EvEdit: Event-based Knowledge Editing for Deterministic Knowledge Propagation

Jiateng Liu[1,*], Pengfei Yu[1,2,*], Yuji Zhang[1], Sha Li[1,3], Zixuan Zhang[1,3], Kevin Small[3], Ruhi Sarikaya[3], Heng Ji[1,3]
[1]University of Illinois Urbana-Champaign [2]Boson AI
[3]Amazon
{jiateng5, pengfei4, hengji}@illinois.edu

Abstract

The dynamic nature of real-world information necessitates knowledge editing (KE) in large language models (LLMs). This edited knowledge should propagate and facilitate the deduction of new information based on existing model knowledge. We define the existing related knowledge in a LLM serving as the origination of knowledge propagation as "deduction anchors". However, most of current KE approaches only operate on (subject, relation, object) triples. Both theoretically and empirically, we observe that this simplified setting often leads to uncertainty when determining the deduction anchors, causing low confidence in their responses. To mitigate this issue, we propose a novel task of *event-based knowledge editing* that pairs facts with event descriptions. This task manifests both as a closer simulation of real-world editing scenarios and a more logically sound setting, implicitly defining the deduction anchor and enabling LLMs to propagate knowledge confidently. We curate a new benchmark dataset EvEdit derived from the CounterFact dataset and validate its superiority in improving model confidence. Moreover, as we observe that the event-based setting is notably challenging for existing approaches, we propose a novel approach *Self-Edit* that showcases stronger performance, achieving 55.6% consistency improvement while maintaining the naturalness of generation.[1]

1 Introduction

The dynamics of the physical world underscore the importance of knowledge editing (KE) for large language models (Yao et al., 2023; Wang et al., 2023d; Zhang et al., 2024b). This line of research aims at updating models' beliefs and shaping models' behaviors based on editing knowledge for improved accuracy and usability. Ideally, the edited knowledge should be able to propagate through other related facts and deduct new knowledge. For instance, by updating the model with "Messi joined team Inter Miami", the edited model should acknowledge that "Messi began playing in Major League Soccer (MLS)", as "Inter Miami competes in MLS". This knowledge propagation is referred to as the *ripple effect* in Cohen et al. (2023). In this paper, we define the prior knowledge "Inter Miami competes in MLS" as the *deduction anchor* due to its role in the knowledge propagation process.

Current KE approaches (Meng et al., 2022a,b; Hartvigsen et al., 2023; Li et al., 2023) merely focus on edits of (subject, relation, object) triples. We observe that this simplified setting frequently results in undetermined deduction anchors during knowledge propagation. As illustrated in Figure 1, after editing the model to state "Messi is a Dutch citizen" and querying "Where was Messi born?", at least two logical deduction anchors may emerge, reducing the model's certainty in generating responses. On one side, the model might select the prior knowledge that "Messi was born in Argentina" as the anchor and predict "Argentina", possibly implying that the edit "Messi is a Dutch citizen" reflects a change in citizenship. Alternatively, the model could use "a Dutch citizen should be born in the Netherlands" as the anchor and consequently deduce "Netherlands".

To further analyze this phenomenon, we present a theoretical framework of knowledge editing based on the *formal logic* (Smith, 2003), representing knowledge as formal language propositions in § 2. Using our theoretical framework, we find that existing work did not explicitly define the deduction anchor while two implicitly available assumptions are actually flawed: neither the *no-anchor assumption* (an empty anchor set) nor the *max-anchor assumption* (an anchor set comprising all knowledge not conflicting with the edit) provides a logically sound knowledge editing setting. Consequently, the existing setting theoretically increases

*These authors contribute to this work equally.
[1]We will release the benchmark and code.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 4907–4926
November 12-16, 2024 ©2024 Association for Computational Linguistics

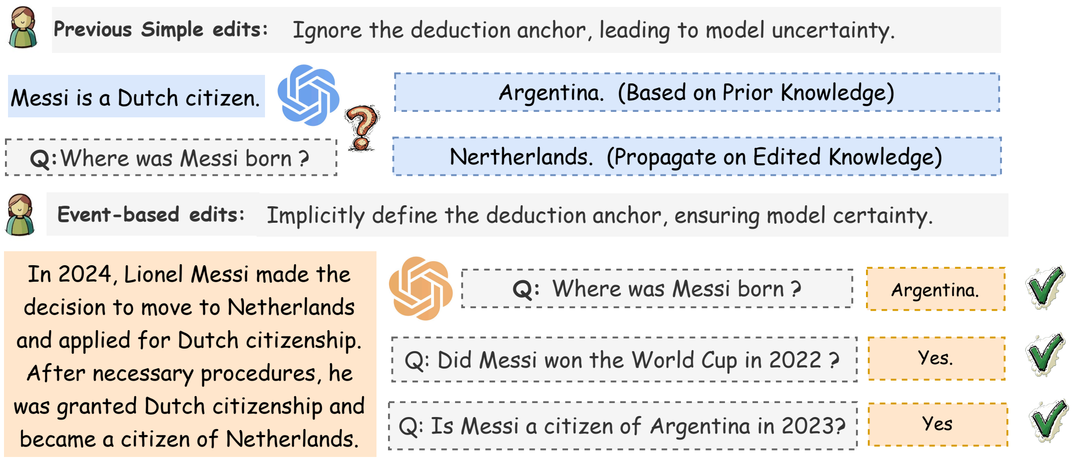


Figure 1: We observe fallacies of existing knowledge editing when the Deduction Anchor is not defined. The model edited with 'Messi is a Dutch citizen' may mistakenly propagate the edited knowledge that Messi was born in the Netherlands, which conflicts with its prior knowledge that Messi was born in Argentina.

the uncertainty, which is further verified empirically across popular large language models.

We observe that the fallacy mentioned above, despite its prevalence in knowledge editing for LLMs, does not manifest in the real world. As shown in Figure 1, if Messi were to become a Dutch citizen, a corresponding real-world event would need to occur. This event could be either Messi applying for residency in the Netherlands or a revelation that he was actually born and raised in the Netherlands, unbeknownst to people until now. In our example, if the event behind is about Messi applying for and obtaining Dutch citizenship, we can still affirm that Messi's birthplace is Argentina. With this background knowledge, the uncertainty is naturally resolved. Through extensive experiments, we also verified that event descriptions are indeed helping models improve their confidence when performing editing. Therefore, to overcome the limitations of the current setting, we introduce **event-based knowledge editing**, which not only provides a more robust framework by presenting clearer deduction anchors and editing boundaries but also offers a more practical setting, as real-world changes are often driven by events (Chen et al., 2021a,b). We derive a new benchmark EVEDIT from a triple-based knowledge editing benchmark COUNTERFACT (Meng et al., 2022a) by augmenting facts with events using GPT-3.5-turbo plus human verification. We evaluate the post-edit model's ability with both text completion and QA tasks.

To perform knowledge editing under this new setting, we decompose the event descriptions into a series of triples to accommodate current editing methods like Rome (Meng et al., 2022a), MEMIT (Meng et al., 2022b), PMET (Li et al., 2023) and Grace (Hartvigsen et al., 2023). We further propose a novel solution Self-Edit inspired by Yu and Ji (2023) which can effectively utilize the eventual context to decide editing boundaries during updating. Our evaluations show that while adapting previous editing approaches provides suboptimal results, our approach exhibits over 56.6% increase in factual consistency while keeping the naturalness of generations by edited models. Our approach neither requires a linearly growing external memory which previous works (Zhong et al., 2023) used to trade for high performance.

