Joint Knowledge Editing for Information Enrichment and Probability Promotion

Wenhang Shi[1], **Yiren Chen**[2], **Shuqing Bian**[3], **Xinyi Zhang**[1*], **Zhe Zhao**[3*], **Pengfei Hu**[3], **Wei Lu**[1], **Xiaoyong Du**[1]

[1] Renmin University of China
[2]Peking University
[3]Tencent
{wenhangshi, xinyizhang.info, lu-wei, duyong}@ruc.edu.cn, yrchen92@pku.edu.cn, shuqingbian@gmail.com, {nlpzhezhao, alanpfhu}@tencent.com

Abstract

Knowledge stored in large language models requires timely updates to reflect the dynamic nature of real-world information. To update the knowledge, most knowledge editing methods focus on the low layers, since recent probes into the knowledge recall process reveal that the answer information is enriched in low layers. However, these probes only and could only reveal critical recall stages for the original answers, while the goal of editing is to rectify model's prediction for the target answers. This inconsistency indicates that both the probe approaches and the associated editing methods are deficient. To mitigate the inconsistency and identify critical editing regions, we propose a contrast-based probe approach, and locate two crucial stages where the model behavior diverges between the original and target answers: **Information Enrichment** in low layers and **Probability Promotion** in high layers. Building upon the insights, we develop the **J**oint knowledge **E**diting for information **E**nrichment and probability **P**romotion (JEEP) method, which jointly edits both the low and high layers to modify the two critical recall stages. Considering the mutual interference and growing forgetting due to dual modifications, JEEP is designed to ensure that updates to distinct regions share the same objectives and are complementary. We rigorously evaluate JEEP by editing up to thousands of facts on various models, *i.e.*, GPT-J (6B) and LLaMA (7B), and addressing diverse editing objectives, *i.e.*, adding factual and counterfactual knowledge. In all tested scenarios, JEEP achieves best performances, validating the effectiveness of the revealings of our probe approach and the designs of our editing method.

Code — https://github.com/Eric8932/JEEP

1 Introduction

Large Language Models (LLMs) are renowned for their extensive knowledge storage, addressing queries by recalling the encoded knowledge (Petroni et al. 2019; Touvron et al. 2023). However, their original knowledge might be incorrect or outdated due to the swift pace of global events, demanding timely updates to this stored information (Jang et al. 2022). For instance, the answer "France" to the query "World Cup's winner is" remains valid until year 2022, but now "Argentina" is the correct response. Common methods like retraining model with revised corpora is prohibitively costly, while continually fine-tuning on the corrected dataset causes catastrophic forgetting (McCloskey and Cohen 1989; Kirkpatrick et al. 2017; Shi et al. 2023). Consequently, there is a growing interest in *Knowledge Edit*, whose goal is to update the model's original knowledge to the target knowledge both efficiently and accurately (De Cao, Aziz, and Titov 2021; Mitchell et al. 2022).

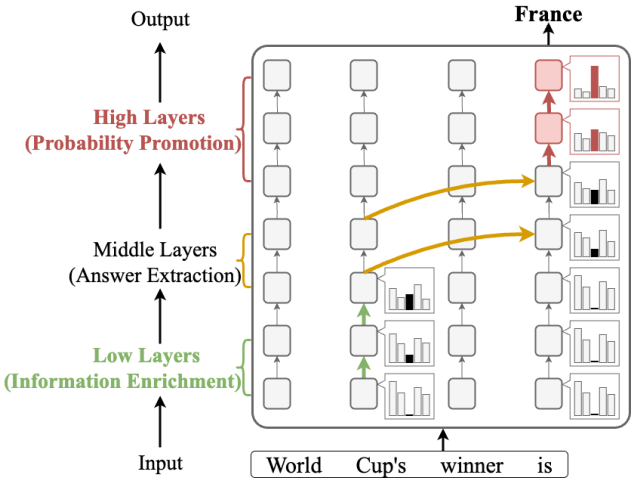

Figure 1: Using probability as information indicator, we directly observe the original answer's information flow within the model. By further contrasting information flow of the original and target answers, we identify two critical recall stages for knowledge editing: Information Enrichment in low layers and Probability Promotion in high layers.

Knowledge editing methods can be categorized based on whether they modify model parameters (Yao et al. 2023). Weight-preserved methods incorporate additional structures to handle each editing requirement, facing scalability issues as the number of edits increases. (Huang et al. 2023; Hartvigsen et al. 2024; Yu et al. 2024). Therefore, we focus on weight-modified methods, which could accommodate considerable updates in a single operation, and include two distinct methods (Meng et al. 2022b; Tan, Zhang, and Fu 2023; Li et al. 2024). Meta-learning methods train

*Corresponding Author

hyper-networks for generating updated parameters, suffering from poor generalization when edits increase (Mitchell et al. 2021; Tan, Zhang, and Fu 2023). Locate-then-edit methods locate primary storage locations of knowledge, by probing into model's recall process of the original answers, then edit the specific locations (Dai et al. 2021; Meng et al. 2022a). Recent explorations show that the output answer processing in LLMs involves two critical phases: information enrichment in low layers and answer extraction in middle layers (Meng et al. 2022a; Geva et al. 2023). Building on the insights, locate-then-edit methods all focus updates on the low layers (Meng et al. 2022a,b; Li et al. 2024).

