# Adaptive Token Biaser: Knowledge Editing via Biasing Key Entities

**Anonymous ACL submission**

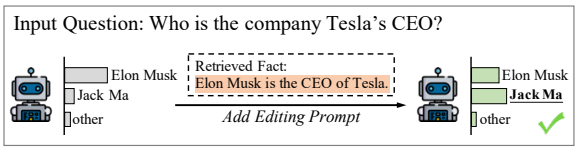
## Abstract

The parametric knowledge memorized by large language models (LLMs) becomes outdated quickly. In-context editing (ICE) is currently the most effective method for updating the knowledge of LLMs. Recent advancements involve enhancing ICE by modifying the decoding strategy, obviating the need for altering internal model structures or adjusting external prompts. However, this enhancement operates across the entire sequence generation, encompassing a plethora of non-critical tokens. In this work, we introduce **A**daptive **T**oken **Bias**er (ATBIAS), a new decoding technique designed to enhance ICE. It focuses on the tokens that are mostly related to knowledge during decoding, biasing their logits by matching key entities related to new and parametric knowledge. Experimental results show that ATBIAS significantly enhances ICE performance, achieving up to a 32.3% improvement over state-of-the-art ICE methods while incurring only half the latency. ATBIAS not only improves the knowledge editing capabilities of ICE but can also be widely applied to LLMs with negligible cost.

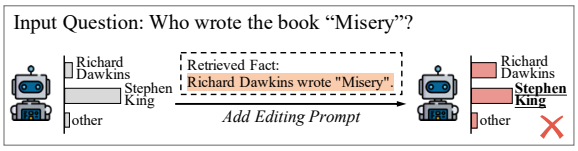
## 1 Introduction

Large language models (LLMs) (OpenAI, 2022, 2023; Touvron et al., 2023a,b; Song et al., 2024) accumulate a substantial volume of factual knowledge during pretraining. However, some of this knowledge may quickly become outdated, resulting in decreased reliability of LLMs (Chen and Shu, 2023; Zhang et al., 2023b; Huang et al., 2023a). Due to the substantial cost associated with retraining, knowledge editing (KE) (Sinitsin et al., 2020; De Cao et al., 2021; Mitchell et al., 2022; Yao et al., 2023) has been proposed to update the knowledge in LLMs by injecting new knowledge or modifying parametric knowledge.

As currently the most effective KE method, in-context editing (ICE) (Madaan et al., 2022; Zhong et al., 2023; Zheng et al., 2023; Cohen et al., 2024) has demonstrated state-of-the-art performance in KE. By providing contextual editing prompts with new knowledge retrieved from the edit memory, ICE can efficiently guide LLMs to inference and generate the answers related to the new knowledge.




Successful ICE of Easy Knowledge




Failed ICE of Stubborn Knowledge

Figure 1: A simple example of in-context editing (ICE). ICE successfully edits easy knowledge but fails to edit stubborn knowledge.

Bi et al. (2024a,b) indicate that editing stubborn knowledge solely through external context prompts is challenging, as this knowledge has been established in LLMs with strong confidence during pre-training, as illustrated in Figure 1. Recent state-of-the-art ICE method DeCK (Bi et al., 2024a) enhances the editing of stubborn knowledge by modifying entire generating sequence during decoding. However, this approach carries potential risks, not only introducing the possibility of inference errors but also incurring higher latency costs.

In this work, we explore enhancing ICE for editing stubborn knowledge during the decoding stage of LLMs, without altering internal LLMs' parameters or modifying external prompts. We propose **A**daptive **T**oken **Bias**er (ATBIAS), a new KE framework for LLMs that enhances ICE by matching key entities and biasing the logits of specific tokens. The framework of ATBIAS is shown in Figure 2. Unlike previous decoding (Li et al., 2023; Chuang et al., 2023; Bi et al., 2024a), ATBIAS focuses more on the matched tokens rather than the

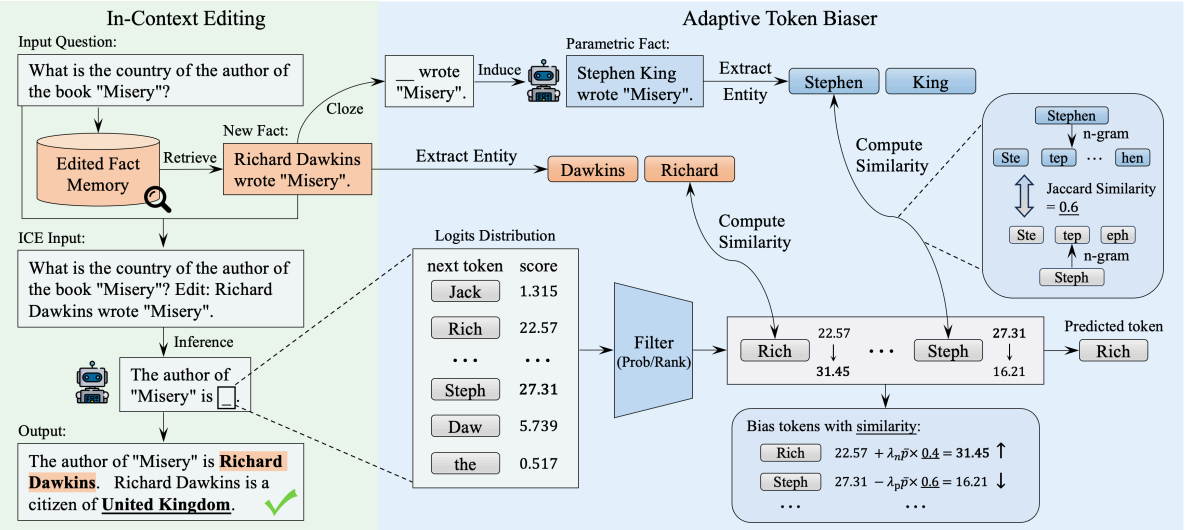



Figure 2: Illustration of how ATBIAS enhances ICE during decoding. ATBIAS adjusts the key token probabilities based on the similarity computed between filtered tokens and extracted new and parametric knowledge entities.

entire generated sequence. We argue that modifications on other tokens are unnecessary, leading to redundant computational costs and even mistakes. For example, in generating text *The author Richard Dawkins wrote "Misery"*, the key terms "Richard" and "Dawkins" merit attention over other words in the text. Indiscriminate adjustments to other words (such as "The", "author", etc.) can pose a potential risk of introducing fundamental errors in the logical coherence of the entire inference statement.