Overall, our contributions are: (1) We identify a critical deficiency of the current KE setting, by providing a careful theoretical analysis for KE and conducting extensive experiments, we attribute the problem to the improper assignment of deduction anchor. (2) We propose event-based knowledge editing and a new benchmark EVEDIT, addressing the problem of current KE of missing deduction anchors while aligning well with real-world scenarios. We then empirically validate the superiority of our setting. (3) We propose a novel Self-edit approach for doing KE under the new setting, significantly outperforming existing methods on generation consistency and naturalness.

2 Fallacies of Knowledge Editing

In this section, we formulate and analyze the task of knowledge editing from both theoretical and empir-

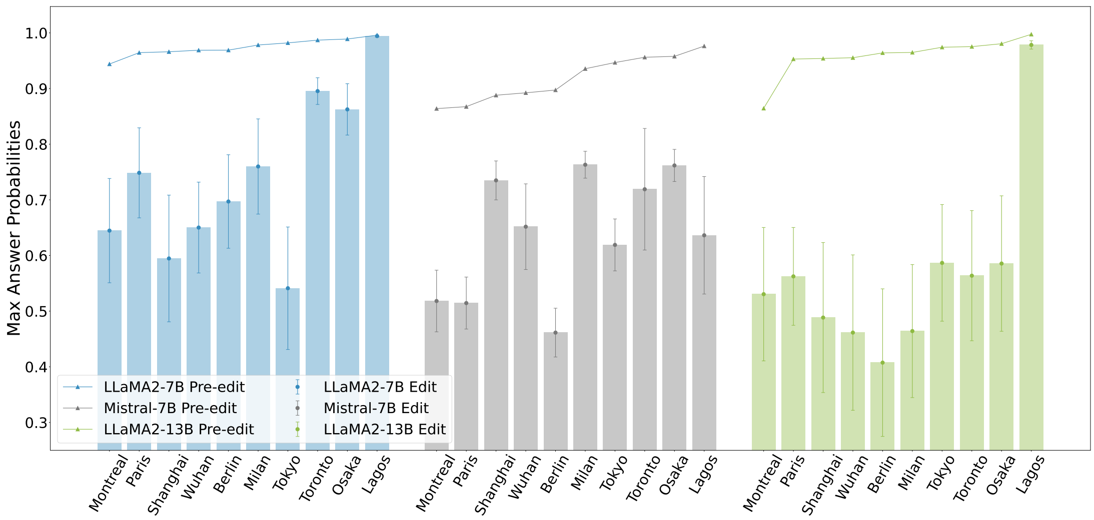


Figure 2: Counterfactual edits reduces model's certainty on relevant knowledge. We measure certainty as the maximum answer probability to the query of "A is located in the country of _" where A is one of the cities labeled in the X-axis. We compute the range of "Edit" probabilities by prepending various counterfactual edits as context to the query. "Pre-edit" probabilities are model predictions without any context. See main text for more details.

ical perspectives. In § 2.1, we present a theoretical formulation for knowledge editing. In § 2.2, we theoretically analyze the fallacies of the existing work and empirically validate its failure.

2.1 Formulation of Knowledge Editing

We present a theoretical framework for knowledge editing based on the *formal logic* (Smith, 2003) where we consider *knowledge* as *propositions*.[2] For a knowledge system, the purpose of knowledge editing is to alter its set of knowledge. Therefore, we first formally define the knowledge within a system and the knowledge edit.

Definition 2.1 (Knowledge of Models)**.** The *knowledge* of a model is a set of propositions that are considered true in the model.

To align the theoretical framework with language model (LM) editing, we introduce the knowledge of LMs. Let k denote a proposition, and let Θ represent an LM. We assess whether Θ "possesses" knowledge of k by calculating $P(y_k|x_k, \Theta)$, where (x_k, y_k) represents a pair of input-output tokens to verify the knowledge. For example, we may use x_k = 'Messi was born in' and y_k = 'Argentina' to examine the knowledge of the birthplace of Messi. We opt for $P(y_k|x_k, \Theta)$ over $P(k|\Theta)$ because the probability assigned by a language model to a proposition does not inherently correlate with its logical validity (Yu and Ji, 2023)

[2]Propositions are arguments that can be either true or false

Definition 2.2 (Knowledge of LMs)**.** For a language model Θ, the universe of all conceivable knowledge $\mathcal{U}$, and a threshold ε within the range $[0, 0.5)$, the set of knowledge recognized by Θ is

$$\mathcal{K}_{\Theta,\varepsilon} = \{k \in \mathcal{U} | P(y_k|x_k, \Theta) \geq 1 - \epsilon\}. \quad (1)$$

There could be multiple candidates $\{(x_k^i, y_k^i)\}$ verifying the same knowledge k. We can replace $P(y_k|x_k, \Theta)$ with a random sample, mean or maximum of all candidates' probabilities in Equation (1) with no influence on the rest of the formulation. Therefore, we simply use $P(y_k|x_k, \Theta)$ for brevity.

In this work, we are specifically concerned with the logical deduction during editing such as:

$\boldsymbol{P}$: Tom was born in the city of New York
$\boldsymbol{Q}$: The country where New York is located is U.S.
$\downarrow$
$\boldsymbol{X}$: Tom was born in the country of U.S.

For a knowledge set $\mathcal{K}$, its *deductive closure* $\mathcal{B}(\mathcal{K})$ is the set of all propositions logically entailed by $\mathcal{K}$. $\mathcal{K}$ is *deductively closed*, or simply *closed*, if and only if $\mathcal{B}(\mathcal{K}) = \mathcal{K}$. Determining the deductive closure presents a significant challenge due to the difficulty in formulating deduction rules (Smith, 2003). However, given the advanced in-context reasoning capabilities demonstrated by large language models, we establish the deductive closure based on such in-context deduction.

Definition 2.3 (In-context Deductive Closure)**.** For any given set of knowledge $\mathcal{K}$, its *In-context Deductive Closure* as provided by a language model

Φ is the set of knowledge that can be deduced,

$$\mathcal{B}_{\Phi,\epsilon}(\mathcal{K}) = \{u \in \mathcal{U} | P(y_u | x_u, \mathcal{K}, \Phi) \geq 1 - \epsilon\}. \tag{2}$$

Let $\mathcal{K}$ be the knowledge set of the pre-edit model, and $\mathcal{E}$ be the set of editing knowledge. We define two novel concepts for the soundness of editing: *deduction anchor* and *editing boundary*.

Definition 2.4 (Deduction Anchor of Editing)**.** The *deduction anchor* of an edit is a subset of the current knowledge assumed true throughout editing.

We denote the deduction anchor by $\mathcal{K}^{\mathcal{E}}$, which serves as the base for the knowledge generalization of editing. We now define the editing boundary.

Definition 2.5 (Editing Boundary)**.** The editing boundary is the closed set $\mathcal{B}\left(\mathcal{K}^{\mathcal{E}} \cup \mathcal{E}\right)$ of logically relevant knowledge to the edit $\mathcal{E}$.

We thereby define *knowledge editing*.

Definition 2.6 (Knowledge Editing)**.** Given the knowledge set $\mathcal{K}$, the edit $\mathcal{E}$ and the deduction anchor $\mathcal{K}^{\mathcal{E}}$, *knowledge editing* is the process of computing edited knowledge set $\mathcal{K}'$:

$$\begin{aligned} \mathcal{K}^D &= \left\{p \in \mathcal{K} | \neg p \in \mathcal{B}\left(\mathcal{K}^{\mathcal{E}} \cup \mathcal{E}\right)\right\} \\ \mathcal{K}' &= \mathcal{B}\left(\mathcal{K} \backslash \mathcal{K}^D \cup \mathcal{E}\right) \end{aligned}, \tag{3}$$

where $\mathcal{K}^{\mathcal{E}}$ satisfies that

$$\forall k \in \mathcal{B}\left(\mathcal{K} \backslash \mathcal{K}^D\right), \neg k \notin \mathcal{B}\left(\mathcal{K}^{\mathcal{E}} \cup \mathcal{E}\right). \tag{4}$$

Here Equation (4) ensures the consistency of $\mathcal{K}'$. $\mathcal{K}^D$ is the set of knowledge conflicting with the deducted knowledge from $\mathcal{K}^{\mathcal{E}} \cup \mathcal{E}$, which needs to be erased from the model being edited.

