. and Koller, D. (eds.), *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, University of Washington, Seattle, Washington, USA, August 2-5, 2001*, pp. 411–420. Morgan Kaufmann, 2001. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=126&proceeding_id=17.

Peng, B., Galley, M., He, P., Cheng, H., Xie, Y., Hu, Y., Huang, Q., Liden, L., Yu, Z., Chen, W., and Gao, J. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *CoRR*, abs/2302.12813, 2023. doi: 10.48550/arXiv.2302.12813. URL https://doi.org/10.48550/arXiv.2302.12813.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sinitsin, A., Plokhotnyuk, V., Pyrkin, D. V., Popov, S., and Babenko, A. Editable neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJedXaEtvS.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Canton-Ferrer, C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M., Lavril, T., Lee, J., Liskovich, D., Lu, Y., et al. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/arXiv.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., and Shieber, S. M. Investigating gender bias in language models using causal mediation analysis. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Informa-*

tion Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

Vrandecic, D. and Krötzsch, M. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, 2014. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

Wang, B. and Komatsuzaki, A. Gpt-j-6b: A 6 billion parameter autoregressive language model, 2021.

Wang, P., Zhang, N., Xie, X., Yao, Y., Tian, B., Wang, M., Xi, Z., Cheng, S., Liu, K., Zheng, G., and Chen, H. Easyedit: An easy-to-use knowledge editing framework for large language models. *CoRR*, abs/2308.07269, 2023. doi: 10.48550/arXiv.2308.07269. URL https://doi.org/10.48550/arXiv.2308.07269.

Wang, Y., Chen, M., Peng, N., and Chang, K.-W. Deepedit: Knowledge editing as decoding with constraints. *arXiv preprint arXiv:2401.10471*, 2024.

Wu, X., Li, J., Xu, M., Dong, W., Wu, S., Bian, C., and Xiong, D. DEPN: detecting and editing privacy neurons in pretrained language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 2875–2886. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.174.

Wu, Y. and Liu, Y. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479):974–983, 2007.

Yao, Y., Wang, P., Tian, B., Cheng, S., Li, Z., Deng, S., Chen, H., and Zhang, N. Editing large language models: Problems, methods, and opportunities. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 10222–10240. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.632.

Yin, X., Jiang, J., Yang, L., and Wan, X. History matters: Temporal knowledge editing in large language model. *CoRR*, abs/2312.05497, 2023. doi: 10.48550/ARXIV.2312.05497. URL https://doi.org/10.48550/arXiv.2312.05497.

Zhang, B., Haddow, B., and Birch, A. Prompting large language model for machine translation: A case study. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 41092–41110. PMLR, 2023a. URL https://proceedings.mlr.press/v202/zhang23m.html.

Zhang, N., Yao, Y., Tian, B., Wang, P., Deng, S., Wang, M., Xi, Z., Mao, S., Zhang, J., Ni, Y., et al. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*, 2024.

Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., Wang, L., Luu, A. T., Bi, W., Shi, F., and Shi, S. Siren's song in the AI ocean: A survey on hallucination in large language models. *CoRR*, abs/2309.01219, 2023b. doi: 10.48550/arXiv.2309.01219. URL https://doi.org/10.48550/arXiv.2309.01219.

Zhong, Z., Wu, Z., Manning, C. D., Potts, C., and Chen, D. Mquake: Assessing knowledge editing in language models via multi-hop questions. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 15686–15702. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.emnlp-main.971.

Zhu, C., Rawat, A. S., Zaheer, M., Bhojanapalli, S., Li, D., Yu, F. X., and Kumar, S. Modifying memories in transformer models. *CoRR*, abs/2012.00363, 2020. URL https://arxiv.org/abs/2012.00363.

A. Datasets

A.1. Construction of False Answers

For the *additivity* proposed in this paper, given an edit $\mathcal{E} = \{s, r, O, o^*, p\}$ where $t_r(s) = p$, both the original correct answers and sampled false answers are utilized. The false answers were sampled in two settings *Hard* and *Random*. For the *Hard* setting, some objects establish direct relations with the new object o^* are selected. Specifically, denote the triples of Wikidata as $W(r) = \{(s, r, o)\}$, we firstly seek entities in $W(r)$ which have a link r' with o^* (could be the form of $(o^*, r', *)$, $(*, r', o^*)$), then we collect them and choose the entities has same entity type with o^*. For example, if o^* is a person, then the selected entities are persons, too. Finally, we sample some selected entities as the false answers. For the *Random* setting, we seek entities do not have links with o^*, while other parts are the same. In this way, the false answers in the *Hard* setting are more semantically close to the new appended object, while false answers in the *Random* setting are semantically distant from the new object.

