

TANGLESCORE: TANGLE-GUIDED PURGE AND IMPRINT FOR UNSTRUCTURED KNOWLEDGE EDITING

Hao-Xiang Xu^{1,2*}, Ziqi Peng^{1*}, Jun-Yu Ma^{1,2}, Yuhao Sun¹, Zhen-Hua Ling^{1,2}, Jia-Chen Gu^{3†}

¹University of Science and Technology of China

²National Engineering Research Center of Speech and Language Information Processing

³University of California, Los Angeles

{nh2001620, aisis, m jy1999, syh3327}@mail.ustc.edu.cn,
zhling@ustc.edu.cn, gujcc@ucla.edu

ABSTRACT

Large language models (LLMs) struggle with inaccurate and outdated information, driving the emergence of *knowledge editing* as a lightweight alternative. Despite their effectiveness in modifying structured knowledge, existing editing methods often fail to generalize to unstructured cases, particularly those involving inherently *hard-to-edit* knowledge, where the original facts tend to be more resistant to change. To address this, we propose a metric, TANGLESCORE, that quantifies the intrinsic difficulty of editing a given knowledge instance. This difficulty, in turn, strongly correlates with the model’s ability to generalize the edit to paraphrased and related prompts. Building on this insight, we introduce a TANGLESCORE-driven method termed **Purge-Imprint Patch Editing (PIPE)**, an editing framework that adaptively modulates the purge and imprint of knowledge based on TANGLESCORE of the target knowledge to be edited, thus adjusting the editing strength to match the instance’s difficulty, thereby enabling more precise and effective model updates. Experiments applying PIPE to four LLMs of varying sizes on two unstructured knowledge editing datasets show that PIPE significantly outperforms previous editing methods by 6.49% in terms of generalization performance. Extensive evaluation show that PIPE also exhibits effectiveness in structured knowledge editing and strong robustness under batch and sequential editing. The code is available at <https://github.com/famoustourist/TangleScore>.

1 INTRODUCTION

Despite the remarkable capabilities of large language models (LLMs) (OpenAI, 2023; Dubey et al., 2024), they often exhibit hallucinations due to incorrect or outdated knowledge embedded in their parameters (Zhang et al., 2023; Ji et al., 2023). Given the high cost of retraining, there has been increasing interest in *knowledge editing* (Sinitsin et al., 2020; Zhu et al., 2020; Dai et al., 2022; Mitchell et al., 2022b; Meng et al., 2022; 2023; Wang et al., 2024; Ma et al., 2024; Gu et al., 2024; Fang et al., 2025; Xu et al., 2025), which aims to revise knowledge in LLMs efficiently. However, most existing methods focus on structured knowledge, while approximately 80% of real-world knowledge is unstructured (Bavota, 2016), as illustrated in Figure 1(a). This mismatch limits their applicability in broader scenarios. Although recent studies have attempted to improve editing in unstructured contexts (Deng et al., 2025; Jiang et al., 2025), a clear gap remains between accuracy and generalization, making unstructured knowledge editing a significant challenge for LLMs.

To diagnose this discrepancy, we conduct a knowledge-wise analysis of the editing outcomes across existing methodologies. Our findings reveal that while these methods demonstrate balanced performance in terms of accuracy and generalization for certain types of knowledge, they exhibit significant degradation when applied to specific samples. Besides, such degradation is not randomly distributed but follows consistent patterns across different editing methods and diverse models. We hypothesize that this stems from a misalignment between the intensity of editing perturbations and

*Equal contribution.

†Corresponding author.

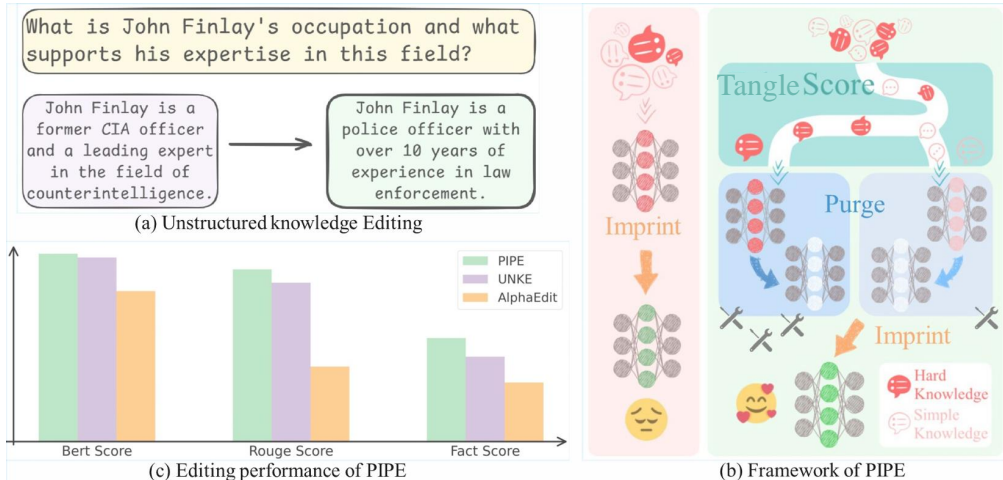


Figure 1: (a) An example of unstructured knowledge editing. (b) A comparison between PIPE and current editing methods. PIPE contains a "purge" phase that utilizes TANGLESORE to quantify knowledge difficulty, applying different purge rates based on their TANGLESORE values. (c) A comparison of editing performance between PIPE and current editing methods.

the degree of coupling between the target knowledge and the model: current methods often fail to effectively overwrite the model’s internal reliance on outdated knowledge, especially when the model’s deeply embedded prior knowledge strongly resists modification.

To identify these knowledge instances, we propose a metric termed TANGLESORE, designed to measure the degree of entanglement between the to-be-edited knowledge and the model’s existing knowledge. This metric jointly considers the discrepancy between model responses to pre-edited and target knowledge, along with their semantic similarity. TANGLESORE is an intrinsic property determined solely by the base model and the knowledge instance, independent of the editing methods used. Notably, TANGLESORE strongly correlates with the generalization performance of editing operations, effectively identifying cases where the model may memorize an edit yet still fail to answer related paraphrases or follow-up questions correctly. By leveraging TANGLESORE, editing methods can adapt to the varying difficulty of knowledge instances, enabling more targeted intervention on hard cases and substantially improving editing effectiveness where previous methods tend to fail.

Consequently, we propose **Purge-Imprint Patch Editing (PIPE)**, a method that can adaptively adjust the purge strategy based on the difficulty of the knowledge, applying more aggressive forgetting for hard-to-edit knowledge. As shown in Figure 1(b), PIPE adopts a two-stage editing paradigm. PIPE first introduces a knowledge purge function to actively suppress the influence of conflicting original knowledge. It dynamically adjusts the degree of purge based on the difficulty of the target knowledge, promoting deeper forgetting for harder edits to minimize interference. Next, PIPE imprints new knowledge using a cause-driven optimization strategy, directly modifying the final token of the input through a dedicated knowledge imprint function. This localized editing improves precision without affecting the overall behavior of the model. As depicted in Figure 1(c), these two stages enable PIPE to separate purging from imprinting, enhancing its robustness to prior knowledge interference.

In order to validate the effectiveness of the proposed PIPE, we conduct a comprehensive evaluation using state-of-the-art LLMs of varying sizes, such as LLaMA3-8B-Instruct (Dubey et al., 2024), LLaMA2-7B-Chat (Touvron et al., 2023), LLaMA2-13B-Chat (Touvron et al., 2023) and Qwen2.5-7B-Instruct (Yang et al., 2024). The UNKEBench (Deng et al., 2025) and AKEW (Wu et al., 2024) are selected to demonstrate the generalizability of our approach across different knowledge formats. Experimental results show that PIPE improves generalization by 6.49% in knowledge editing tasks, compared to the state-of-the-art methods. We also evaluate PIPE on the traditional structured knowledge editing benchmark KEbench (Wei et al., 2024), and conduct both batch and sequential editing, demonstrating the strong robustness of our approach across various settings.

In summary, our contributions in this paper are three-fold: (1) This paper presents the TANGLESORE to quantify the difficulty of editing certain knowledge and demonstrates that prior methods are less

effective at samples with higher difficulty. (2) A method termed PIPE is proposed, which aims to improve the model’s ability to edit hard-to-edit unstructured knowledge by explicitly introducing a purge phase before imprinting new factual information. (3) Evaluation results show that PIPE enhances both the effectiveness and generalization of unstructured knowledge editing.

2 RELATED WORK

Knowledge Editing Conventional knowledge editing methods either modify parameters or preserve them via auxiliary components. *Parameter-Modifying Methods* directly update weights to encode new knowledge. Meta-learning approaches such as KE (Cao et al., 2021), MEND (Mitchell et al., 2022a), and InstructEdit (Zhang et al., 2024) use hypernetworks, with MEND employing low-rank decomposition. Locate-then-edit methods such as ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), and EAC (Xu et al., 2025) apply causal tracing and compress edits to reduce forgetting, while PRUNE (Ma et al., 2025) restricts conditional branches. *Parameter-Preserving Methods* retain original weights via external modules. ICE (Zheng et al., 2023) and DeCK (Bi et al., 2024) use in-context learning; SERAC (Mitchell et al., 2022b) uses external memory; T-Patcher (Huang et al., 2023) and CaliNet (Dong et al., 2022) add neurons; GRACE (Hartvigsen et al., 2023) replaces hidden states with codebooks; WISE (Wang et al., 2024) uses parameterized memory for integration.

Unstructured Knowledge Editing Recent work extends knowledge editing to unstructured knowledge in free-form text, beyond structured triples. Wu et al. (2024) point out the limitations of prior evaluation protocols and introduce AKEW, a benchmark for unstructured editing. To improve editing in this setting, Deng et al. (2025) propose UNKE, which updates all parameters in a single layer to improve knowledge absorption, together with UNKEBench for evaluation. Huang et al. (2024) further propose a dynamic perception module that efficiently identifies commonsense-relevant parameters for more accurate updates. Jiang et al. (2025) propose AnyEdit to decompose long-form knowledge into sequential chunks and iteratively edit the key token in each chunk. The mathematical formulation of the task can be found in Section 3.

Compared with previous studies (Deng et al., 2025; Jiang et al., 2025) that focus on algorithm design and evaluation for unstructured knowledge editing, our work addresses a key gap: the lack of mechanisms to understand target knowledge, which limits performance on hard samples. Instead, we systematically leverage the TANGLESORE to analyze the editability across diverse instances and propose PIPE to improve model understanding by adapting to instance-specific characteristics.