Despite the demonstrated capability of existing locate-then-edit methods for large-scale knowledge editing (Meng et al. 2022b; Li et al. 2024), their effectiveness is hampered by the inconsistency between the probe approaches for original answers and the editing goals for target answers. Previous probes assess the impact of ablating specific modules by comparing the probabilities of the original answers in model's predictions before and after such interventions. This reliance on final probabilities limits their applicability to the target answers, which have low probabilities in the predictions. However, the stages vital for effective knowledge editing do not merely align with those critical for recalling original answers, but are distinctively associated with the stages where the model's behavior diverges between the original and target answers. To rectify the inconsistency, we propose a contrast-based probe approach. It observes the answer's probability across different representations as in Fig. 1, which depicts the original answer's information flow through each layer (nostalgebraist 2020). By contrasting information changes of the original and target answers, we identify two critical stages for knowledge editing: **Information Enrichment** in low layers and **Probability Promotion** in high layers. The low layers enrich answer-related information to the representations, and the high layers promote answer's probability to make it the final output.

Building on these insights, we develop the JEEP method, which strategically targets both low and high layers for a holistic alternation of knowledge recall process. It jointly edits layers responsible for information enrichment and probability promotion stages. However, modifications to different model regions can interact in complex ways, potentially leading to conflicting outcomes. To ensure the updates are not only cohesively integrated but also complementary, JEEP first synergizes the optimization objectives of both updates. It then adaptively adjusts the update degree for different layers based on the information changes required at each recall stage, thereby effectively altering model's predictions with minimal parameter changes. To validate our method, we conduct extensive experiments involving edits ranging from 1 to 10,000 across various model architectures, including GPT-J (6B) and LLaMA (7B) (Wang and Komatsuzaki 2021; Touvron et al. 2023), and datasets such as zsRE and Multi-COUNTERFACT (Levy et al. 2017; Meng et al. 2022b). In all tested scenarios, JEEP consistently delivers the optimal performances, confirming the effectiveness of our methodological designs and validating our probe approach to identify critical editing stages.

The main contributions of our study are threefold: (1) We identify the inconsistency between existing probes for the original answers and the editing goals for the target answers. By introducing a contrast-based probe approach, we address the inconsistency and locate two critical regions for effective knowledge editing. (2) We propose JEEP, a knowledge editing method that jointly edits both the low and high layers to modify the stages of information enrichment and probability promotion, offering a more holistic and effective approach to knowledge editing. (3) We conduct extensive comparative experiments across various numbers of edits, model architectures, and datasets. These experiments not only demonstrate the effectiveness of our JEEP method but also highlight the practical values of our probe findings in designing knowledge editing methods.

2 Related Work

2.1 Probe into Knowledge Recall Process

Probe approaches into knowledge recall process try to elucidate a model's internal processing of the output answers (Olah 2022; Gurnee et al. 2023; Bills et al. 2023), where our focus is on critical regions for knowledge editing (Katz and Belinkov 2023). It is widely believed that the Multi-Layer Perceptron (MLP) module serves as a primary repository of knowledge (Geva et al. 2020; Dai et al. 2021; Kobayashi et al. 2023), and numerous studies analyze its behavior as key-value memories (Geva et al. 2022; Dar et al. 2022). By restoring corrupted hidden states, (Meng et al. 2022a) reveals that early-to-middle MLP layers enrich subject-related factual information in representations of subject's last token, thereby pinpointing a more specific knowledge storage location. Regarding the Multi-Head-Self-Attention (MHSA) module, (Dar et al. 2022) shows that MHSA parameters also encapsulate knowledge. Further, (Geva et al. 2023) blocks the attention computations to trace detailed information flows of final predictions, underscoring MHSA's role in extracting answers to the prediction position. These findings outline the critical stages of the knowledge recall process. However, the used ablation-based probes only and could only reveal critical stages for the original answers, failing to provide complete critical regions for knowledge editing, which aims to edit for the target answers. By observing and contrasting the original and target answers' information changes across layers, we endeavor to resolve the inconsistency and provide a more comprehensive understanding of the knowledge recall process for effective editing.

2.2 Knowledge Editing

Knowledge editing methods can be categorized into two lines based on whether they modify model parameters. Weight-preserved methods incorporate additional structures to handle incoming editing demands, such as new neurons (Huang et al. 2023), scope classifiers with counterfactual models (Mitchell et al. 2022), adapters (Hartvigsen et al. 2024), LoRA modules (Yu et al. 2024), and in-context learning examples (Zheng et al. 2023). But they are not scalable, as they become more costly and less effective

with an increasing number of edits. In contrast, weight-modified methods directly update the model's weights, enabling large-scale simultaneous edits. The most straightforward approach, constrained fine-tuning, suffers from severe overfitting issues (Zhu et al. 2020). Meta-learning-based methods train hyper-networks to generate parameter updates that satisfy the generalization and locality of editing, yet they struggle to maintain performance at larger editing scales (De Cao, Aziz, and Titov 2021; Mitchell et al. 2021; Tan, Zhang, and Fu 2023). Locate-then-edit methods (Meng et al. 2022b; Li et al. 2024) leverage findings from probe approaches into knowledge recall process (Meng et al. 2022a; Geva et al. 2023), locating low-layer MLPs and editing them as key-value memories. While these methods could handle extensive edits in one operation, the inconsistency between the probes and editing targets limits their effectiveness. Typically, they modify only partial critical regions and fail to align the final predictions with the updated information effectively. To address these challenges, we propose to edit both the low and high layers to modify the information enrichment and probability promotion stages based on our probe revealings. This approach leads to a more thorough modification of the knowledge recall process and improved editing performance.