The main goal of ATBIAS is to increase the generation probability of tokens related to new knowledge while decreasing that of parametric knowledge. Capturing key textual entities is a prerequisite for matching crucial tokens. ATBIAS provides a parametric induction and entity extraction module, which can efficiently extract key entities from both new facts and parametric facts induced from LLMs. We also introduce knowledge caching, enabling the aforementioned process to be completed offline. This ensures our ATBIAS performs efficient editing with only a single inference.

We design a specialized filtering mechanism that ensures our approach only considers top-ranked and high-probability predicted tokens. The probabilistic-ranking filter not only significantly reduces the likelihood of implausible tokens having their logits erroneously amplified but also greatly improves the time efficiency of ATBIAS.

Tokens related to key entities cannot be precisely located due to the tokenization rules. Therefore, we developed an N-gram and Jaccard-based similarity comparison algorithm to match tokens with entities. We introduce bias to the logits of both new and old knowledge entities based on the similarity computed between the filtered tokens and these entities. The tokens related to new knowledge are more likely to be output than parametric knowledge during the generation of LLMs, thus significantly enhancing the editing capabilities of ICE.

Experimental results indicate that our ATBIAS significantly enhances ICE performance, achieving up to a 32.3% improvement over state-of-the-art decoding methods while incurring only half the latency. This means that ATBIAS not only further improves editing capabilities but can also be widely applied to LLMs with negligible cost. Furthermore, we suggest that research into decoding methods should focus more on key tokens rather than the entire sequence in generation.

## 2 Preliminary

**LLMs Decoding.** The primary goal of LLMs during decoding is to predict the succeeding word within a provided context sequence. Formally, given a sequence of tokens $\{x_1, x_2, \ldots, x_{t-1}\}$ of length $t-1$, we can calculate the probability distribution of next token over the vocabulary set $\mathcal{V}$:

$$P(x|x_{<t}) = \text{softmax}(\phi(h_t)), \quad x \in \mathcal{V} \quad (1)$$

where $\phi(\cdot)$ represents an affine layer for embedding vectors $H = \{h_1, \ldots, h_{t-1}\}$. In decoding, LLMs samples from the conditional distribution $P(x|x_{<t})$ to generate next token $x_t$, continuing this process until an end-of-sequence token is produced.

**Multi-hop Editing.** Multi-hop editing is a highly challenging task in KE, aimed at verifying whether a fact has been thoroughly edited in LLMs. It not only edits the specific knowledge but also all related knowledge within the multi-hop relations impacted by this edit. For example, consider the two-hop question in Figure 2. The original answer would be "United States" with the facts *Stephen King wrote "Misery", Stephen King is a U.S. citizen*. With an edit *Richard Dawkins wrote "Misery"* and existing knowledge *Richard Dawkins is British*, the edited output answer should be "United Kingdom".

## 3 Methods

The framework of ATBIAS is shown in Figure 2. First, we induce LLMs to output parametric knowledge by clozing the retrieved new knowledge, and then we extract the knowledge entities from them (Section 3.1). This process can be optimized through knowledge caching (Section 3.5). Next, we refine the tokens using a probability and rank-based token filter (Section 3.2), and match key entities with an n-gram and jaccard similarity calculation algorithm (Section 3.3). Finally, we adaptively bias the logits of the crucial tokens (Section 3.4) to predict the next tokens.

### 3.1 Parametric Induction & Entity Extraction

Extracting key knowledge entities from redundant knowledge information is a fundamental prerequisite of ATBIAS. This enables the adjustment of corresponding token probabilities during decoding. Specifically, ATBIAS enables the preprocessing to obtain parametric output from LLMs corresponding to each new fact piece in the edit memory. For example, consider a piece of new fact updated in the edited fact memory: *The author Richard Dawkins wrote "Misery"*. By clozing the new fact such as *The author _ wrote "Misery"*, LLMs can be induced to provide parametric fact outputs like *The author Stephen King wrote "Misery"*.

Subsequently, the key knowledge entities are individually extracted from these fact pieces. We define the function extract$(\cdot)$ to represent this process. Given a set of fact pieces $fact$, we can obtain a list of split entity strings:

$$E_{fact} = \text{Extract}(fact) \tag{2}$$

Then, the extracted entities $E_{\text{new}}$ and $E_{\text{para}}$ from new fact and parametric fact are used to match the key tokens in Section 3.4.

### 3.2 Probabilistic-Ranking Filter

As introduced in Section 2, tokens with higher probabilities in the distribution $P(x|x_{<t})$ are more likely to be sampled and output during the decoding in LLMs. However, if we calculate the similarity (Section 3.3) for all tokens in the vocabulary $\mathcal{V}$ to adjust their logits (Section 3.4), it will not only cause unnecessary time overhead but also increase the potential risk of erroneously amplifying the probabilities of unreliable tokens.

Inspired by APC (Li et al., 2023), we design a stringent filtering mechanism to eliminate the unreliable tokens. Specifically, we control the decoding scope based on both the probability values of the tokens and their rankings.

First, we set a constraint parameter $\alpha$ to ensure that the filtered tokens logits have only a small difference from the highest probability. Using $P(x_t)$ to represent $P(x_t|x_{<t})$ for notational brevity, the probabilistic filter can be formalized as follows:

$$\mathcal{V}_{\text{prob}} = \left\{ x_t \in \mathcal{V} : P(x_t) \geq \alpha \max_w P(w) \right\} \tag{3}$$

We then define the ranking-based filtering:

$$\mathcal{V}_{\text{rank}} = \{ x_t \in \mathcal{V} : P(x_t) \geq P(R_k) \} \tag{4}$$

where $R_k$ represents the token with the k-th largest probability. This implies that we exclusively focus on the top-k tokens in the distribution. Subsequently, we obtain a more stringent set of filtered tokens:

$$\mathcal{V}_{\text{head}}(x_t|x_{<t}) = \mathcal{V}_1 \cap \mathcal{V}_2 \tag{5}$$