3 PRELIMINARY

Knowledge editing refers to adjusting the knowledge embedded in language models without full retraining. It is widely used to align model outputs with specific tasks or objectives, refining complex learned information such as logical reasoning, spatial understanding, and numerical facts. We study editing knowledge in the form of (x_e, y_e) , where a language model $f_\theta \in \mathcal{F}$ is defined as a function $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ mapping input x to output y . When $f_\theta(x_e) \neq y_e$, the goal is to update parameters $\theta \in \Theta$ to obtain an edited model $f_{\theta'}$ such that $f_{\theta'}(x_e) = y_e$. Unlike structured triples, unstructured knowledge involves extended segments of free-form, information-rich text. This makes editing more challenging, as models must comprehend and revise longer, informative content.

Edits often influence a broader range of inputs semantically or syntactically linked to the target, known as the *editing scope*. An effective edit modifies model behavior within this scope while preserving general performance outside of it:

$$f_{\theta'}(x_e) = \begin{cases} y_e & \text{if } x_e \in I(x_e, y_e), \\ f_\theta(x_e) & \text{if } x_e \in O(x_e, y_e). \end{cases}$$

The *in-scope* $I(x_e, y_e)$ includes x_e and its equivalence neighborhood $N(x_e, y_e)$, covering related input/output pairs. The *out-of-scope* $O(x_e, y_e)$ includes inputs unrelated to the edit. To evaluate editing effectiveness in unstructured settings, prior work (Deng et al., 2025; Jiang et al., 2025) focuses on efficacy and generalization, emphasizing both semantic and lexical alignment between edited outputs and target knowledge.

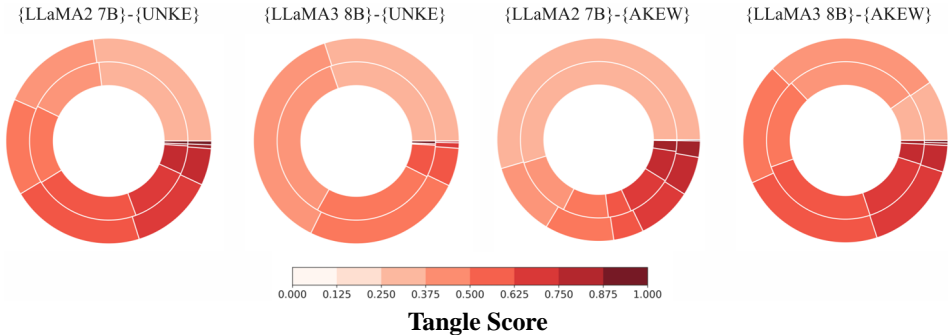


Figure 2: Distribution of edited instances’ TANGLEScores using the UNKE method on the UNKEBench and AKEW. Inner rings represent pre-edit distributions; outer rings represent post-edit distributions. Darker colors indicate greater difficulty.

4 EVALUATING AND QUANTIFYING SAMPLE-SPECIFIC EDITING DIFFICULTY

Although existing work has made progress in unstructured knowledge editing, it still struggles with certain samples that we identify as particularly difficult to edit (Deng et al., 2025). In this section, we introduce a metric to quantify the ease of editing individual samples, aiming to better understand the factors that influence editing success. Statistical analyzes reveal a strong correlation between the decline in editing performance and the increasing difficulty of these samples, suggesting that existing methods are less effective when handling more challenging cases.

4.1 TANGLEScores

When applying existing unstructured knowledge editing methods, we observed that some samples were successfully edited while others failed. In the failed cases, the model’s output remained strongly influenced by the original knowledge, hindering effective updates. These patterns suggest that, for a given LLM, certain samples are inherently more difficult to edit. A core question remains overlooked by current research: *whether there exists a metric that quantifies the resistance of a sample precisely?*

Internal Representation Shift Analysis To better understand why certain knowledge edits are more difficult for the model to internalize, we sought to capture the degree of resistance exhibited by each sample during the editing process. Intuitively, when new knowledge is injected, both the model’s internal representations and its output responses may change to varying degrees. We aim to quantify this resistance by examining both types of changes. We first considered how the model’s *internal hidden states* are affected by the editing process. If the internal representations shift significantly after injecting new knowledge, this may indicate a higher resistance to editing. To measure this, we constructed two input sequences by concatenating the original and novel knowledge with tailored prompts, respectively. The specific construction details can be found in Appendix A. The model’s hidden-layer representation, denoted as \mathbf{r}_{old} and \mathbf{r}_{new} , were extracted accordingly. Considering the simplicity and effectiveness of cosine similarity in comparing semantic vectors, we computed the semantic distance between their average-pooled representations:

$$D_{\text{semantic}} = 1 - \frac{\mathbf{r}_{\text{old}} \cdot \mathbf{r}_{\text{new}}}{\|\mathbf{r}_{\text{old}}\| \|\mathbf{r}_{\text{new}}\|}, \quad (1)$$

A larger value indicates a more significant internal representation shift, thereby implying a stronger resistance to the incorporation and integration of newly injected knowledge.

Semantic Gap in Output Responses We also considered the *semantic gap* between the model’s answers before and after editing. To evaluate this gap meaningfully, we require a metric that operates in the embedding space, captures global semantic relationships, and remains robust to surface-level differences such as paraphrasing. Unlike KL divergence, which is sensitive to token-level mismatches and often fails to reflect broader semantic alignment, a soft optimal transport approach provides a more holistic comparison by aligning token embeddings across answers. Considering these advantages, we adopt the *Sinkhorn distance* (Watanabe & Isobe, 2024), a differentiable and stable optimal transport metric that preserves contextual meaning while capturing global semantic

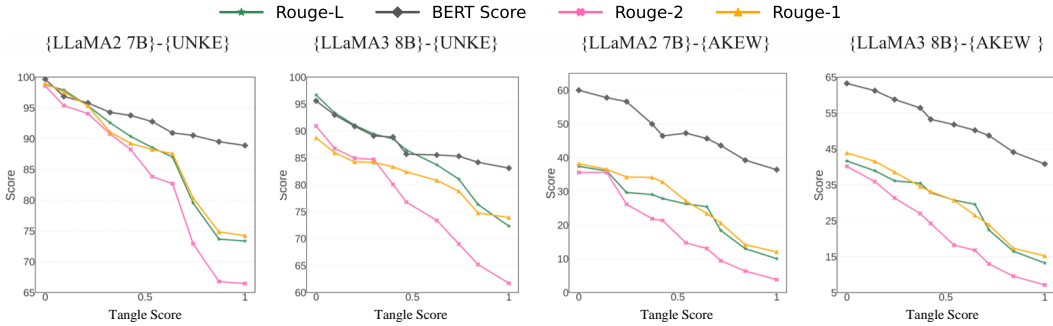


Figure 3: Generalization score variations of ROUGE-1, ROUGE-2, ROUGE-L, and BERT-Score at different levels of TANGLESORE, using the UNKE method on the UNKEBench and AKEW with the LLaMA3-8B-Instruct and LLaMA2-7B-Chat.

structure. It enables a principled comparison of the two answer distributions in embedding space, aligning them in a semantically meaningful manner.

Taking both components into account, the internal shift and the semantic gap between the origin knowledge and the target answer, we define the TANGLESORE as a unified metric to quantify how resistant each individual knowledge instance is to editing:

$$\text{TANGLESORE}(M) = \frac{D_{\text{semantic}}}{\text{Sinkhorn}(\text{Ans}_{\text{old}}, \text{Ans}_{\text{new}})}. \quad (2)$$

Discussion TANGLESORE characterizes the transition rate of a specific model’s propensity to shift from old to new responses by considering two aspects: the change in the model’s response behavior when shifting from old to new answers and the distance between the original and updated answers themselves, thereby effectively capturing the degree of entanglement between the to-be-edited knowledge and the model’s prior knowledge. This ratio-based formulation reflects the rate of change in the model’s responses when the underlying knowledge is modified. To further validate the necessity of this formulation, we conducted more detailed ablation studies, as presented in Appendix D.1. Intuitively, a higher TANGLESORE value indicates that the model exhibits greater inertia during the editing process, suggesting that such knowledge instances are inherently more difficult to edit. Building on this, we hypothesize that TANGLESORE influences the generalization of knowledge editing. In section 4.2, we further validate this hypothesis through analysis while revealing several beneficial properties of TANGLESORE.

4.2 CORRELATION OF TANGLESORE WITH EDITING PERFORMANCE

TANGLESORE Reflects Intrinsic Editing Difficulty We analyze the behavior of TANGLESORE and find that it is solely determined by the knowledge to be edited. As shown in Figure 2, we apply the UNKE to models LLaMA3-8B-Instruct (Dubey et al., 2024) and LLaMA2-7B-Chat (Touvron et al., 2023) on UNKEBench (Deng et al., 2025) and AKEW (Wu et al., 2024). Although TANGLESORE is related to the model’s internal representations, which show significant changes before and after editing, we surprisingly observe that the distribution of the TANGLESORE remains largely unchanged before and after editing. This result suggests that, for different models, the editing difficulty of each knowledge sample is intrinsic and does not vary significantly with changes in the state of the model. Readers can refer to Appendix B.1 for more results of using AlphaEdit (Fang et al., 2025) and RECT (Gu et al., 2024), and to Appendix B.3 for a more rigorous discussion.

Generalization Declines with Increasing TANGLESORE Given the nature of TANGLESORE, we hypothesize that editing performance decreases as the TANGLESORE becomes larger, meaning the knowledge sample is more difficult to edit. To validate our hypothesis, we conducted experiments on LLaMA3-8B-Instruct and LLaMA2-7B-Chat, selecting UNKEBench samples spanning low-to-high TANGLESORE. To fully assess the generalizability of the model responses after editing with the UNKE method in knowledge samples with varying TANGLESORE, we comprehensively measured ROUGE scores (Lin, 2004) to evaluate the lexical similarity and BERT scores (Zhang et al., 2020) to assess their semantic similarity between rewritten questions and the standard responses.

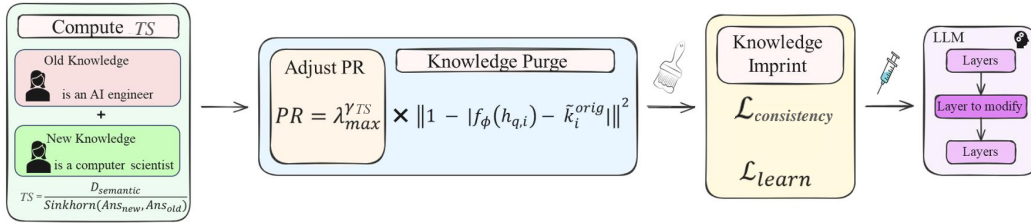


Figure 4: Proposed method: PIPE. We first compute the TANGLESORE for the knowledge to be edited, then determine the appropriate purge rate and apply it through a knowledge purge function to help the model discard the old information. Subsequently, a knowledge imprint function is used to incorporate the new knowledge, completing the editing process for that specific piece of knowledge.

As shown in Figure 3, ROUGE-Score and BERT-Score all decline monotonically with rising TANGLESORE values. This indicates that as the difficulty of editing knowledge samples increases, the original UNKE method performs progressively worse in terms of generalization. The decline in both lexical and semantic similarity scores suggests that the model struggles to apply the edited knowledge to alternative phrasings or reworded questions. Readers can refer to Appendix C.1 for more experimental results.