3 Preliminaries

3.1 Language Modeling

In a decoder-only language model $\mathcal{F}_\theta$ (Brown et al. 2020), the input sequence $[\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_E]$ goes through D-layer computations, and the last token representation in the final layer h_E^D would be mapped to the vocab distribution through the language model (LM) head W_{lm}, to decode the probabilities of the next token x_{E+1}:

$$\mathcal{F}_\theta\left([\mathrm{x}_1, \mathrm{x}_2 \ldots \mathrm{x}_E]\right) \triangleq \mathbb{P}_E^D = softmax(W_{lm}(h_E^D)). \quad (1)$$

So the representation h_E^D encodes information of the next token. This information is accumulated during D-layer residual connections:

$$\begin{aligned} h_E^D &= h_E^0 + \sum_{l=1}^{D}(a_E^l + m_E^l), \\ where\ a_E^l &= W_{O^{\mathrm{MHSA}}}^l \mathrm{MHSA}(\gamma(h_1^{l-1}, h_2^{l-1}, ..., h_E^{l-1})) \\ m_E^l &= W_{O^{\mathrm{MLP}}}^l \sigma(W_{IN^{\mathrm{MLP}}}^l \gamma(h_E^{l-1})), \end{aligned} \quad (2)$$

and h^0 is the embedding and γ denotes layer normalization.

3.2 Knowledge Editing

LLM $\mathcal{F}$ has encoded abundant knowledge in its parameters:

$$K_{\mathcal{F}} = \left\{(x_i, y_i)_{i=1}^N, \mathcal{F}(x_i) = y_i\right\}, \quad (3)$$

where $K_{\mathcal{F}}$ is the **original knowledge** in the model and (x_i, y_i) denotes a input prompt and answer pair. Given a knowledge pair (x, y), such as (World Cup's winner is, France), there are two responding sets: Equivalent Set $E(x, y)$, containing all semantically equivalent knowledge pairs to (x, y), an example would be (Who is the World Cup's winner?, France); Unrelated Set $U(x, y)$, containing all unrelated knowledge pairs to (x, y), an unrelated pair would be (Which country does Paris belong to?, France).

For m pieces of **target knowledge** pairs $(x_i, y_i')_{i=1}^m$ to be edited, knowledge editing aims to change model's predictions to the target answers on these inputs (Efficacy), while generalizing the model to their equivalent pair collection $\bigcup_{i=1}^m E(x_i, y_i')$ (Generalization) , and maintaining the predictions on the unrelated pair collection $\bigcap_{i=1}^m U(x_i, y_i')$ (Locality). Therefore, the edited model $\tilde{\mathcal{F}}$ should simultaneously satisfy the following three goals:

$$\begin{aligned} &\tilde{\mathcal{F}}(x) = y \ \wedge\ \tilde{\mathcal{F}}(x^e) = y^e \ \wedge\ \tilde{\mathcal{F}}(x^u) = y^u, \\ &(x, y) \in \left\{(x_i, y_i')_{i=1}^m\right\}, (x^e, y^e) \in \bigcup_{i=1}^m E, (x^u, y^u) \in \bigcap_{i=1}^m U. \end{aligned} \quad (4)$$

Detailed measuring metrics are in Appendix C.

4 Method

Exploring the knowledge recall process aids in designing effective knowledge editing methods (Meng et al. 2022b; Li et al. 2024). But existing probes could only locate prominent regions for original answers, which is inconsistent to the knowledge editing for target answers. To mitigate this, we propose a contrast-based probe approach and identify two critical recall stages for editing (§4.1). Based on our discoveries, we develop JEEP to more thoroughly modify the knowledge recall process to alter model predictions, while addressing the issues associated with joint updates (§4.2).

4.1 Unveiling Critical Editing Stages

Our probe contrasts the detailed information flow of the original and target answers across different representations. Guided by the probe, we identify two critical editing stages: Information Enrichment in low layers and Probability Promotion in high layers.

The Contrast-Based Probe Approach Since previous ablation-based probes fail to analyze the target answers, we directly observe and contrast the differences in the information flow between the original and target answers. Extended from Eq. 2, representation at any position i and any layer l is refined by residual connections. So the LM head's mapping could be applied to any representation h_i^l (nostalgebraist 2020):

$$\mathbb{P}_i^l = softmax(W_{lm}(h_i^l)). \quad (5)$$

Recording the probability $\mathbb{P}(t)$ of the answer's first token t within the distribution $\mathbb{P}$, we can observe the complete information flow of answers inside the model. But probability only reflects absolute information and are often very low for the target answers. Therefore, we further calculate the rank of the target token t in the distribution:

$$\mathrm{rank}(t|\mathbb{P}) = \sum_{t' \neq t} \mathbf{1}(\mathbb{P}(t') > \mathbb{P}(t)). \quad (6)$$

Note that the subsequent "increase" for rank indicates the rank value is approaching 0, and the representation contains

more information about the token. In Fig. 2, we input 10,000 prompts x into LLaMA-7B model, recording average probability and rank of the first token in the original answers y and target answers y' for editing (see Appendix A for details). To differentiate information flow in different positions, we decompose the prompt-answer knowledge pair (x, y) into the form of a triplet $< subject, relation, object >$, the most common format for representing knowledge (Petroni et al. 2019). For example, (World Cup's winner is, Argentina) can be decomposed into <World Cup, 's winner is, Argentina>. And we focus on representations at subject's last and prediction positions, since the former encapsulates all the subject information (Meng et al. 2022a), and the later is where the prediction is made.