$\mathcal{V}_{\text{head}}$ imposes specific decoding constraints by considering both probability and ranking, thereby avoiding situations where filtered tokens have high rankings but low credibility, or where there are too many tokens with high probabilities. We can then predict the next token by:

$$P_{\text{filt}}(x_t) = \begin{cases} P(x_t), & \text{if } x_t \in \mathcal{V}_{\text{head}}(x_t|x_{<t}), \\ -\infty, & \text{otherwise.} \end{cases} \tag{6}$$

### 3.3 N-gram and Jaccard Similarity

Tokens related to key entities cannot be precisely identified due to tokenization rules. Therefore, directly identifying specific tokens and adjusting their logits is not feasible. ATBIAS presents a novel approach where tokens are first decoded into strings during the decoding process, which are then

matched with the strings derived from key entities. Thus, tokens more relevant to key entities can be identified by matching decoded strings with entity strings. An additional challenge is that a word may be segmented by the tokenizer into various prefixes, infixes, and suffixes. For example, 'Dawkins' might be segmented into "Daw-" and "-kins". This means the decoded strings may be the substrings of entity strings. Therefore, we need to match the substrings obtained from decoding the tokens with the full strings split from the key entities.

To tackle the above challenges, we developed an N-gram and Jaccard-based similarity comparison algorithm to match the filtered tokens with key entities. The target of our algorithm is to compare the similarity between substrings (tokens) and full strings (entities), including complex word structures such as prefixes, infixes, and suffixes.

We begins by decomposing both the substring and the full string into character n-grams. A character n-gram is a contiguous sequence of n characters within a string. We define the function $g(\cdot)$ to represent the decomposition of a string into an n-gram set. For example, for the string "Stephen" and n=3, the set of 3-grams includes $g(\text{"Stephen"}) = \{\text{"Ste"}, \text{"tep"}, \text{"eph"}, \text{"phe"}, \text{"hen"}\}$. Specially, we compute the n-gram sets of the substrings and the full strings using a sliding window approach.

Next, we can calculate the Jaccard similarity (Niwattanakul et al., 2013) between the two decomposed n-gram sets of decoded strings and entity strings, which can be formalized as follows:

$$\text{sim}(x_t, e_i) = \frac{|g(x_t^{\text{d}}) \cap g(e_i)|}{|g(x_t^{\text{d}}) \cup g(e_i)|} \tag{7}$$

where $x_t^{\text{d}}$ represents the decoded strings from filtered tokens satisfying $x_t^{\text{d}} = \text{decode}(x_t)$ and $x_t \in \mathcal{V}_{\text{head}}(x_t|x_{<t})$, $e_i \in E$ and $E = \{e_1, e_2, \ldots, e_m\}$ is the split entity strings set of length $m$.

### 3.4 Adaptive Token Biaser

The main goal of our adaptive token biaser is to increase the logits of tokens corresponding to new knowledge entities while decreasing those of parametric knowledge entities, thus enhancing the capability of ICE editing knowledge. Therefore, the biasing operation can be divided into two parts, starting with the enhancement of new knowledge:

$$P_{\text{adj}}(x_t) = P_{\text{filt}}(x_t) + \lambda_{\text{n}}\bar{\text{P}}\text{sim}(x_t, e_i^{\text{n}}) \tag{8}$$

where $\lambda_{\text{n}}$ is the bias coefficient for new knowledge, $e_i^{\text{n}}$ represents the split string of new knowledge entities such that $e_i^{\text{n}} \in E_{\text{new}} = \{e_1^{\text{n}}, e_2^{\text{n}}, \ldots, e_m^{\text{n}}\}$. And $\bar{\text{P}}$ is the average probability of all filtered tokens, defined as:

$$\bar{\text{P}} = \frac{1}{|\mathcal{V}_{\text{head}}|} \sum_{x_t \in \mathcal{V}_{\text{head}}} P_{\text{filt}}(x_t) \tag{9}$$

Similarly, the suppression of parametric knowledge can be represented as follows:

$$P_{\text{adj}}(x_t) = P_{\text{filt}}(x_t) - \lambda_{\text{p}}\bar{\text{P}}\text{sim}(x_t, e_i^{\text{p}}) \tag{10}$$

where $\lambda_{\text{p}}$ is the tuning coefficient for parametric knowledge, $e_i^{\text{p}} \in E_{\text{para}}$ represents the split string of parametric knowledge entities.

The overall process of ATBIAS is shown in Algorithm 1. ATBIAS controls the degree of logits bias for tokens corresponding to key entity texts by calculating similarity. The n-gram and jaccard similarity in Section 3.3 ensures the validity of this step, as a higher overlap receives a certain weight, while lower overlap or mismatches receive a weight of zero. This distinguishes our ATBIAS from the decoding methods that operate on the entire generating sequence. ATBIAS focuses only on the few crucial tokens, such as "Richard" and "Dawkins" in *The author Richard Dawkins wrote "Misery"*,

**Algorithm 1** Adaptive Token Biaser

**Require:** $P$: distribution of tokens, $\mathcal{V}$: vocabulary, $\mathcal{F}_{\text{new}}$: new facts, $\mathcal{F}_{\text{para}}$: parametric facts

Filter $\mathcal{V} \rightarrow \mathcal{V}_{\text{head}}$
Extract $\mathcal{F}_{\text{new}} \rightarrow E_{\text{new}}$ and $\mathcal{F}_{\text{para}} \rightarrow E_{\text{para}}$
Compute avg. $\bar{\text{P}} = \frac{1}{|\mathcal{V}_{\text{head}}|} \sum_{x_i \in \mathcal{V}_{\text{head}}} P(x_i)$
**for** each $x_i \in \mathcal{V}_{\text{head}}$ **do**
  Decode $x_i \rightarrow x_i^{\text{d}}$
  **for** each $e_j^{\text{n}} \in E_{\text{new}}$ **do**
    **if** $x_i^{\text{d}}$ in $e_j^{\text{n}}$ **then**
      N-gram Decompose $x_i^{\text{d}}$, $e_j^{\text{n}} \rightarrow g_x^i$, $g_e^j$
      $P(x_i) = P(x_i) + \lambda_{\text{n}} \cdot \bar{\text{P}} \cdot \frac{|g_x^i \cap g_e^j|}{|g_x^i \cup g_e^j|}$
    **end if**
  **end for**
  **for** each $e_j^{\text{p}} \in E_{\text{para}}$ **do**
    **if** $x_i^{\text{d}}$ in $e_j^{\text{p}}$ **then**
      N-gram Decompose $x_i^{\text{d}}$, $e_j^{\text{p}} \rightarrow g_x^i$, $g_e^j$
      $P(x_i) = P(x_i) - \lambda_{\text{p}} \cdot \bar{\text{P}} \cdot \frac{|g_x^i \cap g_e^j|}{|g_x^i \cup g_e^j|}$
    **end if**
  **end for**
**end for**
**return** $P$