A.2. Relations and Templates of Datasets

Table 6 and 7 show some examples of relations and their templates.

Table 6. Part of templates $T(r)$ for relation r in PEAK-CF. Actually, there are several templates for each relation. Here we only display one or two templates for each relation.

Relation (r)	$T(r)$
producer	"{} is the producer of what products?", "{} has produced many products such as'
official language	"{}, which is the official language of", "{} has been the official language of"
illustration	"The illustrator {} has created illustrations"
written	"{}, who has written some books like", "Which books have been written by {}?"

Table 7. Part of templates $T(r)$ for relation r in PEAK-T.

Relation (r)	$T(r)$
plays for	"For so long, {} has been at which clubs?", " What clubs have hired {} as a player?"
created	"{} was the individual responsible for creating"
has won prize	"{} was the recipient of the prize","What prizes have been gained by {}?"

B. Experimental Setup

B.1. Baseline Methods

Five popular model editing methods were selected as baselines including:

- **FT** (Zhu et al., 2020): this method simply performed gradient descent on the edits to update model parameters. It fine-tuned one layer in the model with a norm constraint on weight changes to prevent overfitting. Since the original authors did not publish their code, we followed Meng et al. (2022) re-implementation in their study.

- **KN** (Dai et al., 2022)[3]: it first selected neurons that were associated with knowledge expression via gradient-based attributions, and then modified MLP layer at the rows corresponding to those neurons by adding scaled embedding vectors.

- **MEND** (Mitchell et al., 2022a)[4]: it learned a hypernetwork to produce weight updates by decomposing the fine-tuning gradients into rank-1 form.

[3]https://github.com/EleutherAI/knowledge-neurons.
[4]https://github.com/eric-mitchell/mend

- **ROME** (Meng et al., 2022)[5]: it first localized the factual knowledge at a specific layer in the transformer MLP modules, and then updated the knowledge by directly writing new key-value pairs in the MLP module.
- **MEMIT** (Meng et al., 2023)[6]: it extended ROME to edit a large set of facts and updated a sequence of MLP layers to update knowledge.

The ability of these methods were assessed based on EasyEdit (Wang et al., 2023), an easy-to-use knowledge editing framework which integrates the released codes and hyperparameters from previous methods.

B.2. Hyperparameters for APP

To set the hyperparameters, we additionally created a small validation set. Table 8 shows the details of hyperparameters set for different LLMs. Besides, the margin $\mathcal{M}$ is set to 2.

Table 8. The hyperparameters for APP on each method on each model.

Editor	GPT-2 XL (1.5B)			GPT-J (6B)			LLaMA-2 (7B)		
	α	β	γ	α	β	γ	α	β	γ
FT+APP	0.2	0.5	0.2	0.2	0.5	0.2	0.2	0.5	0.2
MEND+APP	0.5	0	0	0.5	0	0	0.3	0	0
MEMIT+APP	0.05	0.05	0.05	0.1	0.3	0.2	0.6	1.0	0.3
ROME+APP	0.2	0.2	0.1	0.2	0.2	0.1	0.5	0.8	0.1

B.3. Examples of previous metrics

Here we provide an example to show what each previous metric means.

Efficacy Given an editing fact (s = Apple, r = products, O = {AirPods 3, MacBook Air,..., iPhone 14}, o^*=iPhone 15), if the edited model $\mathcal{F}^*$ assigns a higher probability to the new object o^* than the minimum probability in O[7] under the corresponding question $t_r(s)$ "What are the products of Apple?", then this new knowledge is appended effectively.

Generalization The edited model is considered to have generalized successfully if it can recall the new knowledge with paraphrased prompt like "What items does Apple produce?". That is, to determine whether the probability assigned by the edited model $\mathcal{F}^*$ to the new object o^* is higher than the minimum probability in O under the paraphrased prompt.