By contrasting the differences in information flow between the original and target answers, we identify critical stages with significant model behavior differences, which are suitable for knowledge editing. As depicted in Fig. 2, these stages align with the low, middle, and high layers of the model. In the low layers, at the subject's last position, the rank of original answers increases, marking an **Information Enrichment** stage, mainly driven by the MLP module (Meng et al. 2022a). Target answers, however, do not experience similar enrichment, as their rank only increases marginally, highlighting this stage's importance for effective knowledge editing. In the middle layers, information for both original and target answers at the prediction position starts to increase, evidenced by a sharp rise in rank, primarily due to the MHSA module (Geva et al. 2023). This indicates that the MHSA consistently extracts all potential answers fitting the relation, including the target answers, suggesting a stable extraction pattern. Therefore, this stage requires no modification during editing, aligning with previous editing methods (Meng et al. 2022b; Li et al. 2024). Finally, in the early high layers at the prediction position, original answers undergo **Probability Promotion**. Although this probability increase might seem minor due to the slight rank rise of original answers at this stage, the rank of target answers, lacking probability promotion, decreases instead of increasing. This contrast highlights the crucial role of Probability Promotion in elevating original answers to become the top-ranked output, indicating that this stage should be modified for effective knowledge editing.

Besides, we note a decline in the probability of original answers at the prediction position in the final layers, and even the rank decreases in the penultimate layer. To explore it, we expand our analysis to include additional word sets, examining their information changes. We discover that the uppermost layers function as distribution normalizers, increasing information entropy at the prediction position and aligning outputs with realistic data distribution. So the early high layers are more conducive for editing. We leave the detailed analysis in Appendix A.

4.2 Joint Knowledge Editing

Based on the observations above, we develop the JEEP method, which edits both the low and high layers, specifically targeting the MLP modules therein, to simultaneously modify the stages of information enrichment and probability promotion. Initially, we outline the comprehensive process of the joint editing. Subsequently, we present two challenges associated with dual updates, mutual interference and increased forgetting, and our tailored designs to solve them.

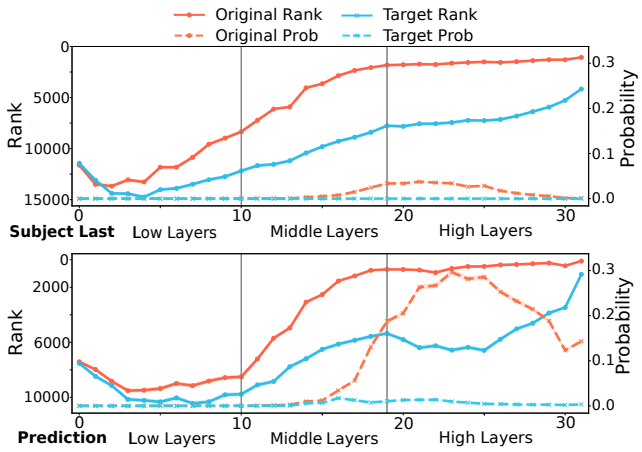

Figure 2: Original and Target answers' information change in low, middle and high layers, indicated by rank and probability (Prob). The top and bottom graphs are for Subject Last and Prediction positions respectively.

Joint Editing Procedure To revise the knowledge recall process more completely, we edit the MLP modules in both low layers $[l', L']$ and high layers $[l^*, L^*]$. We analyze the MLP module as a key-value memory, where W_{IN} encodes input into key vectors and W_O maps keys to output values containing knowledge. Suppose the model memorizes n pieces of original knowledge $(x_i, y_i)_{i=1}^n$, we have n mappings encoded in W_O:

$$W_O K_0 = V_0, \quad \begin{matrix} K_0 \triangleq [k_1 \mid k_2 \cdots \mid k_n] \\ V_0 \triangleq [v_1 \mid v_2 \cdots \mid v_n]. \end{matrix} \tag{7}$$

To edit m knowledge pairs, we update the model in three steps as shown in Fig. 3. Firstly, for every target knowledge pair (x, y'), we calculate the corresponding value vector v_i^L ($i \in \{i', i^*\}, L \in \{L', L^*\}$) to replace the current hidden state h_i^L, at subject last position i' and prediction position i^* in last critical layer L' and L^*. We optimize the residual vector δ ($\delta \in \{\delta', \delta^*\}$) by gradient descend to alter model's prediction on x to y' (Step1):

$$v_i^L = h_i^L + \underset{\delta}{argmin} \frac{1}{P} \sum_{j=1}^{P} \mathcal{L}_{\mathcal{F}(h_i^L += \delta)}(y' \mid p_j + x). \tag{8}$$

Different prefixes p_j are used to bolster generalization. We formulate the editing to adding m new mappings while preserving the existing n ones with minor change Δ to W_O:

$$(W_O + \Delta)[K_0 K_1] = [V_0 V_1], \quad \begin{matrix} K_1 \triangleq [k_{n+1} \mid \cdots \mid k_{n+m}] \\ V_1 \triangleq [v_{n+1} \mid \cdots \mid v_{n+m}]. \end{matrix} \tag{9}$$