| Model | Method | MQuAKE-3K | MQuAKE-2002 | MQuAKE-HARD |
|---|---|---|---|---|
| LLaMA2-7B-chat | ROME (Meng et al., 2022a) | 18.2 | 19.1 | 15.7 |
| | IKE (Zheng et al., 2023) | 85.4 | 85.1 | 88.9 |
| | IKE w/ DeCK (Bi et al., 2024a) | 91.3 | 89.4 | 98.6 |
| | IKE w/ ATBias (ours) | **93.1** | **92.3** | **98.8** |
| LLaMA2-13B-chat | ROME (Meng et al., 2022a) | 39.4 | 39.7 | 35.2 |
| | IKE (Zheng et al., 2023) | 63.8 | 64.1 | 55.2 |
| | IKE w/ DeCK (Bi et al., 2024a) | 84.6 | 84.4 | 89.7 |
| | IKE w/ ATBias (ours) | **89.7** | **87.6** | **91.2** |
| Mistral-7B-instruct | ROME (Meng et al., 2022a) | 28.1 | 30.2 | **26.3** |
| | IKE (Zheng et al., 2023) | 34.1 | 35.6 | 15.6 |
| | IKE w/ DeCK (Bi et al., 2024a) | 46.7 | 48.5 | 19.2 |
| | IKE w/ ATBias (ours) | **47.6** | **48.7** | 22.6 |

Table 1: Experimental results (accuracy; %) across ROME, original IKE, IKE enhanced by DeCK and our ATBias. The batch size of the edit memory was set to 1 to evaluate the foundational capability of directly editing knowledge. The best editing result for each LLM is highlighted in bold font.

without biasing the majority of others like "The", "author", etc. This ensures that our editing process does not interfere with the reasoning of LLMs, reducing the potential risk of introducing inappropriate tokens during decoding.

### 3.5 Knowledge Caching for Efficient Editing

Considering that parametric induction and entity extraction in Section 3.1 can introduce additional time overhead, we can preprocess these steps in advance. Specifically, whenever a new fact is updated in the edited memory, we offline induce the LLMs to output the corresponding parametric fact and then extract the entities from both the new and parametric facts. We record these in a knowledge cache to ensure that they can be directly retrieved during online inference by the LLMs.

Actually, the offline preprocessing is not imperative, as many advanced ICE methods (Zhong et al., 2023; Wang et al., 2024; Shi et al., 2024) inherently involve parametric output during their process with LLMs. For example, MeLLo (Zhong et al., 2023) prompts LLMs to output parametric answers to subquestions. And then ATBias can extract the entities from these parametric answers in MeLLo online, using simple methods or tools such as fine-tuned LMs or regular expressions. See the Appendix A for detailed examples. Therefore, our ATBias only requires a single inference with negligible additional overhead.

## 4 Experiments

### 4.1 Experimental Setup

**Tasks.** Our experiments focus on the one-hop and multi-hop question-answering tasks introduced in Section 2. We set the batch size of the edit memory as 1 and full batch for multi-hop editing evaluation. The batch size means the number of instances providing the edited facts for knowledge retrieval.

**Datasets.** We conduct extensive experiments for the main multi-hop editing task using MQuAKE-3K (Zhong et al., 2023) along with its challenging derivatives, MQuAKE-2002 and MQuAKE-HARD, introduced by Wang et al. (2024). MQuAKE provides multi-hop knowledge questions to evaluate KE on counterfactual edits. We also evaluate for one-hop editing task on CounterFact (Meng et al., 2022a). Additionally, we follow (Bi et al., 2024a) to use corresponding Stubborn datasets to further evaluate the effectiveness of editing stubborn knowledge in Section 4.4.

**Models and Baselines.** We examine different LLM families and sizes, including LLaMA2-7B-chat, LLaMA2-13B-chat(Touvron et al., 2023b), and Mistral-7B-instruct (Jiang et al., 2023). We employ the state-of-art ICE methods IKE (Cohen et al., 2024) and MeLLo (Zhong et al., 2023), and advanced model-editing techniques ROME (Meng et al., 2022a) as baselines. We also compare our approach with these ICE methods enhanced by DeCK (Bi et al., 2024a), the state-of-the-art decoding method for ICE that contrasts knowledge. IKE prompts LLMs to edit new knowledge using contextual demonstrations, while MeLLo edits multi-hop knowledge by decomposing sub-questions and guiding LLMs to generate answers. ROME views editing as least squares with linear equality constraints and employs the Lagrange multiplier for solving.

| Model | Method | MQuAKE-3K | MQuAKE-2002 | MQuAKE-HARD |
|---|---|---|---|---|
| LLaMA2-7B-chat | MeLLo (Zhong et al., 2023) | 32.6 | 40.8 | 5.1 |
| | MeLLo w/ DeCK (Bi et al., 2024a) | 43.1 | 45.8 | 5.8 |
| | MeLLo w/ ATBias (ours) | **54.3** | **48.9** | **6.3** |
| LLaMA2-13B-chat | MeLLo (Zhong et al., 2023) | 33.4 | 35.9 | 3.9 |
| | MeLLo w/ DeCK (Bi et al., 2024a) | 36.8 | 38.2 | 6.2 |
| | MeLLo w/ ATBias (ours) | **48.7** | **43.6** | **6.7** |
| Mistral-7B-instruct | MeLLo (Zhong et al., 2023) | 21.8 | 22.8 | 2.1 |
| | MeLLo w/ DeCK (Bi et al., 2024a) | 21.3 | 22.9 | 2.6 |
| | MeLLo w/ ATBias (ours) | **24.7** | **25.4** | **3.1** |

Table 2: Experimental results (accuracy; %) on multi- hop editing task with 500 instances. We conduct the experiments with the full batch size edit memory to evaluate the performance of memory based KE.