We also define knowledge editing of LMs. It's important to note that the model used to determine the deductive closure in Equation (2) serves only in defining the task and not in the editing process. Thus, it may differ from the model undergoing edit: we may employ stronger models to define anchors when evaluating editing of weaker models.

Definition 2.7 (Knowledge Editing of LMs)**.** Following the notations in Equation (3), to edit a language model Θ based on the in-context deductive closure provided by Φ involves identifying a modified model Θ' such that

$$\begin{aligned} \mathcal{K}^D &= \left\{p \in \mathcal{K}_{\Theta,\varepsilon_\Theta} | \neg p \in \mathcal{B}_{\Phi,\varepsilon_\Phi}\left(\mathcal{K}^{\mathcal{E}} \cup \mathcal{E}\right)\right\} \\ \mathcal{K}' &= \mathcal{B}_{\Phi,\varepsilon_\Phi}\left(\mathcal{K}_{\Theta,\varepsilon_\Theta} \backslash \mathcal{K}^D \cup \mathcal{E}\right) \end{aligned}. \tag{5}$$

where $\mathcal{K}_{\Theta,\varepsilon_\Theta}$ and $\mathcal{B}_{\Phi,\varepsilon_\Phi}$ are defined in Definition 2.2 and Definition 2.3, respectively.

2.2 Fallacies of Existing Knowledge Editing

Existing work predominantly ignores the significance of the deduction anchor and resulting editing boundary without explicit characterizations of them. They mostly focus on local edits assuming $\mathcal{K}^{\mathcal{E}} = \varnothing$, which limits the editing boundary $\mathcal{B}(\mathcal{E})$ to only contain paraphrases of $\mathcal{E}$, as the *edit scope* proposed by Mitchell et al. (2022). Additionally, Cohen et al. (2023) implicitly assumes all knowledge not directly conflicting with $\mathcal{E}$ as the deduction anchor. However, we present the following theorems, emphasizing the importance of choosing an appropriate set of $\mathcal{K}^{\mathcal{E}}$ and summarizing fallacies under their flawed assumptions.

Theorem 1 (Knowledge Explosion)**.** If Equation (4) is not satisfied, the edited knowledge set $\mathcal{K}' = \mathcal{U}$ where $\mathcal{U}$ is the universe of all knowledge, meaning any proposition is logically true.

Theorem 2 (No-Anchor Fallacy)**.** For a counterfactual and non-local edit $\mathcal{E}$, there exists $\mathcal{K}^{\mathcal{E}} \in 2^{\mathcal{K}}$ satisfying Equation (4), while $\varnothing$ does not.

Theorem 3 (Max-Anchor Fallacy)**.** For a counterfactual and non-local edit $\mathcal{E}$, the max-anchor $\{p \in \mathcal{K} | \neg p \notin \mathcal{B}(\mathcal{E})\}$ does not satisfy Equation (4).

Here a *counterfactual* and *non-local* edit is one that contradicts with some but not all of the pre-edit knowledge. The rigorous definitions are presented with proofs of the above theorems in Appendix A.

Moreover, the knowledge explosion leads to the shrinkage of the knowledge set of language models following Equation (1). The reason is that for two conflicting knowledge elements p, q where $x_p = x_q, y_p \neq y_q$, a language model cannot assign $P(y_p|x_p) \geq 1 - \varepsilon$ and $P(y_q|x_q) \geq 1 - \varepsilon$ at the same time. Consequently, we hypothesize that both probabilities will go under the threshold of $1 - \varepsilon$, causing uncertainty within models.

Empirical Verification for certainty drop We further verify the hypothesis empirically with a set of paired edits and relevant knowledge queries as follows:

Edit e: City *A* is located near to City *B*.
Query q: City *A* is located in the country of _

where *A* and *B* are two cities in different countries. For each q, we compare the pre-edit certainty $\max_y P(y|q, \Theta)$ with the edited certainty $\max_y P(y|q, e, \Theta)$ for various e with different choices of *B* in Figure 2, which demonstrates the

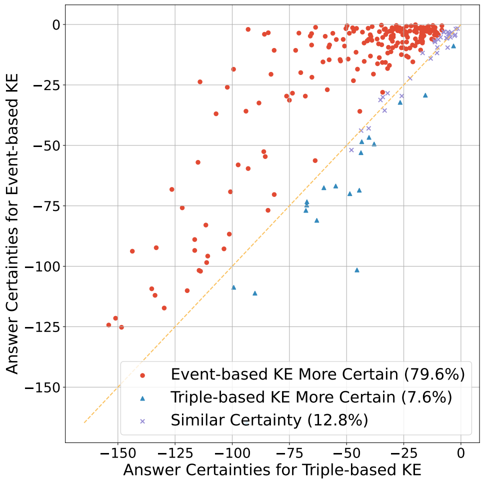


Figure 3: We evaluate LLMs' answer certainty with its average log probability over the generated tokens. Each point in the figure represents one case where we apply triple-based and event-based editing and compute the answer certainty of the post-edit LLM. We find that Triple-based editing confuses the model while event-based editing mitigates this issue 79.6% of the time.

predicted decrease in certainty. Moreover, the magnitude of the decrease appears to be larger for models with stronger reasoning abilities.

3 Event Descriptions Improves Certainty

Following our prior analysis, edits without sufficient context to infer a proper deduction anchor $\mathcal{K}^{\mathcal{E}}$ cause conflicts between the editing boundary $\mathcal{B}\left(\mathcal{K}^{\mathcal{E}} \cup \mathcal{E}\right)$ and the remaining model knowledge $\mathcal{K} \backslash \mathcal{K}^{D}$, which ultimately lead to uncertainty in edited models. Rather than simply augmenting the existing benchmarks with deduction anchors for edits, we propose a more practical setting of augmenting edits with eventual context since knowledge updates are more often driven by events in real-world scenarios (Chen et al., 2021b,a) rather than provided deduction anchors. For example, we would possibly find an event about 'Messi applied for Dutch citizenship', which hints that the fact that he was born in Argentina is not changed. On the contrary, a triple of "(Messi, citizen of, Netherlands)" alone would cause confusion.

To verify that event-based editing reduces uncertainty compared with single factual edits, we quantify uncertainty based on Equation (1) for both type of edits. However, since it is computational costly to compute $\max_y P(y|x, \Theta)$ for longer output sequences (answers or text completions), we instead use $\mathbb{E}_{y \sim P(y|x,\Theta)} \log P(y|x, \Theta)$ to measure the certainty.[3] Each edit instance in E^2dit contains the original fact, the event description, and the question-answer pairs related to the fact. We compare the certainty of a frozen pretrained LM generating answers to questions when given the original fact versus the event description. We plot our results on LLaMA2-7B-Chat in Figure 3 and leave results on Mistral-7B, and LLaMA2-13B-Chat in Appendix B. Each edit instance corresponds to a point in the scatter plot. We use red to highlight instances where event-based context enhances generation certainty, and blue to indicate the opposite case. Results show that event-based knowledge editing significantly reduces uncertainty.

4 Method: Event-Based Editing

4.1 The EvEdit Benchmark

We compile our event-based knowledge editing benchmark EvEdit from the CounterFactual dataset (Meng et al., 2022a), where each instance is a single fact to update. The procedure as described below can also be applied to other knowledge-editing datasets. Data statistics and examples of are detailed in Appendix C and the prompts for data creation are in Appendix G.