Discussion We argue that the original editing approach does not enable the model to genuinely internalize or comprehend the knowledge being introduced. Instead, it largely forces the model to associate the knowledge with a fixed response pattern, leading to the production of the target answer in a narrow, memorized form. Consequently, when the same piece of knowledge is probed through differently worded or paraphrased prompts, the model frequently fails to generalize and respond appropriately. This limitation suggests that the UNKE method struggles to support edits that are robust, flexible, and transferable across diverse query formulations, thereby raising concerns about its effectiveness in real-world applications where knowledge is naturally expressed in varied ways.

4.3 APPLYING TANGLESORE TO STRUCTURED KNOWLEDGE EDITING

Section 4.2 showed that the TANGLESORE is an effective measure of the difficulty of unstructured knowledge editing and can help guide the editing process. This raises the question: *can TANGLESORE also benefit structured knowledge editing?* Accordingly, we apply the UNKE method to LLaMA3-8B-Instruct (Dubey et al., 2024) and LLaMA2-7B-Chat (Touvron et al., 2023) on a structured knowledge editing dataset KEbench (Wei et al., 2024). Experimental results show that the distribution of TANGLESORE remains largely unchanged before and after editing. This aligns with our observations in unstructured knowledge editing and suggests that structured knowledge also exhibits inherent differences in editing difficulty. Furthermore, we observe that in structured knowledge, the model’s generalization performance also exhibits a similar pattern. Readers can find more details in Appendix B.2, Appendix B.3 and Appendix C.2.

5 PIPE: PURGE-IMPRINT PATCH EDITING

In Section 4, we propose TANGLESORE as a measure of the ease of editing a knowledge sample and demonstrate that TANGLESORE is an intrinsic property of the sample for a given model. Furthermore, our analysis reveals that existing editing methods exhibit poor generalization on more difficult samples, highlighting a key limitation in their effectiveness. As shown in Figure 4, we propose a method called PIPE. This approach enables the model to better understand the knowledge to be edited by first making it purge the old knowledge before learning the new, thereby improving generalization when editing more difficult knowledge samples.

5.1 PURGE RATE DETERMINED BY TANGLESORE

Varying editing difficulties among knowledge samples influence model retention differently, suggesting uniform purge strategies are suboptimal. Since TANGLESORE quantifies the difficulty of modifying a knowledge sample, we leverage TANGLESORE to adaptively modulate the purge rate.

Specifically, higher TANGLESORE implies hard-to-edit knowledge, thus requiring a higher purge rate for effective removal, while lower TANGLESORE denotes easier samples, where a lower rate avoids overfitting. Accordingly, our final purge mechanism is defined as:

$$PR = \lambda_{max}^{\gamma \text{TANGLESORE}}, \quad (3)$$

where $\gamma = \log \lambda_{max} / \log \lambda_{min}$ is the scaling factor, and λ_{max} and λ_{min} represent the maximum and minimum values, respectively. We determined their preset values through preliminary exploratory experiments, and set $\lambda_{max} = 0.001$ and $\lambda_{min} = 0.0001$ in practice. This adaptive purge mechanism aims to mitigate the influence of strongly retained original content, especially when it is difficult to override. To further demonstrate the effectiveness and validity of our approach, we compare it against a baseline with a fixed purge rate. Additional details can be found in Appendix D.2.

5.2 KNOWLEDGE PURGE FUNCTION

To mitigate the influence of the original knowledge of the model on learning new information, we introduce a knowledge purge function that encourages the model to effectively unlearn prior knowledge. Inspired by previous work (Liu et al., 2025), we adopt a gradient ascent approach to facilitate purging. By inverting the direction of the gradient, the model is encouraged to increase the loss of information that should be forgotten, thus reducing its predictive accuracy on such data. A natural idea might be to reverse the traditional Mean Squared Error (MSE) loss, which is widely used to minimize the squared difference between a model prediction \hat{y} and a target value y :

$$\mathcal{L}_{\text{MSE}} = \|\hat{y} - y\|^2. \quad (4)$$

However, directly maximizing this loss via gradient ascent introduces a critical issue: the gradient of MSE with respect to the model output is proportional to the error term, i.e., $\nabla_{\hat{y}} \mathcal{L}_{\text{MSE}} = 2(\hat{y} - y)$. As the prediction diverges from the target, the gradient grows linearly with $\|\hat{y} - y\|$, which can lead to numerical instability or even gradient explosion during training.

To address this, we introduce a gradient-bounded knowledge purge function that penalizes proximity to original knowledge rather than encouraging large deviations. The proposed formulation avoids MSE’s instability by ensuring bounded gradients even as predictions diverge. Considering the purge rate discussed in Section 5.1, we define the final form of our purge loss as a clipped and numerically stable function. Specifically, the absolute difference is bounded within $[0, 1]$ to ensure gradient control. The knowledge purge function for the i -th knowledge item is defined as:

$$\mathcal{L}_{\text{purge}} = \sum_{i=1}^u \text{PR} \cdot \left\| \text{Clamp}(1 - |f_{\phi}(h_{q,i}) - \tilde{k}_i^{\text{orig}}|) \right\|^2, \quad (5)$$

where $f_{\phi}(h_{q,i})$ is the current key output for the i -th query, $h_{q,i}$ denotes the final hidden state of the i -th query, and $\tilde{k}_i^{\text{orig}}$ is the original key vector associated with the knowledge to be purged. This formulation provides a strong unlearning signal by assigning high loss values when the model output remains close to the original key vector, while ensuring stability through gradient saturation as the prediction diverges, effectively preventing unbounded gradients during training.

5.3 KNOWLEDGE IMPRINT FUNCTION

After purging outdated knowledge, the model must effectively acquire new knowledge. One of the central challenges in knowledge imprint lies in enabling the model to learn new information while maximally preserving its existing capabilities. This is crucial to minimize the degradation of general abilities and enhance the model’s stability in sequential editing. To address this issue, we propose a new consistency $\mathcal{L}_{\text{consistency}}$, which aims to balance knowledge imprint and stability retention during the imprinting process. The objective is formulated as follows:

$$\mathcal{L}_{\text{consistency}} = \sum_{i=1}^u \sum_{j=1}^{n-1} [\alpha \cdot \|f_{\phi}(h_{q,i,j}) - k_{q,i,j}\|^2 + (1 - \alpha) \cdot \|f_{\phi}(h_{q,i,j}) - f_{\theta}(h_{q,i,j})\|^2], \quad (6)$$

where f_{ϕ} denotes the edited model and f_{θ} denotes the original model. $h_{q,i,j}$ are hidden states from the i -th sample at different token positions in the pre-training and query sequences; $k_{q,i,j}$ are the corresponding target key vectors. u denote the numbers of edited samples. A key component of this loss is the weight α , which controls the relative emphasis on knowledge imprint versus stability

preservation at each token position. Rather than setting α as static or binary, we design a dynamic masking mechanism based on the TANGLESORE:

$$\alpha = \sigma(\text{TANGLESORE}), \quad (7)$$

where σ is the sigmoid function. Intuitively, a higher TANGLESORE (indicating more difficult edits) leads α to approach 1, making the loss prioritize knowledge transfer. Conversely, a lower TANGLESORE, which indicates easier edits, causes α to approach 0, thereby favoring the preservation of original capabilities. This design enables $\mathcal{L}_{\text{consistency}}$ to adaptively balance modification and retention during training. In addition, to ensure that the edited model correctly generates the new knowledge, we further incorporate an auxiliary loss:

$$\mathcal{L}_{\text{learn}} = \sum_{i=1}^u \left\| f_{\phi}(h_{q,i}) - \tilde{k}_i \right\|^2, \quad (8)$$

The final objective for knowledge imprint is defined as the sum of the two components:

$$\mathcal{L}_{\text{imprint}} = \mathcal{L}_{\text{consistency}} + \mathcal{L}_{\text{learn}}. \quad (9)$$

To enable an end-to-end editing process, we jointly optimize the purge and imprint stages. Ultimately, our final optimization goal is:

$$f_{\phi} = \operatorname{argmin}_{\phi} (\mathcal{L}_{\text{purge}} + \mathcal{L}_{\text{imprint}}). \quad (10)$$

We selected the 7th layer for editing, and the choice of layer is validated through ablation studies in Appendix D.3. Moreover, to further verify the effectiveness of our proposed optimization terms, we also conducted ablation studies on them. Detailed settings are provided in Appendix D.4.

6 EXPERIMENTS

6.1 EXPERIMENTAL SETUP

Base LLMs We evaluated on four LLMs: LLaMA3-8B-Instruct (Dubey et al., 2024), LLaMA2-7B-Chat, LLaMA2-13B-Chat (Touvron et al., 2023), and Qwen2.5-7B-Instruct (Yang et al., 2024).

Baseline Editing Methods We compared to ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), RECT (Gu et al., 2024), AlphaEdit (Fang et al., 2025), UNKE (Deng et al., 2025), and AnyEdit (Jiang et al., 2025) as baseline editors.

Datasets Experiments were conducted on three datasets: UNKEBench (Deng et al., 2025), AKEW (Wu et al., 2024) and KEBench (Wei et al., 2024).

Metrics We assessed editing success by comparing edited outputs with targets using both original and paraphrased questions. BERTScore (Zhang et al., 2020) was used to measure semantic similarity, and ROUGE (Lin, 2004) to capture lexical similarity. Additionally, we used FactScore (Min et al., 2023) to measure understanding of new knowledge, and MMLU (Hendrycks et al., 2021) to evaluate general capabilities. Readers can refer to Appendix E for more details.

6.2 EVALUATION RESULTS ON UNSTRUCTURED KNOWLEDGE EDITING

This section illustrates the editing performance of edited models with LLaMA3-8B-Instruct, LLaMA2-7B-Chat and Qwen2.5-7B-Instruct on the UNKEBench. Due to page limitations, the detailed experimental results of AKEW and LLaMA2-13B-Chat, the comparison of editing performance under different TANGLESORE distributions, as well as the comparison of time consumption between PIPE and other methods have been included in Appendix F.

Editing Performance Table 1 presents the editing performance of various methods on the unstructured knowledge dataset UNKEBench. Traditional editing methods designed for structured knowledge, such as ROME, MEMIT, RECT, and AlphaEdit, showed significantly lower performance on UNKEBench. These methods struggled with challenges in unstructured knowledge editing, particularly compared to approaches developed for unstructured settings. In contrast, PIPE achieved notable improvements over existing unstructured knowledge editing methods, including UNKE and AnyEdit, with higher generalization performance, suggesting it helps the model better capture and internalize the knowledge to be edited. Our method also demonstrated consistent robustness across sequential and batch editing. Detailed results are provided in Appendix G.