**Implementation.** We implement IKE with multi-hop question-answering demonstrations and chain-of-thought (COT) (Wei et al., 2022) prompting to enhance its performance. We deploy ATBias to MeLLo without the need for additional preprocessing, as MeLLo naturally outputs parametric knowledge (Section 3.1). The prompts used in IKE and MeLLo are shown in Appendix C. The model editing methods ROME in our baselines are deployed using EasyEdit (Wang et al., 2023b). We set n to 2 in the n-gram decomposition, the adaptive constraint $\alpha$ to 0.0005 and $k$ to 10, with bias coefficients $\lambda_n$ set to 25 and $\lambda_p$ set to 1.

### 4.2 Overall Performance

We set the batch size of the edit memory as 1 for evaluating the foundational direct editing capabilities of IKE (Zheng et al., 2023) method, especially considering multi-hop questions with 1,000 instances. The batch size means the number of instances providing the edited facts for knowledge retrieval. Table 1 displays the performance on MQuAKE across various models and datasets. Overall, compared to the model-editing method ROME, the ICE method IKE demonstrates a clear advantage. The enhanced IKE by ATBias consistently shows the best performance. Furthermore, as model parameters increase (LLaMA2-13B-chat) and pretraining becomes more refined (Mistral-7B-instruct), the knowledge within LLMs becomes more stubborn to editing. ATBias can enhance ICE to effectively edit this stubborn knowledge.

We follow the setup of previous work (Zhong et al., 2023; Wang et al., 2024) to conduct experiments for MeLLo (Zhong et al., 2023) with the full batch size edit memory. As shown in Table 2, the experimental results illustrate that ATBias enhances MeLLo to varying degrees in full batch experiments. Specifically, the enhancement provided by our ATBias shows a significant advantage, with an impressive improvement of up to 32.3% compared to DeCK. This is because ATBias operates on a small number of key tokens rather than the entire sequence as in DeCK, leaving other tokens in the inference process unaffected. This greatly reduces the potential risk of introducing fundamental errors during the inference stage, making our ATBias's enhancements even more pronounced in longer and more complex editing pipelines. It further indicates that ATBias holds significant potential for real-world KE applications with higher performance and lower costs.

### 4.3 One-hop Editing

Despite the greater challenge of multihop editing, we still used the CounterFact dataset to evaluate one-hop editing for the robustness of our method. As the results shown in Table 3, the ICE method IKE achieved high accuracy in the simpler one-hop editing task, with IKE enhanced by our ATBias consistently outperforming others.

| Model | IKE | w/ DeCK | w/ ATBias |
|---|---|---|---|
| LLaMA2-7B | 98.37 | 98.65 | **99.42** |
| LLaMA2-13B | 93.76 | 94.23 | **95.35** |

Table 3: Experimental results of IKE on CounterFact for one-hop editing task.

### 4.4 Editing on Stubborn Knowledge

Stubborn knowledge in LLMs is difficult to edit because it is established with strong confidence during the pretraining process. We follow (Bi et al., 2024a) to construct the corresponding STUBBORN datasets for different models to specifically evaluate ATBias's performance on stubborn knowledge. The stubborn datasets are categorized into different

| Model | STUBBORN | ROME | IKE | IKE w/ DeCK | IKE w/ ATBIAS |
|---|---|---|---|---|---|
| LLAMA2-7B-CHAT | > 33% | 17.7 | 56.4 | 72.3 | **73.9** |
| | > 67% | 19.3 | 37.8 | 55.9 | **57.8** |
| LLAMA2-13B-CHAT | > 33% | 42.5 | 38.9 | 70.1 | **71.6** |
| | > 67% | 40.2 | 29.4 | 48.5 | **56.5** |
| MISTRAL-7B-INSTRUCT | > 33% | 19.7 | 20.7 | 26.5 | **33.2** |
| | > 67% | 18.5 | 17.9 | 22.6 | **27.9** |

Table 4: Performance of different models on their respective STUBBORN datasets. The edit memory batch size of the IKE methods is set to 1. 'STUBBORN > 33%' indicates instances from the MQUAKE-3K dataset where IKE failed to edit knowledge more than 33% of the time. 'STUBBORN > 67%' follows the same criterion.

difficulty levels based on the proportion of correct answers when using ICE methods to edit the same knowledge multiple times with different questions.

The experimental results on the STUBBORN datasets are presented in Table 4. We find that IKE's performance on STUBBORN datasets significantly declined compared to Table 1, even falling below the model-editing method ROME. This indicates that relying solely on external prompts is insufficient to change LLMs' confidence in this stubborn knowledge. The enhancement methods applied during decoding significantly improve the effectiveness of editing stubborn knowledge, with ATBIAS consistently achieving the best performance. This suggests ATBIAS enhances ICE methods' ability to effectively edit stubborn knowledge.

| Model | Method | Latency (ms/token) | Throughput (token/s) |
|---|---|---|---|
| LLAMA2-7B-CHAT | Baseline | 36.03 (×1.00) | 27.76 (×1.00) |
| | DeCK | **69.99** (×1.94) | **14.29** (×0.51) |
| | ATBIAS | **36.19** (×1.01) | **27.64** (×1.00) |
| LLAMA2-13B-CHAT | Baseline | 51.41 (×1.00) | 19.45 (×1.00) |
| | DeCK | **94.08** (×1.83) | **10.63** (×0.55) |
| | ATBIAS | **49.11** (×0.95) | **20.36** (×1.05) |

Table 5: Decoding latency (ms/tokens) and throughput (tokens/s). **Green** shows low latency and high throughput, **red** shows high latency and low throughput.

### 4.5 Latency & Throughput

Table 5 shows the decoding latency for the baseline, as well as when incorporating DeCK or ATBIAS. DeCK requires generating and comparing two sequences during decoding, resulting in approximately 2x the latency of the baseline. It is worth noting that ATBIAS increases the decoding time by only a factor of 1.01 in LLAMA2-7B-CHAT and is even more efficient in LLAMA2-13B-CHAT compared to the baseline. This efficiency is due to the probabilistic-ranking filter, which filters out most low-probability tokens and only considers highly confident tokens for prediction. It suggests that ATBIAS, with its outstanding editing performance, can also be widely applied at negligible cost.