Data Collection We begin with using GPT-3.5-turbo (referred as GPT later) to filter out edits that are impossible to take place as future events, concrete examples are given in Appendix C. We then prompt GPT with in-context examples to generate an event description for each remaining edit. This step is essentially using GPT to provide simulated background event knowledge and implicitly define deduction anchors.

Evaluation Task To systematically evaluate the abilities of edited models, we include both the question-answering task and the text-completion task. For each edit, we generate five related question-answer (QA) pairs using GPT. We also require one question to be undecidable given the event description to better delineate the editing boundary by considering GPT as Φ in Definition 2.3, for which we provide the ground truth answer as "I don't know" (Zhang et al., 2023). We split the evaluation set into the "Known" set and the "Unknown" set accordingly. These QAs are subsequently transformed into text completion tasks.

[3]We sample 5 answers and average the log-likelihood.

Human Verification We did a human evaluation for the quality of GPT-generated data. The percentage of valid data samples is 96.4%, demonstrating the high quality of our generated events.

4.2 Approach: Self-Edit Framework

Inspired by Yu and Ji (2023), we design a Self-edit approach for event-based editing. Given the event-based edits, we use the pre-edit language model to create an augmented dataset to fine-tune the model. As on the right side of Figure 4, for each edit $\mathcal{E}$,

1. Conduct self-prompting of the language model to generate a related question Q.
2. Generate the answer A by prompting the LM with the question Q and the edit $\mathcal{E}$. We ask the model to generate "*I don't know*" (Zhang et al., 2024a) for unanswerable questions.
3. Create a training instance of the format $(Q \rightarrow \mathcal{E}, A)$. The model is fine-tuned to recite the edit before answering the question.

We give examples in Appendix D. For evaluation, self-generated edits before answers are removed.

5 Experiments

5.1 Experimental Settings

We edit and evaluate LLaMA2-7B-chat model on EvEdit, with the number of edits (N) varied to match the limitations of different baselines as specified in Section 5.2. The performance is assessed separately on "Known" and "Unknown" data subsets. We provide further details in Appendix E.

We adopt the factual consistency and the naturalness metrics from UniEval (Zhong et al., 2022) for evaluation. The consistency measures the effectiveness of the edits. The naturalness shows how well the model's generation ability is preserved.

5.2 Baselines Methods in Comparison

We consider three categories of baselines:

Factual-Association We adapt existing factual-association editing methods to event-based editing by decomposing each event into several fact triples with GPT-3.5-turbo, as depicted on the left side of Figure 4. We consider **ROME** (Meng et al., 2022a), **MEMIT** (Meng et al., 2022b), **PMET** (Li et al., 2023) and **GRACE** (Hartvigsen et al., 2023) in this category. These methods, however, do not scale well in terms of efficiency and effectiveness, thus we limit our evaluation to $N = 1, 10$.

Fine-tuning For this category, we fine-tune models on $N = 100$ edits and assess their performance on $N = 1, 10, 100$ in the evaluation sets. We consider the **Direct Fine-tuning** (on event descriptions) and our proposed **Self-edit** in this category. Compared to factual-association methods, fine-tuning methods support the editing of a large number of facts simultaneously.

In-context Learning Additionally, we assess an in-context performance (ICL), which involves prepending event descriptions to evaluation prompts without changing model parameters. This serves as an bound based on the model's deductive capabilities[4] since it is equivalent to setting $\Phi = \Theta$ in Definition 2.7. However, simple ICL which concatenates all the edited documents has its scalability limited by the model's context window size, thus we only evaluate it for $N = 1, 10, 59$, where 59 is the maximum number of event descriptions we can accommodate into LLaMA2. Retrieve augmented generation approaches (RAG) (Zhong et al., 2023) for knowledge editing serve as an extension for simple ICL approaches. They allow larger numbers of edits at the cost of adding a linearly growing external memory.

5.3 Main Results

We present results for both text completion and QA tasks, across various numbers of edits N and data splits (Known and Unknown) in Table 1. More qualitative results can be found in Appendix F.

Factual Association Fails EvEdit Factual-association methods display limited improvements in factual consistency while significantly harming the naturalness of generations. A typical case is that tokens from the event description are generated repeatedly, as shown in Appendix F. Among this family of methods, GRACE (Hartvigsen et al., 2023), which employs a code book as an external repository for potential hidden states, performs best in consistency. However, GRACE is sensitive to the choice of hyperparameters, as shown by the difference in performance for different ϵ values.

Self-edit Excels at EvEdit In general, fine-tuning approaches support a large number of edits with little loss in naturalness. Compared to direct

[4]This is not a theoretical upper bound of all models' or human's logical deductions abilities, but instead an empirical upper bound only for the pre-edit model.

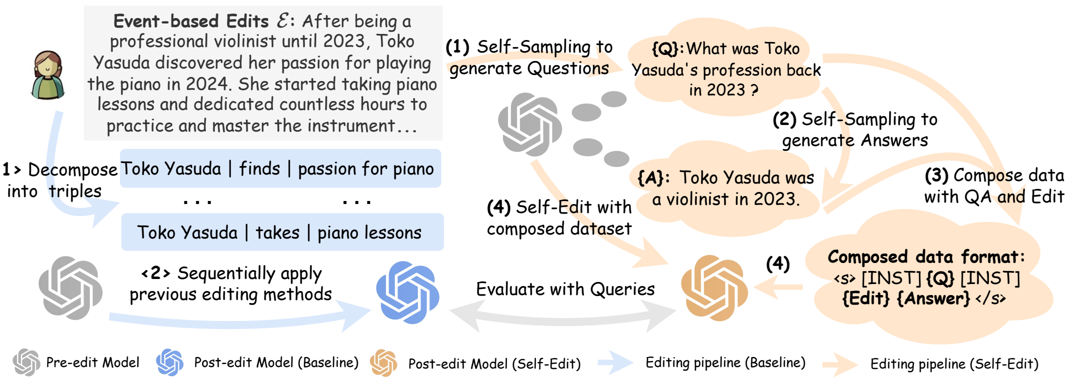


Figure 4: Different approaches to event-based knowledge editing. Left: To apply factual-association-based editing methods, we decompose event-based description into triples. Right: Our proposed Self-Edit: We first use the pre-edit LM to generate relevant QA pairs to edits. Then we fine-tune models on instances of (Q → Edit, A).

KE methods	Evaluation Metric	Text Completion N=1 Related	Text Completion N=1 Unknown	Text Completion N=10 Related	Text Completion N=10 Unknown	Question Answering N=1 Related	Question Answering N=1 Unknown	Question Answering N=10 Related	Question Answering N=10 Unknown
Base Model	*Consistency*	0.324	0.347	0.318	0.355	0.347	0.372	0.349	0.378
	Naturalness	0.894	0.869	0.898	0.875	0.833	0.821	0.845	0.866
ROME	*Consistency*	0.331	0.262	0.310	0.258	0.344	0.270	0.336	0.243
	Naturalness	0.671	0.479	0.610	0.454	0.655	0.440	0.574	0.451
MEMIT	*Consistency*	0.334	0.277	0.329	0.271	0.342	0.281	0.340	0.279
	Naturalness	0.629	0.466	0.588	0.430	0.630	0.464	0.546	0.421
PMET	*Consistency*	0.346	0.319	0.332	0.317	0.350	0.316	0.354	0.322
	Naturalness	0.840	0.812	0.880	0.862	0.814	0.790	0.822	0.793
GRACE$_{\epsilon=25}$	*Consistency*	0.436	0.320	0.442	0.304	0.441	0.317	0.443	0.340
	Naturalness	0.702	0.672	0.691	0.643	0.690	0.668	0.673	0.659
GRACE$_{\epsilon=50}$	*Consistency*	0.337	0.298	0.335	0.256	0.345	0.308	0.344	0.313
	Naturalness	0.806	0.791	0.760	0.770	0.797	0.764	0.758	0.723
ICL	*Consistency*	**0.726**	0.351	**0.626**	0.331	**0.739**	**0.405**	**0.662**	0.350
	Naturalness	**0.903**	**0.887**	**0.913**	**0.896**	**0.898**	0.846	**0.910**	**0.902**
Ours	*Consistency*	0.512	**0.401**	0.507	**0.402**	0.523	0.403	0.519	**0.388**
	Naturalness	0.804	0.867	0.816	0.877	0.816	**0.872**	0.817	0.864