By derivation in (Meng et al. 2022b), we could obtain:

$$\Delta = R K_1^T (C_0 + K_1 K_1^T)^{-1}, \tag{10}$$

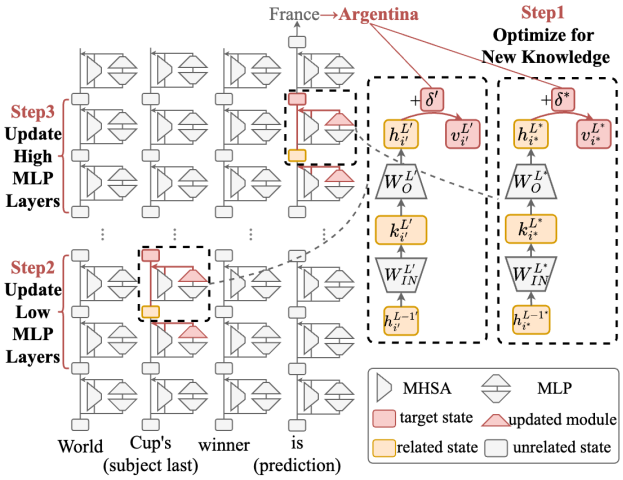

Figure 3: Procedure of JEEP method. Firstly, it computes δ' and δ^* for low and high layers simultaneously, optimizing for injecting new knowledge. Secondly, it uses the residual errors of $v_{i'}^{L'}$ to update the low MLP layers. Finally, it uses the residual errors of $v_{i^*}^{L^*}$ to update the high MLP layers.

where $R \triangleq V_1 - W_O K_1$, denoting the residual errors when assessing new knowledge mappings on the original weights. And $C_0 \triangleq K_0 K_0^T$ denotes the uncentered covariance of the original key vectors. Lacking access to the input prompts for the original knowledge K_0, we instead sample prompts from Wikipedia and estimate C_0 based on their key vectors to W_O: $C_0 \triangleq \lambda \cdot \mathbb{E}_k \left[kk^T\right]$. Utilizing Eq. 10, we initiate updates in the lower region (Step2) before advancing to the upper region (Step3). Within each region, updates proceed sequentially from the lower to the higher layers. Specifically, for the current updating layer l, the key vector k_i^l is calculated by averaging the input keys of editing prompts with different prefixes: $k = \frac{1}{P}\sum_{j=1}^{P} k(p_j + x)$. To distribute the updates evenly across all targeted layers, we spread the current residual error by the number of layers pending update: $r = \frac{v_i^L - h_i^L}{L - l + 1}$. Horizontally stacking different k and r corresponding to new knowledge pairs, we obtain K_1 and R and compute the matrix update by Eq. 10.

Synergistic Optimization The first challenge in joint editing is mutual interference between updates across different regions. To harmonize the dual updates, we compute the value vectors in Eq. 8 simultaneously, ensuring they are optimized in a synergistic manner. The optimization objective is defined as follows:

$$\begin{aligned}\mathcal{L}(\delta', \delta^*) = &-\log \mathbb{P}^D_{n\ \mathcal{F}(h_{i'}^{L'} += \delta', h_{i^*}^{L^*} += \delta^*)} [y' \mid x] \\ &+ \beta' * \|\delta'\| + \beta^* * \|\delta^*\| \\ &+ \alpha' * D_{kl}(\mathbb{P}^D_{i\ \mathcal{F}(h_{i'}^{L'} += \delta')} [x] \mid\mid \mathbb{P}^D_i [x]),\end{aligned} \tag{11}$$

where β' and β^* are coefficients for the weight decay loss of δ' and δ^*, respectively. And α' adjusts the KL-divergence loss for output at the subject's last token. In addition to the language model loss, which is used to inject new knowledge into the model, we further incorporate KL-divergence and weight decay terms to mitigate forgetting, ensuring minimal disruption to other knowledge.

Adaptive Updates The second challenge encountered by a joint editor is increased forgetting due to more parameter updates within the model. To adjust the extent of the updates, we compress the knowledge encoded in the modifications, by clamping the L2 norms of δ and δ' based on their respective original hidden state norms:

$$\|\delta'\| <= \gamma' * \left\|h_{i'}^{L'}\right\|, \|\delta^*\| <= \gamma^* * \left\|h_{i^*}^{L^*}\right\|, \tag{12}$$

where γ' and γ^* are coefficients to control the upper bounds for δ' and δ^*, respectively. We find that modifications in the upper layers of model have a greater impact on the final outputs, suggesting that the required change in probability promotion to alter predictions is less than that for information enrichment (see Appendix B). Thus, more updates should be allocated to the low layers. Coupled with the fact that the lower state norm is smaller, we set: $\gamma' > \gamma^*$. Furthermore, since the knowledge encoded in low layers is denser, we spread the residual errors as follows: $r = \frac{v_{i'}^{L'} - h_{i'}^{L'}}{\sqrt{L' - l' + 1}}$ (Li et al. 2024), enhancing the updates. By adaptively adjusting the update magnitudes in these two regions, we effectively alter model's predictions with minimal parameter changes.

5 Experiment

5.1 Experimental Setup

We conduct experiments on two models, GPT-J (6B) (Wang and Komatsuzaki 2021) and LLaMA (7B) (Touvron et al. 2023), which feature parallel and sequential MHSA and MLP modules, respectively. Based on the three objectives of knowledge editing, we evaluate three corresponding key metrics: **Efficacy Success** (ES), **Generalization Success** (GS) and **Locality Success** (LS), then compute their harmonic mean **Score** as in (Meng et al. 2022b) (see Appendix C for detailed descriptions). For baselines, we first consider **FT-WD**, fine-tuning with weight decay to prevent forgetting (Zhu et al. 2020). Next, for meta-learning-based methods, we include **MEND** and its improved version **MALMEN**, which further supports multiple facts' editing at once (Mitchell et al. 2021; Tan, Zhang, and Fu 2023). Finally, we compare with locate-then-edit methods: **ROME**, **MEMIT** and **PMET** (Meng et al. 2022a,b; Li et al. 2024). Since **ROME** originally targets one fact at a time, we adapt it to a sequential version to accommodate massive edits (Meng et al. 2022b). We evaluate editors' ability to add factual knowledge (§5.2), and further evaluate the capability to add counterfactual knowledge (§5.3), which is a more challenging scenario. Implementation details are in Appendix D.