### 4.6 Why ATBIAS Edits Efficiently?

Bi et al. (2024a) observes that the values of the logits corresponding to the parametric knowledge's tokens are very high before editing. Even though ICE significantly increases the logits of the tokens corresponding to new knowledge, there are still cases where it fails to surpass the parametric ones. To reveal the underlying reasons why ATBIAS can effectively enhance the ICE methods from a model interpretability perspective, we analyzed the probability changes of the new knowledge before and after applying ATBIAS. Specifically, we capture the first tokens of the new and parametric knowledge entities that represent them and then record their normalized logits.

The results illustrated in Figure 3 show that IKE with ATBIAS has a higher distribution within the high probability range, while IKE without ATBIAS is concentrated in the low probability range. Additionally, the probability ranking of new knowledge significantly increased after incorporating ATBIAS. Moreover, the probability distribution of the parametric knowledge exhibited an opposite trend after incorporating ATBIAS. This further explains why ATBIAS can effectively enhance ICE: it increases the probabilities of new knowledge entities and decreases the probabilities of parametric knowledge entities. As shown in the editing example in Figure 2 (*Richard Dawkins is a citizen of the United Kingdom*), the newly generated knowledge entities by ATBIAS serve as new contextual cues during inferencing to reason over multiple hops of knowledge, significantly improving editing performance.

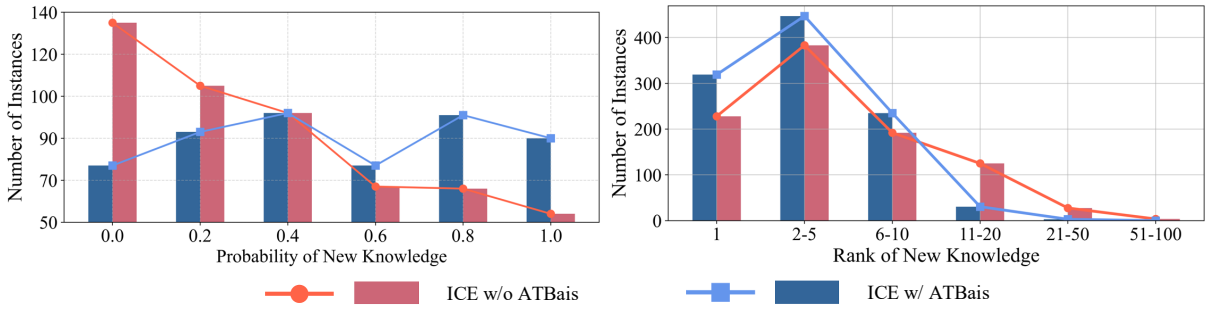



Figure 3: Probability (left) and ranking (right) statistics of new Knowledge for LLAMA2-7B-CHAT on stubborn > 33%. The probabilities are derived from normalize calculations.

### 4.7 Ablation Study

We conducted a comprehensive ablation study on the adaptive constraints, bias coefficients, and key components of ATBIAS. Table 6 presents the results for the filter in ATBIAS, demonstrating the necessity of filtering tokens based on both probability and ranking constraints. Additional ablation study results can be found in the Appendix B.

| Model | Prob | Rank | Prob & Rank |
|---|---|---|---|
| LLAMA2-7B | 90.2 | 81.5 | **93.1** |
| LLAMA2-13B | 81.9 | 72.4 | **89.7** |

Table 6: Ablation study results for the filter of our ATBIAS. Prob and Rank respectively represent probability and ranking constraints in the filter.

## 5 Related Work

**Factual Hallucinations.** Factual hallucinations have garnered widespread attention due to their significant side effects, as LLMs generate content that deviates from established world knowledge (Tonmoy et al., 2024; Huang et al., 2023a; Wang et al., 2023a). These hallucinations can arise from various sources and at different stages of the LLM life cycle (Zhang et al., 2023b). Outdated knowledge is a major factor contributing to factual hallucinations. ATBIAS enhances KE during the inference stage in LLMs to mitigate these hallucinations.

**Knowledge Editing.** KE (Yao et al., 2023) has been proposed to update information in LLMs, enabling accurate responses to current questions. In general, there are three lines of works for KE. Model editing (Zhu et al., 2020; Meng et al., 2022a,b; Huang et al., 2023b) involves adding or altering the model parameters responsible for the undesirable output. Meta-learning methods (De Cao et al., 2021; Mitchell et al., 2021) use a hypernetwork to learn the necessary adjustments for editing LLMs. In-context editing methods (ICE) (Mitchell et al., 2022; Madaan et al., 2022; Zhong et al., 2023; Zheng et al., 2023) demonstrate significant potential, enabling the editing of LLMs by prompting them with edited facts and retrieving editing demonstrations from the edit memory.

**Decoding Strategy.** Recent work modifies various decoding strategies to enhance different alignments by altering the logits of the original tokens during generation. CD (Li et al., 2023) compares powerful expert language models with weaker amateur language models to enhance fluency and coherence. DoLa (Chuang et al., 2023) contrasts mature layers with premature layers, while ICD (Zhang et al., 2023a) compares with models injected with hallucinations, aiming to enhance the factual accuracy of the model. DeCK (Bi et al., 2024a) enhances ICE by highlighting the output probability increment of new knowledge in contrast to the parametric knowledge. Unlike the aforementioned decoding methods, ATBIAS proposed in this paper only needs to adjust key tokens to enhance KE and mitigate factual hallucinations in LLMs.

## 6 Conclusion

In this work, we propose a new KE framework, ATBIAS, to enhance ICE. ATBIAS focuses on the crucial tokens that are mostly related to knowledge during the generation, biasing their logits by matching the knowledge entities. This design effectively reduces the potential risk of introducing fundamental errors in the logical coherence of the entire inference statement. Experimental results show that ATBIAS significantly improves the editing success rate of ICE and outperforms the current best decoding methods. Furthermore, the latency of ATBIAS is at most 1.01 times that of the baseline, meaning ATBIAS not only enhances ICE but can also be widely applied with negligible cost.