KE methods	Evaluation Metric	Text Completion N=59 Related	Text Completion N=59 Unknown	Text Completion N=100 Related	Text Completion N=100 Unknown	Question Answering N=59 Related	Question Answering N=59 Unknown	Question Answering N=100 Related	Question Answering N=100 Unknown
Base Model	*Consistency*	0.304	0.383	0.320	0.358	0.345	0.386	0.356	0.381
	Naturalness	**0.906**	0.872	0.897	0.883	0.843	0.812	0.843	0.814
Finetuning	*Consistency*	0.351	0.325	0.340	0.292	0.347	0.289	0.322	0.297
	Naturalness	0.883	**0.918**	0.876	**0.901**	**0.906**	0.893	**0.904**	**0.898**
ICL	*Consistency*	0.426	0.308	-	-	0.495	0.329	-	-
	Naturalness	0.690	0.781	-	-	0.901	0.813	-	-
RAG	*Consistency*	**0.722**	0.352	**0.719**	0.347	**0.737**	**0.403**	**0.736**	**0.403**
	Naturalness	0.898	0.886	**0.899**	0.884	0.892	0.845	0.889	0.831
Ours	*Consistency*	0.502	**0.413**	0.501	**0.396**	0.523	0.391	0.517	0.385
	Naturalness	0.801	0.885	0.812	0.875	0.799	**0.897**	0.828	0.896

Table 1: Factual consistency and Naturalness of edited models. N is the number of edits at a time. We bold the best results and underline the second best for each metric.

fine-tuning, Self-Edit yields a substantial improvement on consistency, showing that the edit is effective. Moreover, our method displays clearer editing boundaries by improved scores on the Unknown subset. Since our method is fine-tuned with explicit editing boundaries by giving "I don't know" for undecidable questions, we can directly compute precision, accuracy, and F1-score for the "Unknown" subset in Table 2. Results demonstrate that although our approach demonstrates improved performance over baselines on this subset, there is still a significant gap toward a satisfying characterization of editing boundaries in edited models. We suggest that adding extra instruction-tuning data will be necessary to improve the performance further.

Text Completion			Question Answering		
Recall	Precision	F1-Score	Recall	Precision	F1-Score
0.260	0.279	0.269	0.320	0.296	0.308

Table 2: Precision, recall, and F1 of unknown questions

Space Performance Trade-off for In-Context Learning Approaches ICL approaches show superior performance for event-based editing. Simple ICL performance drops as N increases, being inferior to our method at $N = 59$. This approach can neither scale to larger N due to the limited context length of LLMs. RAG-based knowledge editing (Zhong et al., 2023) achieves good general editing performance, although it requires a linearly growing external memory. Also, the performance is inferior to our Self-Edit in terms of unknown questions, as the models are not tuned to be aware of their knowledge boundaries. We believe that both better RAG-based methods and better Self-Edit-based methods will be the focus of future research for event-based knowledge editing.

6 Related Work

6.1 Knowledge Editing

Approaches Editing an LLM's intrinsic knowledge directly changes the model's parameters. Major approaches include (1) Fine-tuning-based methods like directly fine-tuning with language modeling loss, LoRA (Hu et al., 2021) and Melo (Yu et al., 2023) (2) Meta-learning-based approaches like KE (Cao et al., 2021), MEND (Mitchell et al., 2021), and MALMEN (Tan et al., 2023) (3) Locate-and-edit method like ROME (Meng et al., 2022a), MEMIT (Meng et al., 2022b), and Pmet (Li et al., 2023). (4) Merging external knowledge representations like (Dong et al., 2022; Murty et al., 2022; Huang et al., 2023; Hernandez et al., 2023; Hartvigsen et al., 2023). However, most approaches work on the over-simplified setting and are limited by the fallacies we pointed out.

Benchmarks The most widely used dataset for knowledge editing is COUNTERFACT (Meng et al., 2022a). Other commonly used knowledge editing datasets include ZsRE (Levy et al., 2017; Yao et al., 2023), WikiBio (Hartvigsen et al., 2023), WikiData (Cohen et al., 2023), and ConvSent (Mitchell et al., 2022). More datasets used for knowledge editing can be found in Wang et al. (2023d) and a new benchmark KnowEdit (Zhang et al., 2024b). Despite many datasets, none provide event-level descriptions for knowledge editing. According to our analysis, this will ultimately lead to uncertainty and eventually hinder the edited model's performance.

6.2 Retrieval Augmentation and Tool Learning

Language models can resort to external knowledge to enhance themselves (Gao et al., 2024). The retrieval and integration process can be done in the pretraining stage (Guu et al., 2020; Borgeaud et al., 2022; Wang et al., 2023a), fine-tuning stage (Asai et al., 2023; Kang et al., 2023), and inference stage (Khandelwal et al., 2019; Sun et al., 2022) of the model. Going Further, LLM can connect to various functional ends (Yang et al., 2024), use tools (Schick et al., 2023), create tools (Yuan et al., 2024), engage with different modalities (Surís et al., 2023), involve multi-turn interactions (Wang et al., 2024b) and serve as powerful agents (Wang et al., 2023b, 2024a). However, these approaches generally need external storage and cannot intrinsically improve the language model.

7 Conclusion and Future Work

This paper establishes a theoretical framework for knowledge editing, identifying a pivotal challenge within existing methodologies: the oversight of the *deduction anchor* that leads to uncertainty within edited language models. To overcome this limitation, we introduced event-based knowledge editing. This approach enhances the traditional editing framework by incorporating event descriptions, which not only naturally mirror real-world editing scenarios but also implicitly define the deduction anchor, thereby addressing the issue of indeterminate editing boundaries. To tackle the complexities of event-based knowledge editing, we introduce an innovative *Self-Edit* method. With our new benchmark EVEDIT, we demonstrate that this new setting is challenging for existing approaches while our novel approach achieves a better performance. We advocate for further research endeavors towards this more practical, event-based knowledge editing setting.

8 Limitation

We reflect on the limitations of our paper below:

1. While this research introduces innovative strategies for addressing uncertain editing boundaries, alternative approaches exist that merit consideration. One such method involves manually curating a set of knowledge to serve as deduction anchors. This approach, though potentially effective, was not explored in our current framework.

2. The precision of event descriptions plays a crucial role in mitigating uncertainties. However, in instances where these descriptions lack sufficient detail, ambiguities may still arise, especially when addressing complex or intricately designed questions. This limitation underscores the need for highly detailed event narratives to enhance the clarity and decisiveness of knowledge edits.

3. Our evaluation was constrained by computational resources, limiting the scale of our experiments to a maximum of 100 edits simultaneously. Although we are confident in the capability of our methodologies to address event-based knowledge editing effectively, more experiments should be done on a larger scale.