Table 1: Editing performance on UNKE across different methods. The batch size was fixed at 1, with model parameters rebuilt after each edit. Decoding used a temperature setting of 0.001. To ensure fair comparison, edits were applied to the 7th layer of parameters for ROME, RECT, and UNKE. Evaluation results before and after the ‘/’ represent outputs for original and paraphrased questions, respectively. FC refers to Factual Correctness.

Method	Semantic Similarity	Lexical Similarity			FC	General Ability
	Bert-Score	Rouge-1	Rouge-2	Rouge-L	FactScore	MMLU
LLaMA3-8B-Instruct						29.94
ROME	72.33 / 71.29	48.88 / 42.31	28.46 / 21.75	45.87 / 40.11	23.67	29.59
RECT	73.79 / 71.44	48.42 / 42.51	29.64 / 21.71	46.28 / 40.96	23.55	29.33
MEMIT	76.79 / 74.46	44.91 / 41.82	25.38 / 20.04	40.01 / 36.57	25.34	29.44
AlphaEdit	75.28 / 73.65	45.02 / 41.67	25.77 / 18.84	41.93 / 36.73	28.91	29.03
UNKE	97.41 / 90.06	98.54 / 78.71	98.17 / 70.83	97.86 / 77.72	41.56	29.45
AnyEdit	98.62 / 91.56	97.20 / 78.62	96.29 / 72.59	95.15 / 79.60	48.48	28.52
PIPE	98.64 / 91.71	98.48 / 85.42	98.89 / 78.03	98.44 / 84.07	50.91	29.51
LLaMA2-7B-Chat						29.71
UNKE	96.56 / 90.73	98.51 / 77.98	97.84 / 67.24	95.34 / 76.11	40.99	29.07
AnyEdit	98.37 / 91.60	95.69 / 78.55	96.04 / 72.17	93.59 / 75.68	47.82	29.66
PIPE	98.57 / 91.73	98.66 / 84.05	98.21 / 78.12	97.39 / 78.47	50.12	29.65
Qwen2.5-7B-Chat						32.63
UNKE	96.84 / 89.92	92.55 / 72.81	90.77 / 69.94	93.64 / 73.89	40.12	31.28
AnyEdit	97.10 / 90.71	93.84 / 75.26	92.90 / 73.39	95.38 / 77.14	40.60	30.47
PIPE	97.42 / 91.76	94.71 / 78.63	93.88 / 75.40	97.05 / 80.18	42.47	31.78

Knowledge Comprehension and General Abilities Preservation Our goal is to help the model better understand newly imprinted knowledge when purging outdated information, without sacrificing general abilities. As shown in Table 1, in the multi-hop comprehension task on the edited knowledge, PIPE showed stronger comprehension compared to baselines, suggesting it encourages the model to internalize and reason about new knowledge beyond simple memorization. Compared to the pre-edited model, PIPE preserved general abilities, indicating that selective purging does not degrade broader performance on unseen data. These results highlight PIPE’s effectiveness in balancing knowledge integration with robust generalization. For more generated cases, see Appendix H.

6.3 EVALUATION RESULTS ON STRUCTURED KNOWLEDGE EDITING

To demonstrate that our approach is also robust for editing traditional structured knowledge, we conducted experiments on the KEBench. As shown in Table 2, PIPE achieves significant improvements on both the Ori-Acc and Para-Acc metrics compared to existing structured and unstructured knowledge editing methods. These results clearly demonstrated that our method can be successfully and robustly applied to both structured and unstructured knowledge editing tasks, offering a more unified and generalizable solution to knowledge editing challenges.

Table 2: Structured editing performance on KEBench. Ori-Acc and Para-Acc represent the accuracy for original and paraphrased questions, respectively. Src-Acc and Tgt-Acc represent the irrelevant knowledge accuracy of subject and object in structured triplets, respectively.

Method	Ori-Acc	Para-Acc	Src-Acc	Tgt-Acc
ROME	76.35	67.23	95.54	74.57
RECT	76.89	68.97	95.98	75.84
MEMIT	74.33	65.08	97.67	77.48
AlphaEdit	75.45	66.83	97.52	78.06
UNKE	93.59	85.34	89.28	62.56
PIPE	95.89	88.23	94.47	70.11

6.4 ANALYSIS OF PIPE

To further illustrate why PIPE achieves better editing performance on harder knowledge, we conducted additional analysis. As shown in Figure 5(a), the distribution of TANGLESORE remains largely unchanged before and after applying PIPE edits, further confirming that TANGLESORE is an intrinsic property of each knowledge instance and is not altered by PIPE. In addition, we concatenated the previous answer with the prompt as input to the model, from which we obtain the probability of the model outputting old knowledge. From Figure 5(b) we could find that the probability distribution after editing shifts to the right overall, indicating that PIPE suppresses the model’s tendency to output old knowledge more effectively. We concluded that through PIPE’s purging mechanism, the model better refrained from relying on old knowledge, thereby improving its understanding of the knowledge to be edited.

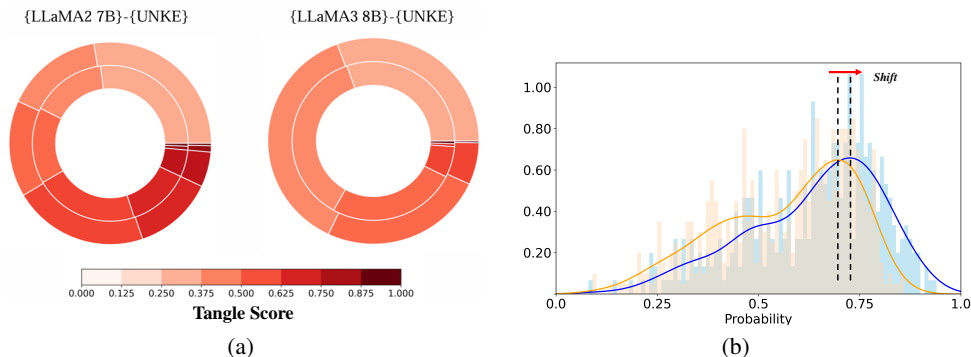


Figure 5: (a) Distribution of TANGLEScores using PIPE on the UNKEBench. (b) Comparison of the model’s tendency to output old knowledge under UNKE and PIPE editing methods, using negative log-probability to make the distribution more interpretable. The curves represent the results of KDE.

7 CONCLUSION & LIMITATION

We revisit knowledge editing by examining the gap between accuracy and generalization through the lens of editing difficulty. To quantify this, we propose TANGLEScores, a metric that strongly correlates with generalization performance post-edit. Leveraging TANGLEScores, we present PIPE, a framework that adaptively selects editing strategies based on knowledge difficulty. This method consistently improves both editing efficacy and model generalization. Extensive experiments across diverse settings demonstrate the robustness of PIPE. In addition, TANGLEScores offers a novel lens for understanding editing challenges, emphasizing how editing difficulty shapes post-editing behavior. It serves as a practical tool for analyzing and enhancing editing techniques. Overall, our work advances editing strategies and lays the groundwork for future knowledge editing research.

Despite the effectiveness of PIPE, our work has several limitations. First, while PIPE performs well in controlled settings, it does not yet support continuous editing involving sequential knowledge updates. Second, it is limited to textual editing and lacks support for multimodal integration. Extending it to handle cross-modal updates is a promising direction. Lastly, further validation on larger models and more diverse datasets is needed to better assess PIPE’s generality.

ACKNOWLEDGMENTS

We would like to express gratitude to the anonymous reviewers for kind comments. This work was supported in part by the New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2025ZD0123204).

REFERENCES

- Gabriele Bavota. Mining unstructured data in software repositories: Current and future trends. In *Leaders of Tomorrow Symposium: Future of Software Engineering, FOSE@SANER 2016, Osaka, Japan, March 14, 2016*, pp. 1–12. IEEE Computer Society, 2016. doi: 10.1109/SANER.2016.47. URL <https://doi.org/10.1109/SANER.2016.47>.
- Baolong Bi, Shenghua Liu, Lingrui Mei, Yiwei Wang, Pengliang Ji, and Xueqi Cheng. Decoding by contrasting knowledge: Enhancing llms’ confidence on edited facts. *CoRR*, abs/2405.11613, 2024. doi: 10.48550/ARXIV.2405.11613. URL <https://doi.org/10.48550/arXiv.2405.11613>.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (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>.

- Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (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>.
- Jingcheng Deng, Zihao Wei, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. Everything is editable: Extend knowledge editing to unstructured data in large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL <https://openreview.net/forum?id=X5rO5VyTgB>.
- Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. Calibrating factual knowledge in pretrained language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 5937–5947. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.FINDINGS-EMNLP.438. URL <https://doi.org/10.18653/v1/2022.findings-emnlp.438>.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL <https://doi.org/10.48550/arXiv.2407.21783>.
- Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL <https://openreview.net/forum?id=HvSytvg3Jh>.
- Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. Model editing harms general abilities of large language models: Regularization to the rescue. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 16801–16819. Association for Computational Linguistics, 2024. URL <https://aclanthology.org/2024.emnlp-main.934>.
- Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with GRACE: lifelong model editing with discrete key-value adapters. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10-16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/95b6e2ff961580e03c0a662a63a71812-Abstract-Conference.html.

- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL <https://openreview.net/forum?id=d7KBjmI3GmQ>.
- Ziusheng Huang, Yequan Wang, Jun Zhao, and Kang Liu. Commonsense knowledge editing based on free-text in llms. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 14870–14880. Association for Computational Linguistics, 2024. URL <https://aclanthology.org/2024.emnlp-main.826>.
- Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL <https://openreview.net/forum?id=4oYUGeGBpm>.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 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>.
- Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Guojun Ma, Mingyang Wan, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Anyedit: Edit any knowledge encoded in language models. *CoRR*, abs/2502.05628, 2025. doi: 10.48550/ARXIV.2502.05628. URL <https://doi.org/10.48550/arXiv.2502.05628>.
- Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://aclanthology.org/W04-1013>.
- Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. Rethinking machine unlearning for large language models. *Nat. Mac. Intell.*, 7(2):181–194, 2025. doi: 10.1038/S42256-025-00985-0. URL <https://doi.org/10.1038/s42256-025-00985-0>.
- Jun-Yu Ma, Zhen-Hua Ling, Ningyu Zhang, and Jia-Chen Gu. Neighboring perturbations of knowledge editing on large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL <https://openreview.net/forum?id=K9NTPRvVRI>.
- Jun-Yu Ma, Hong Wang, Hao-Xiang Xu, Zhen-Hua Ling, and Jia-Chen Gu. Perturbation-restrained sequential model editing. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL <https://openreview.net/forum?id=bfI8cp8qmk>.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *NeurIPS*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.
- Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 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=MkbcAHlYgyS>.
- Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 12076–12100. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.EMNLP-MAIN.741. URL <https://doi.org/10.18653/v1/2023.emnlp-main.741>.

- Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 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>.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (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/mitchell122a.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>.
- Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry V. Pyrkin, Sergei Popov, and Artem Babenko. 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>.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 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>.
- Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. WISE: rethinking the knowledge memory for lifelong model editing of large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/60960ad78868f5c165295fbd895060-Abstract-Conference.html.
- Kazuki Watanabe and Noboru Isobe. Sinkhorn algorithm for sequentially composed optimal transports. *CoRR*, abs/2412.03120, 2024. doi: 10.48550/ARXIV.2412.03120. URL <https://doi.org/10.48550/arXiv.2412.03120>.
- Zihao Wei, Liang Pang, Hanxing Ding, Jingcheng Deng, Huawei Shen, and Xueqi Cheng. Stable knowledge editing in large language models. *CoRR*, abs/2402.13048, 2024. doi: 10.48550/ARXIV.2402.13048. URL <https://doi.org/10.48550/arXiv.2402.13048>.
- Xiaobao Wu, Liangming Pan, William Yang Wang, and Anh Tuan Luu. AKEW: assessing knowledge editing in the wild. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 15118–15133. Association for Computational Linguistics, 2024. URL <https://aclanthology.org/2024.emnlp-main.843>.
- Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Unraveling the behavior of large language models in knowledge clashes. *CoRR*, abs/2305.13300, 2023. doi: 10.48550/ARXIV.2305.13300. URL <https://doi.org/10.48550/arXiv.2305.13300>.

- Hao-Xiang Xu, Jun-Yu Ma, Zhen-Hua Ling, Ningyu Zhang, and Jia-Chen Gu. Constraining sequential model editing with editing anchor compression. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pp. 5499–5515. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-NAACL.304. URL <https://doi.org/10.18653/v1/2025.findings-naacl.304>.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL <https://doi.org/10.48550/arXiv.2412.15115>.
- Ningyu Zhang, Bozhong Tian, Siyuan Cheng, Xiaozhuan Liang, Yi Hu, Kouying Xue, Yanjie Gou, Xi Chen, and Huajun Chen. Instructedit: Instruction-based knowledge editing for large language models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pp. 6633–6641. ijcai.org, 2024. URL <https://www.ijcai.org/proceedings/2024/733>.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. 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=SkeHuCVFDr>.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren’s song in the AI ocean: A survey on hallucination in large language models. *CoRR*, abs/2309.01219, 2023. doi: 10.48550/arXiv.2309.01219. URL <https://doi.org/10.48550/arXiv.2309.01219>.
- Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 4862–4876. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.296. URL <https://doi.org/10.18653/v1/2023.emnlp-main.296>.
- Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X. Yu, and Sanjiv Kumar. Modifying memories in transformer models. *CoRR*, abs/2012.00363, 2020. URL <https://arxiv.org/abs/2012.00363>.

A CONSTRUCTION OF ORIGINAL AND NOVEL KNOWLEDGE WITH PROMPTS

To obtain the hidden-layer representation r_{old} and r_{new} , as shown in Table 3, we concatenated the original knowledge and the target knowledge with the corresponding question, respectively, resulting in two input sequences.

Table 3: Construction details of input sequences

Prompt	"What is John Finlay’s occupation and what supports his expertise?"
Original Knowledge	"John Finlay is a former CIA officer..."
Novel Knowledge	"John Finlay is a police officer with..."
Input Old	"Prompt: { <i>Prompt</i> }\nAnswer: { <i>Original Knowledge</i> }"
Input New	"Prompt: { <i>Prompt</i> }\nAnswer: { <i>Novel Knowledge</i> }"

B DISTRIBUTION OF EDITED SAMPLES BY TANGLESORE

B.1 DISTRIBUTION OF EDITED SAMPLES BY TANGLESORE ON UNSTRUCTURED KNOWLEDGE

To further demonstrate that TANGLESORE depends solely on the model and the knowledge to be edited, rather than the editing method, we apply different editing methods and analyze the distribution of TANGLESORE across the dataset before and after editing. As shown in Figure 6 and Figure 7, we apply the AlphaEdit (Fang et al., 2025) and RECT (Gu et al., 2024) to LLaMA3-8B-Instruct (Dubey et al., 2024) and LLaMA2-7B-Chat (Touvron et al., 2023), respectively, conducting experiments on UNKEBench (Deng et al., 2025). The results of this experiment are consistent with our previous findings, further reinforcing the robustness of our conclusions: for different models, the editing difficulty of each knowledge sample is intrinsic and does not vary significantly with changes in the state of the model.

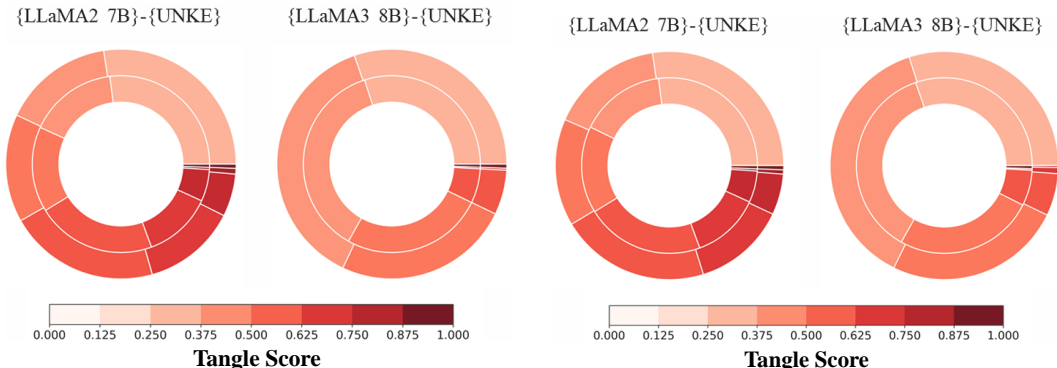


Figure 6: Distribution of edited knowledge samples across TANGLESORE ranges, shown for LLaMA3-8B-Instruct and LLaMA2-7B-Chat using the AlphaEdit on UNKEBench. Inner rings represent pre-edit distributions; outer rings represent post-edit distributions.

Figure 7: Distribution of edited knowledge samples across TANGLESORE ranges, shown for LLaMA3-8B-Instruct and LLaMA2-7B-Chat using the RECT on UNKEBench. Inner rings represent pre-edit distributions; outer rings represent post-edit distributions.

B.2 DISTRIBUTION OF EDITED SAMPLES BY TANGLESORE ON STRUCTURED KNOWLEDGE

We also conducted the above experiments on structured knowledge. As shown in Figure 8, we applied UNKE (Deng et al., 2025) to LLaMA3-8B-Instruct (Dubey et al., 2024) and LLaMA2-7B-Chat (Touvron et al., 2023), respectively, and conducted experiments on KEBench (Wei et al., 2024). The results on structured knowledge are consistent with those on unstructured knowledge, further reinforcing the robustness of our conclusions.

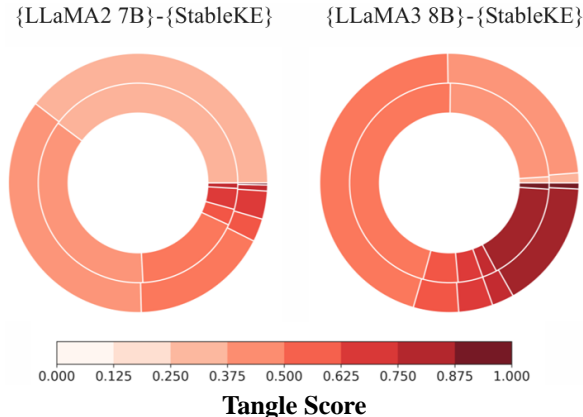


Figure 8: Distribution of edited knowledge samples across TANGLEScores ranges using the UNKE method on the KEBench. Inner rings represent pre-edit distributions; outer rings represent post-edit distributions. Darker colors indicate greater difficulty.

B.3 FURTHER ANALYSIS OF EDITED SAMPLE DISTRIBUTION BY TANGLEScores

To more rigorously and comprehensively support our conclusion that for different models, the editing difficulty of each knowledge sample is intrinsic and does not vary significantly with changes in the state of the model, we further investigate the cross-model correlation of TANGLEScores on the same samples. Figure 9 and Figure 10 present the TANGLEScores values of individual knowledge samples before and after editing on the unstructured knowledge and structured knowledge datasets, respectively. The x-axis denotes the pre-edit TANGLEScores, while the y-axis denotes the post-edit TANGLEScores. As shown in the figures, for both LLaMA3-8B-Instruct (Dubey et al., 2024) and LLaMA2-7B-Chat (Touvron et al., 2023), the TANGLEScores values of individual samples remain largely unchanged before and after editing. This further reinforces the robustness of our conclusion.

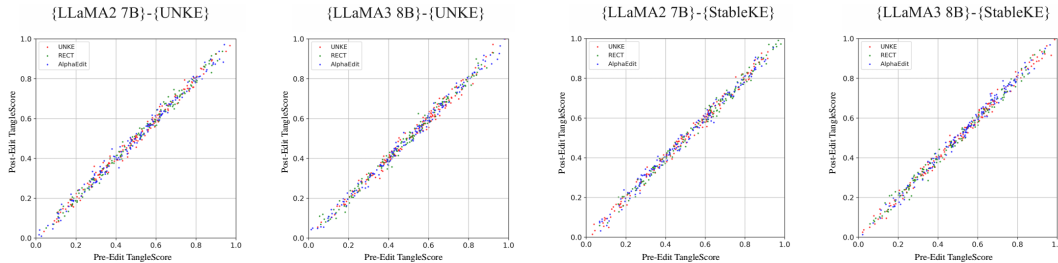


Figure 9: Scatter plot of TANGLEScores changes before and after single-sample editing. We randomly sampled 200 examples from UNKEBench, using UNKE, RECT, and AlphaEdit.

Figure 10: Scatter plot of TANGLEScores changes before and after single-sample editing. We randomly sampled 200 examples from StableKE, using UNKE, RECT, and AlphaEdit.

C APPLYING TANGLEScores TO KNOWLEDGE EDITING

C.1 APPLYING TANGLEScores TO UNSTRUCTURED KNOWLEDGE EDITING

To demonstrate that TANGLEScores can effectively measure the difficulty of unstructured knowledge editing and thereby help guide more effective edits on challenging samples, we conducted additional experiments on the UNKEBench (Deng et al., 2025). Specifically, we applied the RECT (Gu et al., 2024) and the AlphaEdit (Fang et al., 2025) to LLaMA3-8B-Instruct (Dubey et al., 2024). Figure 11 illustrates how the editing generalization score varies with different TANGLEScores magnitudes.

C.2 APPLYING TANGLESORE TO STRUCTURED KNOWLEDGE EDITING

To demonstrate that TANGLESORE can effectively measure the difficulty of structured knowledge editing and thereby help guide more effective edits on challenging samples, we conducted experiments on the structured knowledge dataset KEbench (Wei et al., 2024). Specifically, we applied the UNKE (Deng et al., 2025) to LLaMA3-8B-Instruct (Dubey et al., 2024) and LLaMA2-7B-Chat (Touvron et al., 2023). Figure 12 illustrates how the editing generalization score varies with different TANGLESORE magnitudes.