## Limitations

We mainly evaluate the KE methods on the LLaMA2-7B-chat, LLaMA2-13B-chat, and Mistral-7B-instruct. The efficacy of these methods on other LLMs remains less explored. Additionally, although ATBias is expected to be easily deployable on any ICE method to enhance KE performance, we currently evaluate ATBias on the representative IKE and MeLLo, lacking broader validation. We leave the evaluation on other models and ICE methods for future work.

## Ethical Considerations

Ethical considerations are of utmost importance in our research endeavors. In this paper, we conscientiously adhere to ethical principles by exclusively utilizing open-source datasets and employing models that are either open-source or widely recognized in the scientific community. Moreover, counterfactual public datasets were used in knowledge editing to measure knowledge updates. Our proposed method is designed to ensure that the model does not produce any harmful or misleading information. We are committed to upholding ethical standards throughout the research process, prioritizing transparency, and promoting the responsible use of technology for the betterment of society.

## References


Baolong Bi, Shenghua Liu, Lingrui Mei, Yiwei Wang, Pengliang Ji, and Xueqi Cheng. 2024a. Decoding by contrasting knowledge: Enhancing llms' confidence on edited facts. *Preprint*, arXiv:2405.11613.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, and Xueqi Cheng. 2024b. Is factuality decoding a free lunch for llms? evaluation on knowledge editing benchmark. *arXiv preprint arXiv:2404.00216*.

Canyu Chen and Kai Shu. 2023. Combating misinformation in the age of llms: Opportunities and challenges. *arXiv preprint arXiv:2311.05656*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

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

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *Preprint*, arXiv:2311.05232.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023b. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*.

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

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve gpt-3 after deployment. *arXiv preprint arXiv:2201.06009*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. 2013. Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 380–384.

OpenAI. 2022. large-scale generative pre-training model for conversation. *OpenAI blog*.

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

Yucheng Shi, Qiaoyu Tan, Xuansheng Wu, Shaochen Zhong, Kaixiong Zhou, and Ninghao Liu. 2024. Retrieval-enhanced knowledge editing for multi-hop question answering in language models. *arXiv preprint arXiv:2403.19631*.

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

Zezheng Song, Jiaxin Yuan, and Haizhao Yang. 2024. Fmint: Bridging human designed and data pretrained models for differential equation foundation model. *arXiv preprint arXiv:2404.14688*.

S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *Preprint*, arXiv:2401.01313.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

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, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, et al. 2023a. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *arXiv preprint arXiv:2310.07521*.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. 2023b. Easyedit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*.

Yiwei Wang, Muhao Chen, Nanyun Peng, and Kai-Wei Chang. 2024. Deepedit: Knowledge editing as decoding with constraints. *arXiv preprint arXiv:2401.10471*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*.

Yue Zhang, Leyang Cui, Wei Bi, and Shuming Shi. 2023a. Alleviating hallucinations of large language models through induced hallucinations. *arXiv preprint arXiv:2312.15710*.

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. 2023b. Siren's song in the ai ocean: A survey on hallucination in large language models. *Preprint*, arXiv:2309.01219.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*.

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

## A How Can ATBIAS Be Easily Deployed on MeLLo?

Many advanced ICE methods (Zhong et al., 2023; Wang et al., 2024; Shi et al., 2024) inherently possess parametric knowledge, so ATBIAS does not need to induce LLMs to preprocess it offline. Table 4 demonstrates how MeLLo (Zhong et al., 2023) can easily deploy ATBIAS without additional inference, directly extracting the required entities from the parametric output of subquestion responses. This means that knowledge entities can be extracted

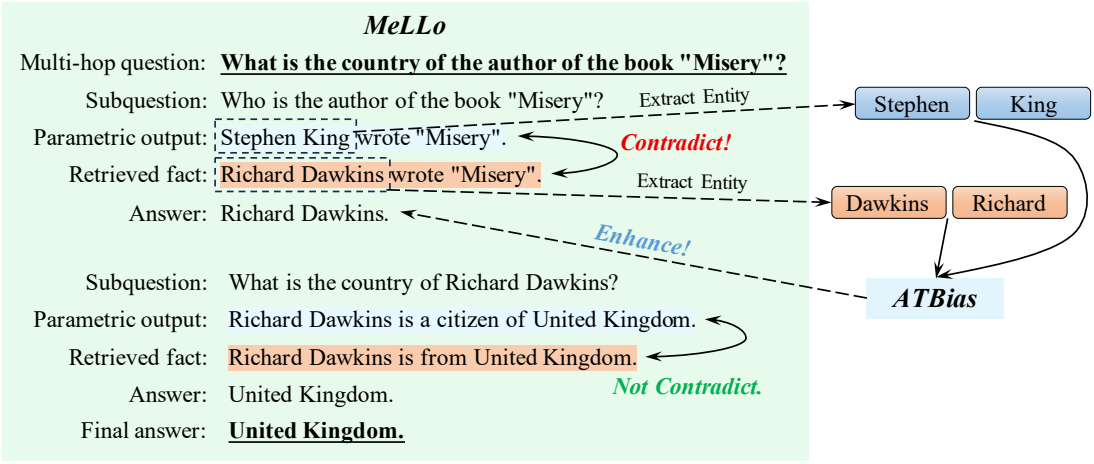



Figure 4: An illustration of ATBIAS's easy deployment on MeLLo.

online and fed into ATBIAS when using MeLLo. Thus, ATBIAS enhances ICE during the decoding stage with just a single inference step.

## B Additional Ablation Study of ATBIAS

We conduct following additional ablation study experiments using the ICE method IKE (Zheng et al., 2023) with LLAMA2-7B-CHAT and LLAMA2-13B-CHAT on the MQUAKE-3K datasets.

### B.1 N-gram Decomposition

The N-gram decomposition is a prerequisite for calculating the similarity between the knowledge entities and filtered tokens (Section 3.3). Table 5 presents the ablation study results for various values of gram n during this process. Both excessively high and low decomposition precision can diminish the matching effectiveness, with $n = 2$ yielding the best editing performance.