Locality The edited model should remain unchanged in response to prompts that are irrelevant or outside the scope of editing. For example, the answer to the question "Which company developed Windows?" should still be "Microsoft".

B.4. Results of LLaMA-2 (13B)

The results of LLaMA-2 (13B) are shown in Table 9. Compared with previous results, we can conclude that previous important conclusions are still valid, such as "The performance of memorizing the new target knowledge is good", "larger LLMs suffer more serious perturbation" and "APP significantly mitigates the perturbation".

B.5. Ablation study

Figure 5 shows other results of the ablation study in Section 7.6. As MEND could only use $\mathcal{L}_1$ in APP, we only list the results of FT.

C. Examples of how APP is coupled with existing methods

Here we take the editing method ROME as an example.

[5]https://github.com/kmeng01/rome
[6]https://github.com/kmeng01/memit
[7]Previous works care about whether the probability surpasses the original o. Here O is a set, so we made slight changes.

Table 9. Evaluation results (%) on the PEAK-CF dataset with LLaMA-2 (13B).

Editor	LLaMA-2 (13B)						
	Previous			Additivity (hard)		Additivity (ran)	
	ES ↑	GS ↑	LS ↑	AFF ↓	ANF ↓	AFF ↓	ANF ↓
FT	99.66	84.46	78.16	81.30	57.86	73.45	37.65
MEMIT	99.66	98.47	93.50	94.00	85.92	86.10	56.18
ROME	100.00	97.90	92.91	93.09	84.93	83.46	54.29
FT+APP	97.61	81.07	80.75	71.04	47.30	61.22	32.19
MEMIT+APP	97.95	92.26	96.13	51.03	26.20	44.92	20.19
ROME+APP	96.59	91.41	94.97	42.08	23.09	35.25	17.26

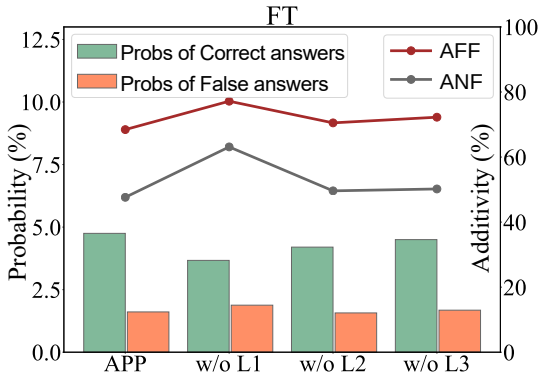

Figure 5. Ablation analysis of probability and additivity for APP. Results were conducted with LLaMA-2 on PEAK-CF dataset.

C.1. Rank-One Model Editing (ROME)

ROME (Meng et al., 2022) applies causal mediation analysis (Pearl, 2001; Vig et al., 2020) to locate the MLP modules that store facts. An MLP module consists of two layers, where the first and second layers are denoted as W_{fc}^l and W_{proj}^l respectively. W_{proj}^l is considered as a linear associative memory and an editing area. A new fact is represented as a key-value pair (k_*, v_*) (Geva et al., 2021), which can be inserted into the MLP module of the model by solving a constrained least-squares problem (Bau et al., 2020). All that remains is to choose the appropriate k_* and v_*.

Denote L as an edited MLP layer, $\mathcal{F}$ as the unedited model, and $e = (s, r, o^*)$ as the editing fact. The key k_* is obtained by calculating the average representation of the **last** token of the subject s outputted by W_{fc}^L as follows:

$$\begin{aligned} k_* &= \frac{1}{N} \sum_{j=1}^{N} k\left(x_j + s\right), \\ \text{where } k(x) &= \sigma\left(W_{fc}^{(L)} \gamma\left(a_{[x],i}^{(L)} + h_{[x],i}^{(L-1)}\right)\right). \end{aligned} \tag{11}$$

Here, x_j is the random sampled sequence and i is the location of the last token of the subject in sentence x. h is the hidden states, a is the attention, γ is the layernorm, and σ is the activation function.