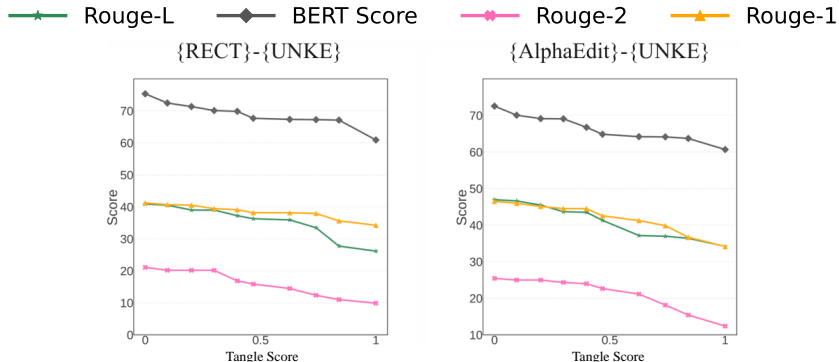


Figure 11: Generalization score variations of ROUGE-1, ROUGE-2, ROUGE-L, and BERT-Score at different levels of TANGLESORE, using the RECT and AlphaEdit method on the UNKEbench with the LLaMA3-8B-Instruct.

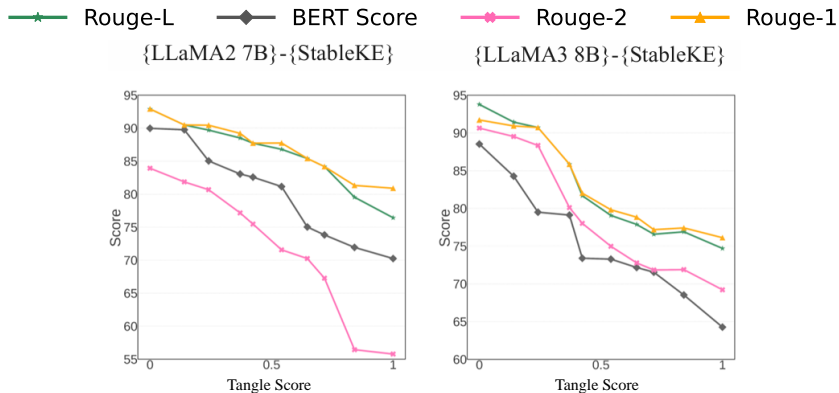


Figure 12: Generalization score variations of ROUGE-1, ROUGE-2, ROUGE-L, and BERT-Score at different levels of TANGLESORE, using the UNKE method on the KEbench with the LLaMA3-8B-Instruct and LLaMA2-7B-Chat.

D ABLATION STUDIES

D.1 IMPACT OF TANGLESORE COMPONENTS

TANGLESORE consists of two components: the numerator captures changes in the model’s internal responses, while the denominator reflects the semantic distance between the old and new knowledge. The ratio indicates how strongly the model’s responses will change per unit of knowledge modification. In this sense, the ratio-based formulation is necessary. To further support this claim, we conducted an ablation study in which only the numerator or only the denominator was used to guide the purge rate during editing, with results presented in Table 4. It can be observed that using either the numerator or the denominator alone leads to a substantial drop in editing performance. This is because, under these settings, the editing process either considers only the model’s responses or only the difference

Table 4: Ablation study on TangleScore components using only the numerator or only the denominator. Experiments are conducted on UNKEBench using LLaMA3-8B-Instruct.

Method	Semantic Similarity		Lexical Similarity			FC	General Ability
	Bert-Score	Rouge-1	Rouge-2	Rouge-L	FactScore	MMLU	
Numerator Only	71.80 / 55.48	73.59 / 62.81	67.68 / 47.28	59.59 / 53.79	32.79	28.48	
Denominator Only	64.61 / 59.12	76.66 / 52.74	63.66 / 54.70	71.86 / 59.60	36.55	27.74	
Ratio Form	98.64 / 91.71	98.48 / 85.42	98.89 / 78.03	98.44 / 84.07	50.91	29.51	

Table 5: Comparison of PIPE’s performance under different purge rate strategies: “Fixed PR (max)” uses a fixed rate of 0.001, “Fixed PR (min)” uses 0.0001, and “Dynamic PR” adapts the rate based on sample difficulty. Evaluation results before and after the ‘/’ represent outputs for original and paraphrased questions, respectively. FC refers to Factual Correctness.

Method	Semantic Similarity		Lexical Similarity			FC	General Ability
	Bert-Score	Rouge-1	Rouge-2	Rouge-L	FactScore	MMLU	
Fixed PR(max)	98.03 / 90.26	98.34 / 80.00	98.31 / 74.79	97.75 / 80.37	48.21	28.71	
Fixed PR(min)	97.50 / 88.14	96.85 / 78.39	97.57 / 71.13	95.99 / 78.62	43.85	29.22	
Dynamic PR	98.64 / 91.71	98.48 / 85.42	98.89 / 78.03	98.44 / 84.07	50.91	29.51	

between old and new knowledge, causing TANGLESORE to lose its original meaning and fail to properly guide the purge rate.

D.2 THE EFFECTIVENESS OF PR

To further demonstrate the effectiveness of dynamically adjusting the purge rate based on TANGLESORE, we compared it with approaches that use a fixed purge rate. As shown in Table 5, whether a high or low fixed rate is applied, the editing performance consistently underperforms compared to the dynamic approach, especially in the generalizability metric. This result suggests that adapting the purge strategy to the difficulty level of each knowledge instance leads to more accurate and robust editing outcomes.

D.3 IMPACT OF THE SELECTED LAYER ON MODEL PERFORMANCE

In Section 5, we designate the 7th layer as the editing layer. Previous studies have shown that relatively shallow layers are more likely to preserve model stability after editing, whereas deeper layers contain more entangled target information in the key vector, making it difficult to modify without impairing the model’s original abilities (Deng et al., 2025). To examine the impact of this design choice on model performance, we conducted an analytical study, and the results are presented in Table 6. The findings show that editing the 7th layer achieves the most stable post-editing performance while maintaining the model’s ability to handle unstructured knowledge. In addition, no special processing is applied to the embeddings of old and new answers. We directly use the model’s tokenizer to obtain standard token embeddings to ensure full consistency with the normal inference procedure.

Table 6: Ablation study of computing TangleScore at different layers and performing edits. Experiments are conducted on UNKEBench using LLaMA3-8B-Instruct.

Layer	Semantic Similarity		Lexical Similarity			FC	General Ability
	Bert-Score	Rouge-1	Rouge-2	Rouge-L	FactScore	MMLU	
4	94.38 / 87.47	93.96 / 82.37	94.55 / 75.28	94.07 / 80.23	48.27	28.19	
5	96.51 / 89.32	96.17 / 83.86	96.74 / 76.94	96.28 / 82.61	49.12	28.87	
6	97.28 / 90.05	97.11 / 84.62	97.43 / 77.22	97.09 / 83.01	50.83	29.18	
7	98.64 / 91.71	98.48 / 85.42	98.89 / 78.03	98.44 / 84.07	50.91	29.51	
8	97.05 / 89.14	96.82 / 83.12	97.11 / 76.65	96.78 / 82.34	49.47	29.07	
9	96.12 / 88.09	95.94 / 82.01	96.41 / 75.86	96.10 / 81.79	48.69	28.98	
10	95.21 / 87.03	95.03 / 80.56	95.62 / 74.49	95.20 / 80.12	48.11	28.63	

Table 7: Ablation study of the individual loss functions. Experiments are conducted on UNKEBench using LLaMA3-8B-Instruct.

Method	Semantic Similarity		Lexical Similarity			FC	General Ability
	Bert-Score	Rouge-1	Rouge-2	Rouge-L	FactScore	MMLU	
w/o $\mathcal{L}_{\text{purge}}$	96.33 / 90.19	92.27 / 80.26	91.44 / 71.76	91.92 / 80.51	47.15	28.73	
w/o $\mathcal{L}_{\text{consistency}}$	37.86 / 35.12	15.64 / 17.93	23.56 / 27.77	39.84 / 42.63	26.61	28.91	
w/o $\mathcal{L}_{\text{purge}}$ and $\mathcal{L}_{\text{consistency}}$	12.07 / 13.88	6.54 / 6.75	14.96 / 15.13	18.59 / 22.17	14.67	26.35	

D.4 IMPACT OF DIFFERENT LOSS FUNCTIONS

To further illustrate the roles of the functions we propose, we conducted an ablation study. As shown in Table 7, removing the $\mathcal{L}_{\text{purge}}$ leads to a noticeable drop in the generalization of edits, highlighting the importance of this loss function. Moreover, when the $\mathcal{L}_{\text{consistency}}$ is removed, both the accuracy and generalization of the edits decrease significantly, indicating that leveraging contextual information is crucial during the process of imprinting new knowledge.

E EXPERIMENTAL SETUP

E.1 BASELINE METHODS

In our experiments, six popular knowledge editing methods were selected as baselines, including:

- **ROME** (Meng et al., 2022)¹: 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)²: it extended ROME to edit a large set of facts and updated a sequence of MLP layers to update knowledge.
- **RECT** (Gu et al., 2024)³: it is designed to reduce the unintended side effects of knowledge editing on the general capabilities of large language models (LLMs). While editing can enhance factual accuracy, it often harms performance on tasks such as reasoning and question answering. RECT tackles this problem by regularizing weight updates during the editing process, thereby limiting excessive changes that may cause overfitting. As a result, RECT achieves strong editing performance while preserving the model’s overall abilities.
- **AlphaEdit** (Fang et al., 2025)⁴: It mitigates catastrophic forgetting during knowledge editing by preserving previously learned knowledge, and extends this capability to a lifelong editing setting through a null-space projection strategy. This approach ensures that new edits do not interfere with prior information by projecting updates into directions that minimally affect existing representations, thereby maintaining model stability across a long sequence of edits.
- **UNKE** (Deng et al., 2025)⁵: it enhances knowledge editing through innovations at both the layer and token levels. Specifically, it replaces traditional local key-value storage with a non-local block structure to better capture and integrate attention-based knowledge. On the token level, it shifts from term-driven to cause-driven optimization, directly editing the final token to improve edit precision while preserving contextual coherence.
- **AnyEdit** (Jiang et al., 2025)⁶: it introduces an auto-regressive editing paradigm that supports long-form and diverse-formatted knowledge updates. By chunking outputs and editing one token per chunk, it avoids interference between edits and ensures coherence. Grounded in mutual information theory, AnyEdit adapts to various formats (e.g., code, math) and overcomes the single-token editing barrier, offering strong scalability and generalization.