5.2 Adding Factual Knowledge

Editing 10k Samples The primary goal of *Knowledge Edit* is to correct inaccuracies in the model's original knowledge. Initially, we evaluate the editing methods' abilities to add *factual* knowledge. We extract 10,000 real-world factual pairs (x, y') from zsRE (Levy et al. 2017), a question-

	Input Prompt	MEMIT	PMET	JEEP
Edited	*Where was **Henry S. LeBlanc** from?*	Canada	Canada	Canada
Equivalent	*Where did **Henry S. LeBlanc** come from?*	of course	Canada	Canada
Unrelated	*Where does **creatine** come from in the body?*	liver	Where is creatine from?	liver

Table 1: A case of three editors' predictions on the prompts for evaluating efficacy, generalization and locality. Input Prompt consists of the subject, identified in bold, and the relation. Answers in red are inaccurate, either meaningless or repetitive.

Method	**Score** ↑	**ES** ↑	**GS** ↑	**LS** ↑
GPT-J	26.4	26.4 (0.6)	25.8 (0.5)	27.0 (0.5)
FT-WD	42.1	69.6 (0.6)	64.8 (0.6)	24.1 (0.5)
MEND	20.0	19.4 (0.5)	18.6 (0.5)	22.4 (0.5)
ROME	2.6	21.0 (0.7)	19.6 (0.7)	0.9 (0.1)
MALMEN	47.1	98.3 (0.3)	90.1 (0.2)	23.6 (0.4)
MEMIT	50.7	96.7 (0.3)	89.7 (0.5)	26.6 (0.5)
PMET	51.0	96.9 (0.3)	90.6 (0.2)	26.7 (0.2)
JEEP	**51.5**	**98.4** (0.2)	**91.5** (0.3)	**26.9** (0.2)
LLaMA	44.4	43.6 (0.3)	42.6 (0.5)	49.2 (0.6)
MALMEN	66.7	90.1 (0.4)	84.6 (0.3)	45.3 (0.5)
MEMIT	70.4	95.1 (0.2)	89.5 (0.4)	47.8 (0.5)
PMET	69.2	91.3 (0.3)	86.4 (0.5)	48.0 (0.3)
JEEP	**72.3**	**96.0** (0.2)	**93.2** (0.4)	**49.1** (0.2)

Table 2: Results on zsRE for 10,000 edits on GPT-J (6B) and LLaMA (7B) models.

Method	**Score** ↑	**ES** ↑	**GS** ↑	**LS** ↑
GPT-J	22.4	15.2 (0.7)	17.7 (0.6)	83.5 (0.5)
FT-WD	67.6	99.4 (0.1)	77.0 (0.7)	46.9 (0.6)
MEND	23.1	15.7 (0.7)	18.5 (0.7)	**83.0** (0.5)
ROME	50.3	50.2 (1.0)	50.4 (0.8)	50.2 (0.6)
MALMEN	63.4	98.2 (0.8)	46.0 (0.8)	65.1 (0.6)
MEMIT	85.8	98.9 (0.2)	88.6 (0.5)	73.7 (0.5)
PMET	86.2	99.5 (0.1)	**92.8** (0.4)	71.4 (0.5)
JEEP	**86.5**	**99.8** (0.1)	90.9 (0.5)	73.1 (0.5)
LLaMA	19.7	12.9 (0.6)	16.0 (0.7)	83.8 (0.4)
MALMEN	66.1	96.0 (0.3)	55.3 (0.8)	59.3 (0.8)
MEMIT	84.1	98.4 (0.2)	80.6 (0.6)	**76.3** (0.5)
PMET	84.8	98.9 (0.2)	**86.2** (0.5)	73.2 (0.6)
JEEP	**85.9**	**99.8** (0.1)	85.4 (0.5)	75.7 (0.5)

Table 3: Results on Multi-COUNTERFACT for 10,000 edits on GPT-J (6B) and LLaMA (7B) models.

answering dataset. As the editing targets are correct answers, all three metrics compute the proportion of tokens in y' that have the highest probability given the input prompt x (see Appendix C). As shown in Table 2, not all methods can handle such a large volume of edits, highlighting the complexity of editing 10,000 samples concurrently. Some baseline methods, such as MEND and ROME, perform even worse than simple fine-tuning. The JEEP method shows superior performance across all three metrics on both models and further improvements on the advanced, knowledge-rich LLaMA model, validating its editing designs. And compared to the low-layers-focused methods MEMIT and PMET, JEEP's improvements emphasize the benefits of a more comprehensive understanding and modification of the knowledge recall process in editing methods. Moreover, the 95% confidence intervals of results confirm that JEEP achieves statistically significant improvements over the baselines, with consistently narrow intervals, highlighting its robustness.

Case Study In Table 1, we showcase the predictions of the LLaMA model, after being edited by different editors, on three metrics for one of the 10,000 editing samples. All editors make accurate predictions on the edited prompt, indicating that achieving basic editing efficacy is feasible. However, only JEEP performs properly on both equivalent and unrelated prompts, suggesting incomplete modifications to the knowledge recall process result in either insufficient or overfitting editing.