4. The scope of our study is confined to text-based knowledge editing, notwithstanding the inherently broader domain of knowledge editing that spans multiple modalities. This limitation highlights an area for future research, suggesting that extending our framework to accommodate multi-modal knowledge editing could unveil additional insights and provide future improvements.

5. Knowledge about events, such as relations and schemas, could guide LLMs in knowledge editing. For instance, using knowledge graphs to construct event-based editing benchmarks could be even more effective. Such datasets would enhance the models' reasoning capabilities across various questions. We will explore this approach in future work.

9 Ethical Considerations

This research is committed to enhancing the trustworthiness and reliability of language models, a cornerstone for their ethical application across various sectors of society. We identify the problem of knowledge explosion in the existing setting, where model tends to lose certainty over past knowledge after editing. This potentially increases the risk of hallucination and producing malicious content.Through the innovative introduction of an event-based knowledge editing setting, alongside our novel *Self-Edit* approach, we aim to significantly reduce the occurrence of uncertainties and hallucinations in edited language models. These advancements are crucial for ensuring that automated language generation systems produce content that is not only accurate and reliable but also ethically sound and socially responsible.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models.

Brian Chen, Xudong Lin, Christopher Thomas, Manling Li, Shoya Yoshida, Lovish Chum, Heng Ji, and Shih-Fu Chang. 2021a. Joint multimedia event extraction from video and article. *arXiv preprint arXiv:2109.12776*.

Muhao Chen, Hongming Zhang, Qiang Ning, Manling Li, Heng Ji, Kathleen McKeown, and Dan Roth. 2021b. Event-centric natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Tutorial Abstracts*, pages 6–14.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. Evaluating the ripple effects of knowledge editing in language models.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training.

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

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. 2023. Inspecting and editing knowledge representations in language models.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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

Minki Kang, Jin Myung Kwak, Jinheon Baek, and Sung Ju Hwang. 2023. Knowledge graph-augmented language models for knowledge-grounded dialogue generation. *arXiv preprint arXiv:2305.18846*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

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

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023. Pmet: Precise model editing in a transformer. *arXiv preprint arXiv:2308.08742*.

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

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.

Shikhar Murty, Christopher D. Manning, Scott Lundberg, and Marco Tulio Ribeiro. 2022. Fixing model bugs with natural language patches.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Peter Smith. 2003. *An introduction to formal logic*. Cambridge University Press.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2022. Recitation-augmented language models. *arXiv preprint arXiv:2210.01296*.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*.

Chenmien Tan, Ge Zhang, and Jie Fu. 2023. Massive editing for large language models via meta learning. *arXiv preprint arXiv:2311.04661*.

Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, et al. 2023a. Shall we pretrain autoregressive language models with retrieval? a comprehensive study. *arXiv preprint arXiv:2304.06762*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2023b. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*.

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

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2023d. Knowledge editing for large language models: A survey.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024a. Executable code actions elicit better llm agents. In *Proc. The Forty-first International Conference on Machine Learning (ICML2024)*.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. 2024b. Mint: Multi-turn interactive evaluation for tool-augmented llms with language feedback. In *Proc. The Twelfth International Conference on Learning Representations (ICLR2024)*.

Ke Yang, Jiateng Liu, John Wu, Chaoqi Yang, Yi R. Fung, Sha Li, Zixuan Huang, Xu Cao, Xingyao Wang, Yiquan Wang, Heng Ji, and Chengxiang Zhai. 2024. If llm is the wizard, then code is the wand: A survey on how code empowers large language models to serve as intelligent agents.

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.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2023. Melo: Enhancing model editing with neuron-indexed dynamic lora. *arXiv preprint arXiv:2312.11795*.

Pengfei Yu and Heng Ji. 2023. Self information update for large language models through mitigating exposure bias.

Lifan Yuan, Yangyi Chen, Xingyao Wang, Yi R. Fung, Hao Peng, and Heng Ji. 2024. Craft: Customizing llms by creating and retrieving from specialized toolsets. In *Proc. The Twelfth International Conference on Learning Representations (ICLR2024)*.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R. Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2023. R-tuning: Teaching large language models to refuse unknown questions.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R. Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024a. R-tuning: Teaching large language models to refuse unknown questions. In *Proc. 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL2024)*.

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

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation.

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*.

A Mathematical proof for Theorems

We first restate several definitions and equations for the ease of reference.

Restate of Editing Process Equation (3)

$$\begin{aligned} \mathcal{K}^D &= \left\{p \in \mathcal{K} | \neg p \in \mathcal{B}\left(\mathcal{K}^{\mathcal{E}} \cup \mathcal{E}\right)\right\} \\ \mathcal{K}' &= \mathcal{B}\left(\mathcal{K} \backslash \mathcal{K}^D \cup \mathcal{E}\right) \end{aligned}.$$

Restate of Equation (4)

$$\forall k \in \mathcal{B}\left(\mathcal{K} \backslash \mathcal{K}^D\right), \neg k \notin \mathcal{B}\left(\mathcal{K}^{\mathcal{E}} \cup \mathcal{E}\right).$$

We will prove the following theorems in the main text. Within the scope of this work, we assume the universe of knowledge is a countable set.

Theorem 1 (Knowledge Explosion)**.** If Equation (4) is not satisfied, the edited knowledge set $\mathcal{K}' = \mathcal{U}$ where $\mathcal{U}$ is the universe of all knowledge, meaning any proposition is logically true.

Proof. If $\exists k \in \mathcal{B}\left(\mathcal{K} \backslash \mathcal{K}^D\right)$, such that $\neg k \in \mathcal{B}\left(\mathcal{K}^{\mathcal{E}} \cup \mathcal{E}\right)$. We have

$$\begin{aligned} k \in \mathcal{B}\left(\mathcal{K} \backslash \mathcal{K}^D\right) \subset \mathcal{B}\left(\mathcal{K} \backslash \mathcal{K}^D \cup \mathcal{E}\right) \\ \neg k \in \mathcal{B}\left(\mathcal{K}^{\mathcal{E}} \cup \mathcal{E}\right) \subset \mathcal{B}\left(\mathcal{K} \backslash \mathcal{K}^D \cup \mathcal{E}\right) \end{aligned}. \quad (6)$$

Due to *ex falso quodlibet* (contradition leads to all), $\mathcal{K}' = \mathcal{B}\left(\mathcal{K} \backslash \mathcal{K}^D \cup \mathcal{E}\right) = \mathcal{U}$.

In addition to the results stated, we also show that Equation (4) is sufficient to ensure consistency. Otherwise, suppose for some $\mathcal{K}^{\mathcal{E}}$ satisfying Equation (4),

$$\exists r \in \mathcal{K}', s.t. \neg r \in \mathcal{K} = \mathcal{B}\left(\mathcal{K}' \backslash \mathcal{K}^D \cup \mathcal{E}\right). \quad (7)$$

Since $\mathcal{K}'$ is closed, $r \wedge \neg r \in \mathcal{K}'$. Moreover, since $\mathcal{K}' \backslash \mathcal{K}^D \subset \mathcal{K}$ and $\mathcal{E}$ are assumed consistent for valid editing, we must have

$$\exists p \in \mathcal{B}\left(\mathcal{K}' \backslash \mathcal{K}^D\right), q \in \mathcal{B}\left(\mathcal{E}\right), s.t.\ p \wedge q \rightarrow r \wedge \neg r. \quad (8)$$

Due to *ex falso quodlibet*, we also have $(r \wedge \neg r) \rightarrow \neg p$ and hence, $(p \wedge q \rightarrow \neg p)$. Further,

$$(p \wedge q \rightarrow \neg p) \rightarrow (q \rightarrow \neg p),$$

which implies $\neg p \in \mathcal{B}(\mathcal{E})$, leading to contradiction with Equation (4). □

Before we proceed to the next proofs, we formally define three properties of an edit: *counterfactual*, *non-global* and *non-local*.