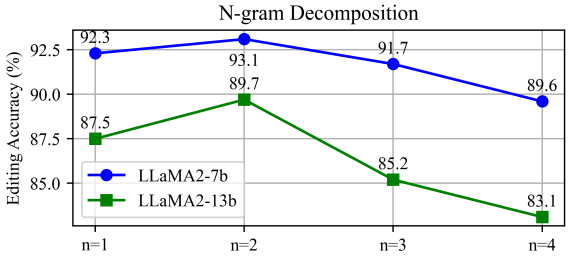



Figure 5: Ablation study results of the gram n for n-gram decomposition process.

### B.2 Probabilistic Constraint of Filter

The probabilistic constraint of ATBIAS's filter (Section 3.2) that represented in Equation 3 is subjected to an ablation study on the parameter $\alpha$. The results of this study are shown in Table 6, indicating that $\alpha = 0.0005$ yields the best editing performance. The fact that smaller $\alpha$ values yield better performance further indicates the strictness of our filtering process, effectively preventing interference from unreasonable tokens.

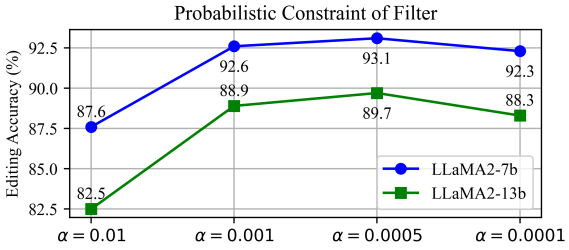



Figure 6: Ablation study results of the probabilistic constraint $\alpha$ of filter.

### B.3 Ranking Constraint of Filter

The ablation study results of ranking constraint (Equation 4) are illustrated in Table 7, showing that $k = 10$ yields the best editing performance.

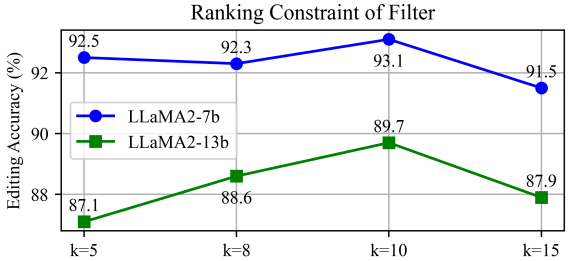



Figure 7: Ablation study results of the ranking constraint $k$ of filter.

### B.4 Bias Coefficient of Knowledge

We adjust the logits of tokens matching with the new and parametric knowledge entities (Section 8) with the bias coefficients $\lambda_n$ (Equation 10) and $\lambda_p$. The ablation study results of $\lambda_n$ and $\lambda_p$ are shown in Table 9 and 10, respectively. ATBIAS achieves the best performance when $\lambda_n = 25$.

[3 in-context demonstrations abbreviated]

Question: **What is the capital city of the country of citizenship of Ivanka Trump's spouse?**
Edit Knowledge: Jared Kushner is a citizen of Canada.
Thoughts: Ivanka Trump's spouse is Jared Kushner. Jared Kushner is a citizen of Canada. The capital city of Canada is Ottawa.
Answer: **Ottawa**

Question: **Which continent is the country where the director of "My House Husband: Ikaw Na!" was educated located in?**
Edit Knowledge: Irene Villamor was educated in New York University.
Thoughts: The director of "My House Husband: Ikaw Na!" is Jose Javier Reyes. Jose Javier Reyes was educated in New York University. De La Salle University is located in United States of America. United States of America is located in the continent if North America.
Answer: **North America**

Table 7: An illustration of the COT based IKE solving two simplified examples. The orange parts are facts retrieved by the retriever.

[2 in-context demonstrations abbreviated]

Question: **What is the capital city of the country of citizenship of Ivanka Trump's spouse?**
Subquestion: Who is Ivanka Trump's spouse?
Generated answer: Ivanka Trump's spouse is Jared Kushner.
Retrieved fact: David Cameron is married to Samantha Cameron.
Retrieved fact does not contradict to generated answer.
Intermediate answer: Jared Kushner
Subquestion: What is the country of citizenship of Jared Kushner?
Generated answer: The country of citizenship of Jared Kushner is United States.
Retrieved fact: Jared Kushner is a citizen of Canada.
Retrieved fact contradicts to generated answer.
Intermediate answer: Canada
Subquestion: What is the capital city of Canada?
Generated answer: The capital city of Canada is Ottawa.
Retrieved fact: The capital city of United States is Seattle.
Retrieved fact does not contradict to generated answer, so the intermediate answer.
Intermediate answer: Ottawa
Final answer: **Ottawa**

Table 8: A step-by-step illustration of MeLLo solving one simplified example. Blue parts are generated by the language model, and orange parts are facts retrieved by the retriever.

| **Model** | $\lambda_n = 20$ | $\lambda_n = 25$ | $\lambda_n = 30$ |
|---|---|---|---|
| LLaMA2-7B | 90.5 | **93.1** | 92.7 |
| LLaMA2-13B | 86.6 | **89.7** | 88.9 |

Table 9: Ablation study results of the bias coefficient of new knowledge $\lambda_n$.

| **Model** | $\lambda_p = 0$ | $\lambda_p = 1$ | $\lambda_p = 2$ |
|---|---|---|---|
| LLaMA2-7B | 85.9 | **93.1** | 88.6 |
| LLaMA2-13B | 70.2 | **89.7** | 83.2 |

Table 10: Ablation study results of the bias coefficient of parametric knowledge $\lambda_p$.

ATBIAS achieves the best performance when $\lambda_p = 1$. An $\lambda_p$ value of 0 means that the logits of tokens matching with parametric knowledge entities are not reduced, and the results indicate that this leads to a decline in performance. Optimal performance is achieved with smaller values of $\lambda_p$ because excessively large $\lambda_p$ values may cause the logits of tokens incorrectly matching old knowledge entities to decrease too much, adversely affecting editing performance.

## C Prompts of ICE for Experiments

The prompt we used in IKE (Zheng et al., 2023) is shown in 7, and the prompt we used in MeLLo is shown in 8. Based on the provided contextual demonstrations, LLMs can be guided to perform the corresponding ICE methods. ATBIAS can enhance these ICE methods without modifying any prompts.