The value v_* encodes the new knowledge (r, o^*) as a property of s. To calculate it, ROME sets $v_* = \operatorname{argmin}_z \mathcal{L}(z)$, where the objective is denoted as:

$$\mathcal{L}_e(o^*, z) = -\log P_{\mathcal{F}\left(m_i^{(L)}:=z\right)}\left[o^* \mid p\right] + D_{\mathrm{KL}}\left(P_{\mathcal{F}\left(m_{i'}^{(L)}:=z\right)}\left[x \mid p'\right] \| P_{\mathcal{F}}\left[x \mid p'\right]\right). \tag{12}$$

The first term in Eq. (12) seeks a vector z that, when substituted as the output of the MLP **at the token** i (notated $\mathcal{F}\left(m_i^{(L)} := z\right)$ and here i is the end of the subject in prompt p), will cause the network to predict the target object o^* in response to the factual prompt p (e.g., p=Eiffel Tower is located in, o^*=London). The second term minimizes the KL divergence (Ma et al., 2022) of predictions for the prompt p' (of the form " { subject } is a") to the unchanged model, which helps preserve the model's understanding of the subject's essence (i' is the location of the last token of the subject in p').

Finally, the MLP weight W_{proj}^{L} is updated with a rank-one update to insert the new fact:

$$\hat{W} = W + \Lambda \left(C^{-1} k_*\right)^T, \tag{13}$$

where C is a constant by estimating the covariance of many existing keys. $\Lambda = (v_* - W k_*) \,/\, \left(C^{-1} k_*\right)^T k_*$, representing the residual error of the new key–value pair on the original memory matrix.

Readers can refer to Meng et al. (2022) for more details of ROME.

C.2. Apply APP to ROME

In ROME, the editing objective is $\mathcal{L}_e(o^*, z)$, and the parameter to be optimized is z for $v_* = \operatorname{argmin}_z \mathcal{L}(z)$. Therefore, the proposed APP could be coupled with $\mathcal{L}_e(o^*, z)$ to optimize z.

On the one hand, the editing objective $\mathcal{L}_1(O, O_h, z)$ designed to maintain a certain margin between the probabilities of original correct answers O and those of false answers O_h could be written as follows:

$$\mathcal{L}_1(O, O_h, z) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} max\{0,\ \mathcal{M} - \log P_{\mathcal{F}(m_i^{(L)}:=z)}(o_i \,|\, p) + \log P_{\mathcal{F}(m_i^{(L)}:=z)}(o_{hj} \,|\, p)\}, \tag{14}$$

N and M represent the number of elements in O and O_h respectively. This equation seeks a z that the log probabilities of correct answers are encouraged to be larger than those of false answers by at least a certain margin $\mathcal{M}$.

On the other hand, it involves ensuring that the absolute probabilities of correct answers do not decrease while simultaneously controlling that the probabilities of false answers do not increase during editing, which can be represented as two objectives $\mathcal{L}_2(O, z)$ and $\mathcal{L}_3(O_h, z)$ respectively as:

$$\mathcal{L}_2(O, z) = \frac{1}{N} \sum_{i=1}^{N} max\{0,\ \log P_{\mathcal{F}}(o_i \,|\, p) - \log P_{\mathcal{F}(m_i^{(L)}:=z)}(o_i \,|\, p)\}, \tag{15}$$

$$\mathcal{L}_3(O_s, z) = \frac{1}{M} \sum_{i=1}^{M} max\{0,\ \log P_{\mathcal{F}(m_i^{(L)}:=z)}(o_{hi} \,|\, p) - \log P_{\mathcal{F}}(o_{hi} \,|\, p)\}. \tag{16}$$

$P_{\mathcal{F}}(o_i \,|\, p)$ refers the probability output by the unedited model. $P_{\mathcal{F}(m_i^{(L)}:=z)}(o_i \,|\, p)$ refers the probability output by the model operated with the same operation in Eq. (12). These two equations respectively seek a z that the log probabilities of correct answers are encouraged not to be decreased and the log probabilities of false answers not to be increased. Finally, these proposed objectives are jointly optimized with the editing objective $\mathcal{L}_e(o^*, z)$ to obtain v_*:

$$v_* = \arg\min_z \mathcal{L}_e(o^*, z) + \alpha \mathcal{L}_1(O, O_h, z) + \beta \mathcal{L}_2(O, z) + \gamma \mathcal{L}_3(O_h, z). \tag{17}$$

Finally, the Eq. (13) is utilized to update parameters in model.