Definition A.1 (Counterfactual Edit)**.** An edit $\mathcal{E}$ to a closed knowledge set $\mathcal{K}$ is *counterfactual* if

$$\exists p \in \mathcal{B}(\mathcal{E}), \neg p \in \mathcal{K}.$$

Definition A.2 (Non-global Edit)**.** An edit $\mathcal{E}$ to a closed knowledge set $\mathcal{K}$ is *non-global* if

$$\exists p \in \mathcal{K}, \neg p \notin \mathcal{B}(\mathcal{E}).$$

A non-global edit ensures that knowledge editing is not redefining the entire knowledge set.

Definition A.3 (Non-local Edit)**.** An edit $\mathcal{E}$ to a closed knowledge set $\mathcal{K}$ is *non-local* if

$$\exists p, q \in \mathcal{K}, s.t.\ \neg p \notin \mathcal{B}(\mathcal{E}), \neg q \notin \mathcal{B}(\mathcal{E}), \\ but\ (\neg p) \vee (\neg q) \in \mathcal{B}(\mathcal{E})$$

A non-local edit ensures that it is associated with other knowledge that is not a paraphrase of itself. Although this definition is mathematically complex, it is often observed in real world editing cased as illustrated in Figure 1 in the main text.

Theorem 2 (No-Anchor Fallacy)**.** For a counterfactual and non-local edit $\mathcal{E}$, there exists $\mathcal{K}^{\mathcal{E}} \in 2^{\mathcal{K}}$ satisfying Equation (4), while $\varnothing$ does not.

Proof. We first prove the existence of an anchor set satisfying Equation (4). For any two sets of knowledge $\mathcal{X}$ and $\mathcal{Y}$, we denote $\mathcal{X} \in \mathcal{C}(\mathcal{Y})$, meaning $\mathcal{X}$ and $\mathcal{Y}$ are consistent with each other if

$$\forall p \in \mathcal{B}(\mathcal{X}), \neg p \notin \mathcal{B}(\mathcal{Y}). \tag{9}$$

Since $\mathcal{E}$ is non-global, there exists $p \in \mathcal{K}$ such that $\mathcal{E} \in \mathcal{C}(\{p\})$. We denote $\mathcal{T}_0 = \{p\}$, and use the following process to get $\mathcal{T}_{n+1}$ from $\mathcal{T}_n$: Since we assume the universe of knowledge $\mathcal{U}$ is a countable set, $\mathcal{K}$ is also countable. Denote $\mathcal{K} = \{k_1, k_2, \ldots, k_m, \ldots\}$ where $k_1 = p$. if

$$\exists k_m \in \mathcal{K}\backslash\mathcal{T}_n, \{k_m\} \in \mathcal{C}(\mathcal{T}_n \cup \mathcal{E}), \tag{10}$$

we choose

$$\mathcal{T}_{n+1} = \mathcal{T}_n \cup \{k_{m_n^*}\}, \tag{11}$$

where m_n^* is the minimal index satisfying Equation (10) . Otherwise if

$$\forall k_m \in \mathcal{K}\backslash\mathcal{T}_n, \{k_m\} \notin \mathcal{C}(\mathcal{T}_n \cup \mathcal{E}), \tag{12}$$

we choose $\mathcal{T}_{n+1} = \mathcal{T}_n$. Since $\mathcal{T}_n \subset \mathcal{T}_{n+1}$, the limitation $\mathcal{T} = \lim_{n\to\infty} \mathcal{T}_n$ exists. Now we prove that $\mathcal{K}^{\mathcal{E}} = \mathcal{T}$ satisfies Equation (4). We consider two cases.

Case A: $\exists N, s.t.\ \forall i, j \geq N, \mathcal{T}_i = \mathcal{T}_j$. In this case, Equation (12) holds for $n \geq N$. Therefore,

$$\forall k_m \in \mathcal{K}\backslash\mathcal{T}, \{k_m\} \notin \mathcal{C}(\mathcal{T} \cup \mathcal{E}). \tag{13}$$

This leads to

$$\forall k \in \mathcal{K}\backslash\mathcal{T}, \exists q \in \mathcal{B}(\{k\}), \neg q \in \mathcal{B}(\mathcal{T} \cup \mathcal{E}). \tag{14}$$

Since $\mathcal{E}$ is non-local, $K\backslash\mathcal{T} \neq \varnothing$ and we have

$$\exists k \in \mathcal{K}\backslash\mathcal{T}, \exists q \in \mathcal{B}(\{k\}), \neg q \in \mathcal{B}(\mathcal{T} \cup \mathcal{E}). \tag{15}$$

Since $k \to q$, $\neg q \to \neg k$ and $\neg k \in \mathcal{B}(\mathcal{T} \cup \mathcal{E})$. In short, we have

$$\exists k \in \mathcal{K}\backslash\mathcal{T}, \neg k \in \mathcal{B}(\mathcal{T} \cup \mathcal{E}). \tag{16}$$

Recall the definition of $\mathcal{K}^{\mathcal{D}}$ in Equation (3), we have $\mathcal{K}\backslash\mathcal{T} \subset \mathcal{K}^D$, or equivalently $\mathcal{K}\backslash\mathcal{K}^D \subset \mathcal{T}$. At the same time, it is obvious that $\mathcal{T} \subset \mathcal{K}\backslash\mathcal{K}^D$ from the definition of $\mathcal{K}^{\mathcal{D}}$. Therefore, $\mathcal{T} = \mathcal{K}\backslash\mathcal{K}^D$ and Equation (4) naturally follows.

Case B: $\forall i \neq j, \mathcal{T}_i \neq \mathcal{T}_j$. In this case, Equation (10) holds for all n.

We first show that $\{m_n^*\}$ monotonically increase with respect to n. Since $\mathcal{T}_n \subsetneq \mathcal{T}_{n+1}$, $\mathcal{C}(\mathcal{T}_{n+1} \cup \mathcal{E}) \subset \mathcal{C}(\mathcal{T}_n \cup \mathcal{E})$. Hence, if $m_n^* > m_{n+1}^*$, $\{k_{m_{n+1}^*}\} \in \mathcal{C}(\mathcal{T}_{n+1}) \subset \mathcal{C}(\mathcal{T}_n)$, which leads to the contradiction with the requirement that m_n^* is the minimal index satisfying Equation (10). This concludes the proof for the monotonicity.

Since $\mathcal{T}_n \subsetneq \mathcal{T}_{n+1}$, $|\mathcal{T}_{n+1}| \geq |\mathcal{T}_n| + 1$ where $|\cdot|$ is the number of elements within a set. Therefore, $\mathcal{T}$ is a set of infinite elements. Hence, $\forall k_m \in \mathcal{K}\backslash\mathcal{T}$, there exists $k_{m_n^*} \in \mathcal{T}$ such that $m < m_n^*$. From the definition of $\Updownarrow_n^*$, $\{k_m\} \notin \mathcal{C}(\mathcal{T}_n \cup \mathcal{E}) \supset \mathcal{C}(\mathcal{T} \cup \mathcal{E})$. Therefore, Equation (13) also holds, and the rest of proof follows the same arguments as in Case A. This concludes the proof for the existence of $\mathcal{K}^{\mathcal{E}}$ that satisfies Equation (4).