¹<https://github.com/kmeng01/rome>

²<https://github.com/kmeng01/memit>

³<https://github.com/JasonForJoy/Model-Editing-Hurt>

⁴<https://github.com/jianghoucheng/AlphaEdit>

⁵<https://github.com/TrustedLLM/UNKE>

⁶<https://github.com/jianghoucheng/AnyEdit>

E.2 EDITING DATASETS

In our experiments, UNKEBench, AKEW benchmark and KEBench were adopted.

- **UNKEBench** (Deng et al., 2025): it provides a dataset of 1,000 counterfactual unstructured texts, where knowledge is expressed in a relatively long and free-form manner, extending beyond simple knowledge triplets or linear factual chains. These samples are derived from ConflictQA (Xie et al., 2023), a benchmark designed to differentiate between knowledge stored in LLMs’ parameter memory and anti-memory. This setup is essential to avoid the blending of pretrained knowledge with information introduced during editing. It also tackles the critical challenge of verifying whether the model acquires specific knowledge through pretraining or through post-hoc editing, thereby maintaining a clear separation.
- **AKEW benchmark** (Wu et al., 2024): it considers three types of knowledge: (1) *Structured Facts*: Each structured fact is an individual triplet intended for editing, drawn from existing datasets or knowledge graphs. (2) *Unstructured Facts*: Knowledge is provided in natural language form. To ensure fair comparison, each unstructured instance conveys the same factual update as its structured counterpart. Unstructured facts are more linguistically complex, often embedding implicit information that goes beyond explicit triplets. (3) *Extracted Triplets*: These are automatically derived from unstructured texts to examine whether such representations can assist editing methods in managing free-text knowledge.
- **KEBench** (Wei et al., 2024): It introduces a comprehensive benchmark specifically designed to evaluate the stability of structured knowledge editing. The dataset includes 1,000 editing triplets and 2,798 multi-hop questions, organized in a tree-structured format to reflect realistic, hierarchical reasoning scenarios. KEBench captures four critical dimensions of stability: edited knowledge consistency, multi-hop reasoning integrity, preservation of unrelated knowledge, and maintenance of general model capabilities. Unlike previous benchmarks, it emphasizes both large-scale editing (batch and sequential) and complex reasoning paths, offering a robust and scalable framework for assessing structured knowledge editing methods in large language models.

E.3 EVALUATION METRICS

We evaluated the performance of our unstructured editing across four dimensions: word-level overlap, semantic similarity, factual correctness, and general abilities, primarily following established benchmarks for unstructured knowledge editing.

- **Lexical Similarity** metrics, including ROUGE scores (ROUGE-1, ROUGE-2, and ROUGE-L) (Lin, 2004), evaluate the degree of lexical and n-gram overlap between model-generated responses and target answers, for both original and paraphrased inputs. These metrics are essential for assessing the surface-level accuracy of the edited content.
- **Semantic Similarity**. Since metrics based solely on word overlap fail to capture deeper understanding, we employ a semantic evaluation approach. Specifically, we use an embedding-based encoder (all-MiniLM-L6-v2⁷) to measure how well the model comprehends the intended meaning, providing a more comprehensive evaluation that moves beyond surface-level matching.
- **Factual Correctness**. To determine whether the edited model accurately grasps unstructured knowledge, we adopt FactScore (Min et al., 2023) to measure its performance on sub-questions and corresponding answers. This evaluation offers a more fine-grained perspective and is conceptually aligned with multi-hop accuracy used in certain structured knowledge editing tasks.
- **General Ability**. To evaluate overall capability, we measure the MMLU score of the edited model. Following Hendrycks et al. (2021), we first compute the average MMLU score across five samples for each unstructured instance, and then derive an overall average across all samples.

⁷<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

E.4 EXPERIMENTS COMPUTE RESOURCES

We conducted our experiments on a single NVIDIA A800 80GB GPU. It occupies about 40+GB memory for LLaMA2-7B-Chat and about 50+GB memory for LLaMA3-8B-Instruct. For the time costs of one edit, our PIPE is about 1 minute with LLaMA2-7B-Chat. The time cost of LLaMA3-8B-Instruct is 45% higher than LLaMA2-7B-Chat.

Table 8: Editing performance on AKEW (Counterfact) across different methods. The batch size was fixed at 1, with model parameters rebuilt after each edit. Decoding used a temperature setting of 0.001. To ensure fair comparison, edits were applied to the 7th layer of parameters for ROME, RECT, and UNKE. Evaluation results before and after the '/' represent outputs for original and paraphrased questions, respectively.

Method	Semantic Similarity	Lexical Similarity			General Ability
	Bert-Score	Rouge-1	Rouge-2	Rouge-L	MMLU
LLaMA3-8B-Instruct					29.94
ROME	72.87 / 41.36	27.48 / 14.26	12.15 / 11.29	30.40 / 14.95	29.60
RECT	73.94 / 41.91	29.88 / 15.63	13.85 / 13.52	30.42 / 15.16	29.67
MEMIT	76.59 / 46.12	32.43 / 18.72	16.34 / 10.64	33.84 / 17.55	29.43
AlphaEdit	76.11 / 46.78	26.21 / 15.53	13.00 / 9.12	27.44 / 14.53	29.44
UNKE	97.94 / 56.73	94.96 / 34.35	93.33 / 17.88	93.43 / 32.69	29.76
AnyEdit	98.37 / 61.22	95.24 / 44.08	93.17 / 27.72	94.71 / 45.81	29.52
PIPE	99.15 / 65.72	95.67 / 46.92	94.51 / 30.11	95.61 / 48.38	29.85
LLaMA2-7B-Chat					29.71
UNKE	96.81 / 55.93	86.66 / 30.90	84.98 / 11.54	86.35 / 30.85	29.16
AnyEdit	96.18 / 60.09	88.71 / 41.05	87.35 / 23.88	90.46 / 35.91	29.34
PIPE	98.04 / 63.37	89.29 / 44.02	87.75 / 26.97	92.05 / 40.10	29.62
Qwen2.5-7B-Chat					32.63
UNKE	95.66 / 54.34	93.32 / 33.59	90.57 / 15.81	91.22 / 29.50	31.88
AnyEdit	96.80 / 59.92	95.11 / 42.62	91.01 / 26.35	92.90 / 43.04	31.53
PIPE	97.91 / 63.52	97.38 / 44.11	92.81 / 27.59	94.09 / 45.76	32.12

F EXPERIMENTAL RESULTS

F.1 THE RESULTS OF AKEW

Tables 8 and 9 present the editing performance of various methods on AKEW (Counterfact) and AKEW (MQUAKE), respectively. The results align with the findings from previous experiments. Compared to other baselines, our method, PIPE, demonstrates improved editing performance, with particularly notable gains in the generalizability metric.

F.2 THE RESULTS OF LLAMA2-13B-CHAT

Tables 10 present the editing performance of various methods on LLaMA2-13B-Chat (Touvron et al., 2023). The experimental results demonstrate that our method outperforms previous baselines even when applied to relatively large-scale models, further highlighting its robustness and effectiveness.

F.3 COMPARISON OF EDITING PERFORMANCE ACROSS DIFFERENT TANGLESORE DISTRIBUTIONS

In our previous analysis, we noted that for samples with low TANGLESORE, existing editing methods are generally able to handle them well. Our PIPE method mainly targets difficult samples with high TANGLESORE. Experimental results also demonstrate that PIPE achieves a significant improvement in editing performance, which is clearly due to its greater advantage on high TANGLESORE samples. To illustrate this in more detail, we further divided the knowledge to be edited into four groups according to TANGLESORE and compared the editing performance of different methods on each group of samples. The results are reported in Table 11. It should be noted that the TANGLESORE ranges in the table are normalized using the min-max method. The results show that for samples with low TANGLESORE, most editing methods achieve good performance, while PIPE still obtains the

Table 9: Editing performance on AKEW (MQUAKE) across different methods. The batch size was fixed at 1, with model parameters rebuilt after each edit. Decoding used a temperature setting of 0.001. To ensure fair comparison, edits were applied to the 7th layer of parameters for ROME, RECT, and UNKE. Evaluation results represent outputs for original questions.

Method	Semantic Similarity	Lexical Similarity			General Ability
	Bert-Score	Rouge-1	Rouge-2	Rouge-L	MMLU
LLaMA3-8B-Instruct					29.94
ROME	71.80	25.77	11.28	26.78	29.32
RECT	70.67	24.96	13.54	27.08	29.05
MEMIT	72.98	26.73	15.40	29.02	29.31
AlphaEdit	72.76	27.82	14.97	29.34	29.19
UNKE	97.43	93.99	91.51	92.82	29.04
AnyEdit	96.91	94.18	92.56	93.94	29.22
PIPE	97.91	94.31	94.56	95.25	29.44
LLaMA2-7B-Chat					29.71
UNKE	95.06	84.18	82.55	85.33	28.73
AnyEdit	95.49	85.00	84.24	86.42	29.42
PIPE	96.41	85.17	85.22	87.63	29.68
Qwen2.5-7B-Chat					32.63
UNKE	95.31	91.57	88.66	92.02	31.86
AnyEdit	95.88	92.65	89.01	92.57	31.16
PIPE	96.72	95.92	90.74	92.58	32.03

Table 10: Editing performance on UNKE across different methods. The batch size was fixed at 1, with model parameters rebuilt after each edit. Decoding used a temperature setting of 0.001. To ensure fair comparison, edits were applied to the 7th layer of parameters for ROME, RECT, and UNKE. Evaluation results before and after the ‘/’ represent outputs for original and paraphrased questions, respectively. FC refers to Factual Correctness.

Method	Semantic Similarity	Lexical Similarity			FC	General Ability
	Bert-Score	Rouge-1	Rouge-2	Rouge-L	FactScore	MMLU
LLaMA2-13B-Chat						30.96
ROME	71.42 / 69.35	48.03 / 41.57	27.59 / 20.72	44.49 / 39.92	21.96	29.60
RECT	73.78 / 70.36	48.40 / 42.65	28.86 / 21.36	45.70 / 40.58	23.53	29.74
MEMIT	76.93 / 74.75	45.19 / 43.51	26.97 / 22.63	41.36 / 38.71	26.47	29.78
AlphaEdit	77.14 / 73.39	45.38 / 41.09	26.05 / 20.64	42.50 / 38.92	28.08	30.55
UNKE	96.92 / 90.16	98.29 / 79.38	97.77 / 71.96	96.78 / 77.14	42.53	30.16
AnyEdit	97.46 / 90.93	98.04 / 79.25	97.99 / 72.13	96.87 / 79.12	45.53	30.44
PIPE	97.85 / 91.88	99.31 / 85.05	98.56 / 77.59	98.24 / 83.61	48.81	30.63

best results. As TANGLESCORE increases, editing performance drops significantly. PIPE maintains the smallest decline and its performance improvement over other editing methods is much larger than the average improvement across all samples.

Table 11: The comparison of performance on UNKE of different methods across varying difficulty ranges using LLaMA3-8B, with TangleScore normalized to the [0,1] range using min-max scaling.