5.3 Adding Counter-Factual Knowledge

Editing 10k Samples We next shift to a more challenging scenario: evaluating methods' capabilities in injecting *counterfactual* information. We edit 10,000 samples from the Multi-COUNTERFACT dataset (Meng et al. 2022b), where the editing updates original correct answers to target incorrect answers. The evaluation for successful editing is more lenient, and the metrics calculate the proportion of cases where the generation probability of the target incorrect answer exceeds that of the original correct answer (Appendix C). But as for locality, a higher probability of the original correct answer is considered successful. Given the difficulty for adding this artificially constructed erroneous knowledge, it's challenging for editing methods to excel in the efficacy, generalization, and locality simultaneously. As illustrated in Table 3, no method consistently exceeds the others across all metrics. In this unnatural setting, our JEEP method achieves the best efficacy and near-best generalization and locality, leading to optimal overall performance. This indicates that more comprehensive modifications of critical knowledge recall stages are also beneficial for adding counterfactual knowledge, validating the effectiveness of our probe findings and editing method.

Editing Scaling To evaluate the scalability of our method, we compare the performance of methods capable of handling 10,000 edits as the number of edits m increases[1]. As shown in Fig. 4, JEEP consistently delivers better overall results, with the improvements becoming more pronounced as the number of edits grows. Specifically, JEEP achieves

[1] m is sampled from a log-scale curve: $m_i = exp(ln(10,000) * \frac{i}{16})$, for non-negative integers i.

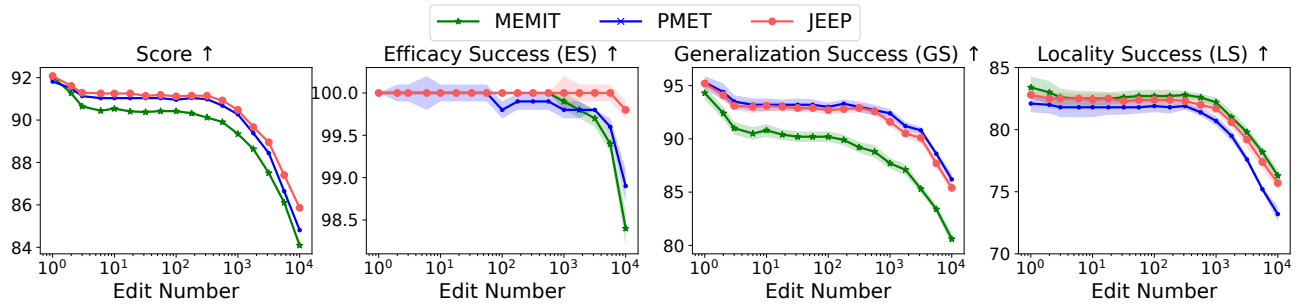

Figure 4: Scaling curves plot performance change against different editing numbers (log-scale) on LLaMA (7B). 95% confidence intervals are shown as areas.

Variant	Score ↑	ES ↑	GS ↑	LS ↑
w/o δ'& Step2	66.3	89.8	56.6	60.7
w/o δ^*& Step3	84.1	98.4	80.6	**76.3**
w/o Step2	40.0	36.5	24.1	81.5
w/o Step3	81.9	95.4	77.5	75.4
Separate Optimization	83.3	98.2	83.3	72.5
Even Spread in Step2	85.4	98.8	84.0	**76.3**
w/ MHSA in Step3	85.2	99.0	**86.6**	73.8
JEEP	**85.9**	**99.8**	85.4	75.7

Table 4: Effects of ablating different designs of JEEP. The top half focuses on single-region editing. The bottom half ablates different designs within JEEP.

100% accuracy in all cases except for 10,000 edits and remains near the best in generalization and locality, demonstrating its robust performance across varying edit numbers.

5.4 Ablation Study

To elucidate the contributions of updates from different regions and various designs in JEEP, Table 4 presents the results of different ablation variants, editing 10,000 samples from Multi-COUNTERFACT on LLaMA.

1) Initially, to assess JEEP's design of dual-region modifications, we examine single-region editing. In **w/o** δ'& **Step2** and **w/o** δ^*& **Step3**, only one residual vector in Eq. 8, either δ^* or δ', is computed and used to update the high or low layers respectively. They both underperform compared to JEEP, underscoring the advantage of updating both regions. Notably, editing the lower layers along performs better, suggesting that without the information enrichment from the lower MLP, modifying probability promotion alone is less effective. Additionally, **w/o Step2** or **w/o Step3** keeps the simultaneous computation of δ' and δ^*, but subsequently updates only one region. This approach considers the interplay between updates but limits the actual modification to one regions. As with previous findings, single-region editing yields inferior results compared to JEEP, with lower-layer editing still shows better outcomes. However, compared to computing only one δ, the dual computations of δ performs worse, suggesting a more holistic consideration of the critical recall stages without corresponding updates may instead harm the editing effect. JEEP improves this by adaptively updating both regions, achieving a more balanced distribution of information changes and superior performance.

2) Moreover, we ablate the designs in JEEP. **Separate Optimization** deviates from JEEP's synergistic design in Eq. 11, by independently computing δ' and updating the low layers before computing δ^* and updating the high layers. Its performance is not only inferior to JEEP but also worse than single-region editing, indicating that isolating the optimization of two regions leads to mutual interference between them. In addition, **Even Spread in Step2** replaces the excessive updates to the low layers by spreading the residual errors evenly: $r = \frac{v_{i'}^{L'} - h_{i'}^{L'}}{L'-l'+1}$. It performs worse than JEEP, achieving better locality at the cost of efficacy and generalization, but still outperforms updating one region. Furthermore, considering the computational similarities between W_O in MLP and MHSA, **w/ MHSA in Step3** instead edits MHSA in high layers. Although this adjustment does not outperform JEEP, it shows improved performance over single-region editing, indicating that both MLP and MHSA could promote answer's probability in the upper layers.

These ablation studies not only validate the insights from our probe experiments but also demonstrate the effectiveness of the design choices in JEEP.

6 Conclusion

In this study, we discover the inconsistency between existing probe approaches for the original answers and the knowledge editing goal for the target answers. To address the inconsistency, we propose a contrast-based probe approach, identifying two critical knowledge recall stages for editing: Information Enrichment in low layers and Probability Promotion in high layers. Based on these insights, we develop JEEP, a joint editing method targeting both low and high layers to modify the information enrichment and probability promotion stages simultaneously. Through synergistic optimization and adaptive updates, JEEP addresses the challenges of mutual interference and increased forgetting associated with updating different model regions, consistently achieving superior performance across various edit scales, models, and datasets.

Acknowledgements

The work was supported by the Outstanding Innovative Talents Cultivation Funded Programs 2023 of Renmin University of China and the National Natural Science Foundation of China under Grant No. 62072458.

References

Bills, S.; Cammarata, N.; Mossing, D.; Tillman, H.; Gao, L.; Goh, G.; Sutskever, I.; Leike, J.; Wu, J.; and Saunders, W. 2023. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; and Wei, F. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Dar, G.; Geva, M.; Gupta, A.; and Berant, J. 2022. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535*.

De Cao, N.; Aziz, W.; and Titov, I. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

Geva, M.; Bastings, J.; Filippova, K.; and Globerson, A. 2023. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*.

Geva, M.; Caciularu, A.; Wang, K. R.; and Goldberg, Y. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*.

Geva, M.; Schuster, R.; Berant, J.; and Levy, O. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Gurnee, W.; Nanda, N.; Pauly, M.; Harvey, K.; Troitskii, D.; and Bertsimas, D. 2023. Finding Neurons in a Haystack: Case Studies with Sparse Probing. *arXiv preprint arXiv:2305.01610*.

Hartvigsen, T.; Sankaranarayanan, S.; Palangi, H.; Kim, Y.; and Ghassemi, M. 2024. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36.

Huang, Z.; Shen, Y.; Zhang, X.; Zhou, J.; Rong, W.; and Xiong, Z. 2023. Transformer-Patcher: One Mistake worth One Neuron. *arXiv preprint arXiv:2301.09785*.

Jang, J.; Ye, S.; Lee, C.; Yang, S.; Shin, J.; Han, J.; Kim, G.; and Seo, M. 2022. TemporalWiki: A lifelong benchmark for training and evaluating ever-evolving language models. *arXiv preprint arXiv:2204.14211*.

Katz, S.; and Belinkov, Y. 2023. Interpreting Transformer's Attention Dynamic Memory and Visualizing the Semantic Information Flow of GPT. *arXiv preprint arXiv:2305.13417*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Kobayashi, G.; Kuribayashi, T.; Yokoi, S.; and Inui, K. 2023. Feed-forward blocks control contextualization in masked language models. *arXiv preprint arXiv:2302.00456*.

Levy, O.; Seo, M.; Choi, E.; and Zettlemoyer, L. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Li, X.; Li, S.; Song, S.; Yang, J.; Ma, J.; and Yu, J. 2024. PMET: Precise Model Editing in a Transformer. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 18564–18572. AAAI Press.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, 109–165. Elsevier.

Meng, K.; Bau, D.; Andonian, A.; and Belinkov, Y. 2022a. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35: 17359–17372.

Meng, K.; Sharma, A. S.; Andonian, A.; Belinkov, Y.; and Bau, D. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Mitchell, E.; Lin, C.; Bosselut, A.; Finn, C.; and Manning, C. D. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Mitchell, E.; Lin, C.; Bosselut, A.; Manning, C. D.; and Finn, C. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, 15817–15831. PMLR.

nostalgebraist. 2020. interpreting GPT: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

Olah, C. 2022. Mechanistic Interpretability, Variables, and the Importance of Interpretable Bases. https://transformer-circuits.pub/2022/mech-interp-essay/index.html.

Petroni, F.; Rocktäschel, T.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A. H.; and Riedel, S. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Shi, W.; Chen, Y.; Zhao, Z.; Lu, W.; Yan, K.; and Du, X. 2023. Create and Find Flatness: Building Flat Training Spaces in Advance for Continual Learning. In Gal, K.; Nowé, A.; Nalepa, G. J.; Fairstein, R.; and Radulescu, R., eds., *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023)*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, 2138–2145. IOS Press.

Tan, C.; Zhang, G.; and Fu, J. 2023. Massive editing for large language models via meta learning. *arXiv preprint arXiv:2311.04661*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, B.; and Komatsuzaki, A. 2021. GPT-J-6B: A 6 billion parameter autoregressive language model.

Yao, Y.; Wang, P.; Tian, B.; Cheng, S.; Li, Z.; Deng, S.; Chen, H.; and Zhang, N. 2023. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*.

Yu, L.; Chen, Q.; Zhou, J.; and He, L. 2024. MELO: Enhancing Model Editing with Neuron-Indexed Dynamic LoRA. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 19449–19457. AAAI Press.

Zheng, C.; Li, L.; Dong, Q.; Fan, Y.; Wu, Z.; Xu, J.; and Chang, B. 2023. Can We Edit Factual Knowledge by In-Context Learning? *arXiv preprint arXiv:2305.12740*.

Zhu, C.; Rawat, A. S.; Zaheer, M.; Bhojanapalli, S.; Li, D.; Yu, F.; and Kumar, S. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.