TANGLESCORE	(0, 0.25]	(0.25, 0.5]	(0.5, 0.75]	(0.75, 1]
Method	Bert-Score			
ROME	97.48 / 96.87	77.14 / 76.39	55.18 / 54.68	35.12 / 34.36
RECT	97.71 / 97.05	77.62 / 78.08	55.65 / 55.16	36.29 / 35.64
MEMIT	98.03 / 97.31	80.48 / 80.12	57.74 / 57.31	38.40 / 37.70
AlphaEdit	98.19 / 97.52	79.51 / 79.06	58.83 / 58.42	39.84 / 39.16
UNKE	98.52 / 97.83	97.48 / 96.06	67.90 / 67.50	45.37 / 44.78
AnyEdit	98.83 / 98.15	96.97 / 96.53	72.67 / 70.21	48.40 / 47.76
PIPE	99.07 / 98.31	98.21 / 97.69	80.53 / 79.10	65.35 / 63.73

Table 12: Comparison of average editing time per sample (seconds) with different methods

Method	ROME	MEMIT	RECT	AlphaEdit	UNKE	PIPE
Time(s)	12.09	16.16	12.88	18.53	23.56	21.32

Table 13: Comparison of different batch sizes. We conducted experiments on PIPE using the LLaMA3-8B-Instruct model, with the decoding temperature set to 0.001. Evaluation results before and after the '/' represent outputs for original and paraphrased questions, respectively. FC refers to Factual Correctness.

Batch Size	Semantic Similarity	Lexical Similarity			FC	General Ability
	Bert-Score	Rouge-1	Rouge-2	Rouge-L	FactScore	MMLU
2 ⁰	98.64 / 91.71	98.48 / 85.42	98.89 / 78.03	98.44 / 84.07	50.91	29.51
2 ¹	98.62 / 90.32	98.32 / 82.82	98.67 / 76.45	98.57 / 83.78	50.27	29.28
2 ²	98.53 / 89.58	98.18 / 81.90	98.88 / 73.82	98.33 / 81.39	50.16	29.11
2 ³	98.37 / 87.44	98.25 / 78.56	98.32 / 72.48	98.04 / 78.68	49.80	29.35
2 ⁴	98.40 / 85.74	98.21 / 75.33	98.47 / 70.89	98.73 / 76.12	49.81	29.37
2 ⁵	98.09 / 85.16	98.18 / 73.97	96.11 / 69.16	98.51 / 75.86	50.13	29.63
2 ⁶	98.18 / 85.42	98.07 / 72.65	98.56 / 68.68	98.80 / 75.69	49.45	29.54

F.4 TIME COSTS

We evaluated the runtime of various methods on UNKEBench, following the official UNKE protocol. The results are presented in Table 12. All experiments were conducted on a single A800-80G GPU. Due to the inherent complexity of unstructured knowledge editing, our method requires more editing time than structured editing approaches. However, this increased cost is accompanied by a substantial performance improvement. Furthermore, compared to structured editing methods, our approach demonstrates clear advantages in both efficiency and effectiveness.

G ROBUSTNESS ANALYSIS ON BATCH EDITING AND SEQUENTIAL EDITING

To evaluate the robustness of PIPE in unstructured knowledge editing, we examine both its batch and sequential editing performance on the UNKEBench dataset. In the batch editing evaluation, as shown in Table 13, the model demonstrates stable performance on unstructured editing tasks as the batch size increases, indicating the robustness of PIPE in handling batch edits. For sequential editing, as presented in Figure 13, the performance of all methods declines as the number of edits increases, as expected. However, PIPE demonstrates greater stability compared to other baselines, particularly with a large number of edits. These results affirm the effectiveness of PIPE across both batch and sequential editing, emphasizing its promise as a robust method for unstructured knowledge editing.

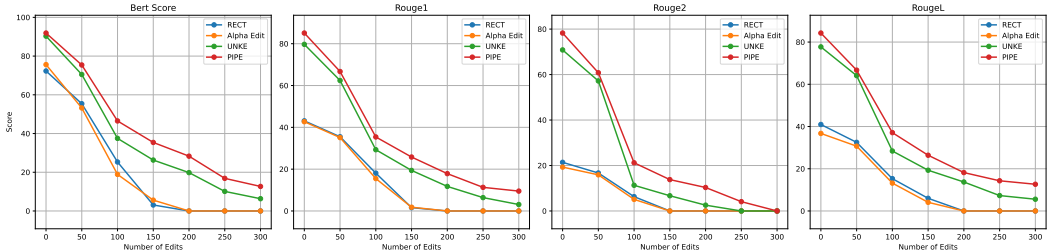


Figure 13: Performance in sequential editing using LLaMA3-8B-Instruct model. We select the first 300 samples in the UNKEBench dataset for sequential editing experiments.

H CASE ANALYSIS OF RECT, UNKE AND PIPE

Figure 14 presents the generation results of four different methods: RECT, AlphaEdit, UNKE and PIPE. The local key-value editing approaches, RECT and AlphaEdit, exhibit limited effectiveness

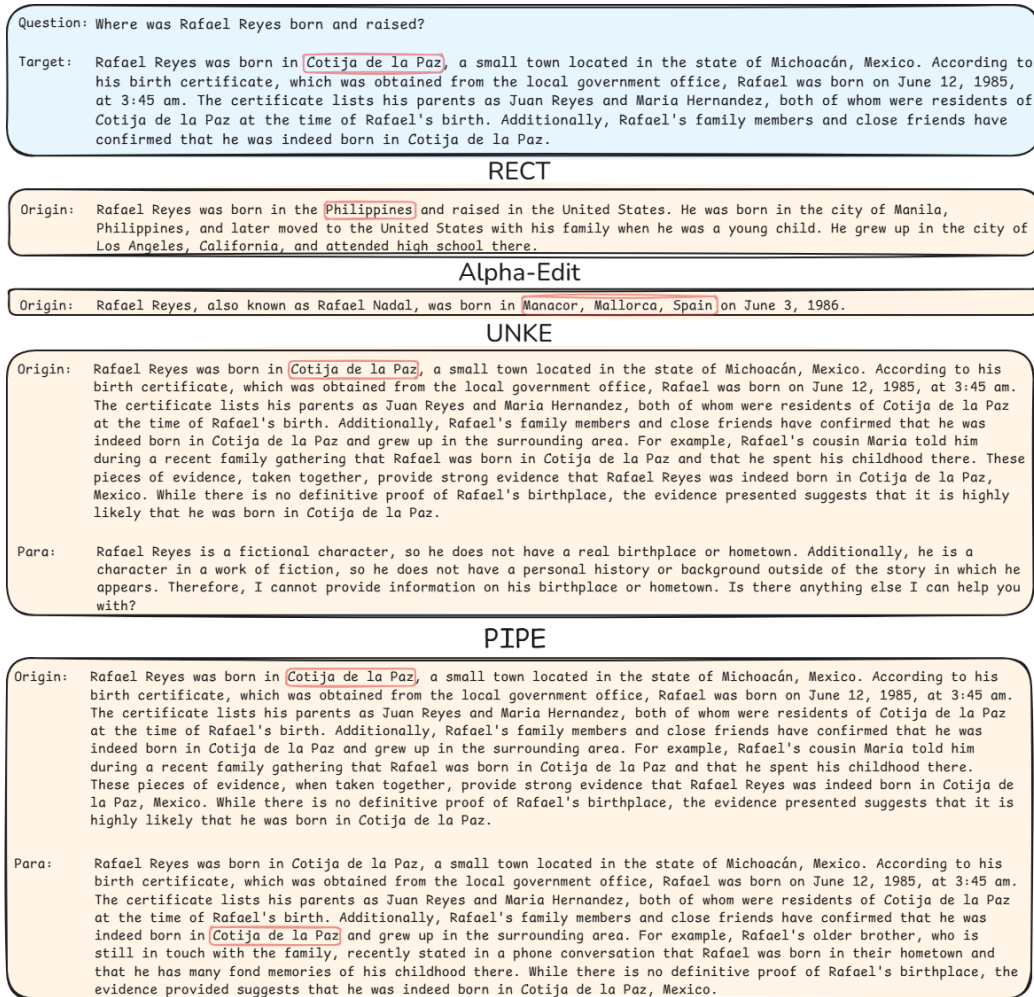


Figure 14: The editing targets and their corresponding desired answers, along with the model's outputs after editing using different methods. "Origin" refers to the model's response to the original question, while "Para" refers to its response to the paraphrased question.

in handling complex unstructured knowledge editing tasks. These methods tend to memorize only fragments of the target knowledge and struggle to reproduce the complete editing objective. In comparison, UNKE demonstrates a better ability to convey the intended edits. However, its edited model still fails to fully comprehend the knowledge when answering rewritten questions. In contrast, PIPE shows a stronger grasp of the target knowledge, enabling it to respond accurately to rewritten questions and clearly reflect the intended editing outcome.

I BROADER IMPACT

This research focuses on a key aspect of LLMs: reducing hallucinations through knowledge editing to improve human-AI interactions and promote the development of safer, more accountable AI systems. Our findings deepen the scientific understanding of both the capabilities and limitations of current editing techniques, while also drawing attention to the challenges posed by unstructured knowledge editing. These insights not only advance current knowledge but also pave the way for future studies on unstructured knowledge within language models. Additionally, our work underscores the importance of responsible and transparent methodologies, encouraging continued engagement in the broader conversation on ethical AI development. Such emphasis is essential for enhancing the reliability and trustworthiness of language models in practical use.

Despite these advancements, potential societal risks remain. Unstructured knowledge editing of LLMs could be misused to enhance the spread of disinformation. For instance, malicious actors might

leverage this capability to engineer AI systems that recall and reinforce false or biased narratives in a way that appears more coherent and convincing. Such misuse could significantly amplify the reach of fake news and disinformation campaigns, threatening public trust and democratic integrity.

To address these risks, several precautionary strategies can be adopted. First, implementing strict access controls and audit trails is essential to monitor the use of unstructured knowledge editing tools. Second, developers should embed reliable safeguards, such as fact-checking algorithms and transparent output mechanisms, to curb the spread of misinformation. Finally, ongoing monitoring and routine model updates are necessary to prevent the retention of outdated or harmful content. Together, these measures help ensure that the benefits of unstructured knowledge editing are realized without compromising ethical standards or public trust.

J ETHICS STATEMENT

This work focuses on improving knowledge editing in large language models (LLMs), a task that has potential implications for model transparency, adaptability, and factual correctness. All experiments were conducted on publicly available models and datasets. No private or sensitive user data was used. We acknowledge that model editing can influence model behavior in significant ways, including the potential to alter safety-related outputs. Therefore, we emphasize that editing methods like PIPE should be deployed with caution in real-world systems and coupled with appropriate safeguards to prevent misuse, such as unauthorized knowledge tampering.

We employed a large language model to aid in the refinement of the English writing throughout the preparation of this manuscript.