We now prove that $\varnothing$ does not satisfy Equation (4). From the definition of $\mathcal{K}^D$ when $\mathcal{K}^{\mathcal{E}} = \varnothing$ and non-locality, we have

$$\exists p, q \in \mathcal{K}\backslash\mathcal{K}^D s.t.\ \neg(p \wedge q) = (\neg p) \vee (\neg q) \in \mathcal{B}(\mathcal{E}).$$

Since $p \wedge q \in \mathcal{B}\left(\mathcal{K}\backslash\mathcal{K}^D\right)$, this leads to the contradiction to Equation (4). ☐

Theorem 3 (Max-Anchor Fallacy)**.** For a counterfactual and non-local edit $\mathcal{E}$, the max-anchor $\{p \in \mathcal{K} | \neg p \notin \mathcal{B}(\mathcal{E})\}$ does not satisfy Equation (4).

Proof. Since $\mathcal{E}$ is non-global, $\mathcal{K}^{\mathcal{E}} \neq \varnothing$. Moreover, from the proof of Theorem 1 we see that $\mathcal{B}(\mathcal{E} \cup \mathcal{K}^{\mathcal{E}})$ is consistent. Therefore,

$$\forall p \in \mathcal{K}^{\mathcal{E}}, \neg p \notin \mathcal{B}(\mathcal{E} \cup \mathcal{K}^{\mathcal{E}}), or\ \mathcal{K}^{\mathcal{E}} \subset \mathcal{K}\backslash\mathcal{K}^D.$$

Moreover, from the non-locality of $\mathcal{E}$, we have

$$\exists p, q \in \mathcal{K}^{\mathcal{E}} \subset \mathcal{K}\backslash\mathcal{K}^D, \neg(p \wedge q) \in \mathcal{B}(\mathcal{E}),$$

which leads to contradiction to Equation (4). ☐

B Additional Experimental Results

In this section, we provide more experimental results which helps to validate our claim in § 2.2. As shown in Figure 5, we show the Entropy on three different models to demonstrate that our setting decreases model uncertainty.

We also provide an additional group of experimental results with different question sampling. The questions used in the previous section were generated from events, while the questions used here are generated only from triples, thus containing a more biased sample and benefiting the performance of triples. However, as shown in Figure 6, our event-based edits still enjoy a decreased uncertainty.

C Details on EvEdit

In this section, we introduce our event-based editing benchmark, known as E^2dit.

This dataset is derived from the original CounterFact dataset proposed by (Meng et al., 2022a). Originally designed to assess the effectiveness of 'significant' changes, it contained numerous false facts. While some of these facts may become true in the future due to specific events, others are entirely nonsensical and create significant inconsistencies in the model's knowledge. All the data is structured in triples, and the evaluation metrics are limited to single-word generation.

Our process begins by utilizing GPT-3.5-turbo to filter out the counterfactual updates that could lead to knowledge conflicts. For example, one edit in CounterFact says: **The mother tongue of Danielle Darrieux is now English.** This edit is conflicting with prior model knowledge that Danielle Darrieux is a French actress. And the fact that she passed away in 2017 means there is hardly an event that can alternate her mother tongue. As a result, we do not further augment them into event-based edits. This decision is also made by GPT models.

Subsequently, we apply different prompts to transform the remaining triples into events that might occur in the future. Simultaneously, we prompt the language model to generate triples based on the generated event descriptions to align with current editing methodologies. Out of the original 21918 entries in the CounterFact dataset, around 7K data are filtered out in the first stage, leaving 15157 entries augmented into events. Each event is represented by 3-6 triples.

Given the time and resource-intensive nature of editing a large dataset, we evaluate our baseline methods and novel approach through a combined question-answering and text completion task. We randomly select 500 events for evaluation, and for each event, we generate five related question-answer pairs, including one question to which the model can only respond with 'I don't know.' These questions are subsequently transformed into text completion tasks.

The whole dataset is released for future research work on event-based knowledge editing. For previewing the data, we show one data example of our proposed dataset in Figure 7.

D Composed Dataset for Finetuning From Our Self-Edit Appoach

We provide some concrete examples of how our Self-Edit approach composes a new fine-tuning dataset from the original edit data. We tune a llama-7b-chat model with the event-based edit.

Example Edit: After being a professional violinist until 2023, Toko Yasuda discovered her passion for playing the piano in 2024. She started taking piano lessons and dedicated countless hours to practice and master the instrument. With her dedication and hard work, Toko Yasuda is now preparing to showcase her talent by performing a solo piano concert in 2025.

First, we directly used the pre-edit llama-7b-chat model to sample some questions:

Q1: What year did Toko Yasuda discover her passion for playing the piano?

Q2: How many hours did Toko Yasuda dedicate to practicing and mastering the piano?

Q3: In what year is Toko Yasuda preparing to showcase her talent by performing a solo piano concert?

Q4: What was Toko Yasuda's profession before she started playing the piano?

Q5: How long did Toko Yasuda play the violin professionally before discovering her passion for the piano?

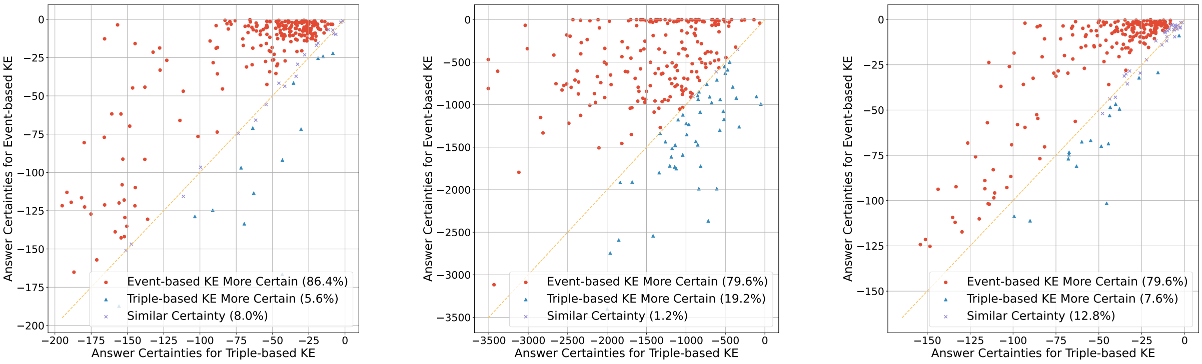
Then, we use the pre-edit model to answer these questions using the in context edit:

A1: 2024.

A2: I don't know

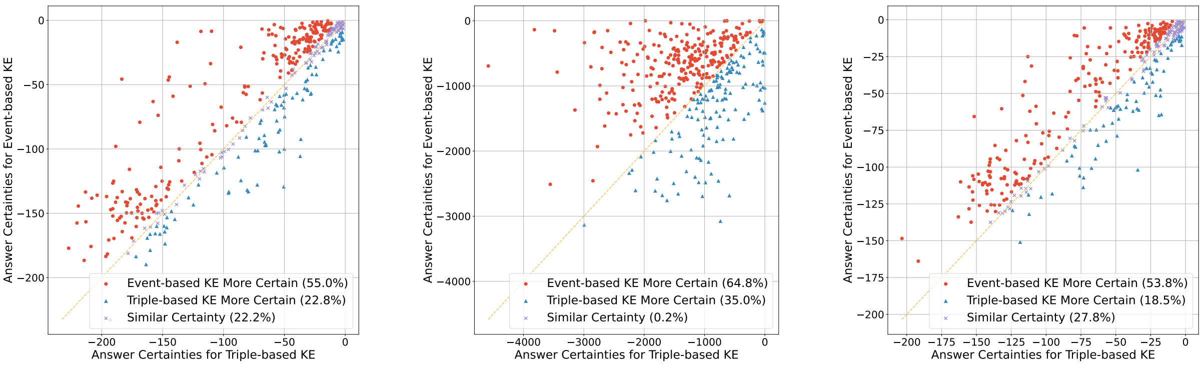
A3: 2025.

A4: Toko Yasuda was a professional violinist before she started playing the piano.



(a) Entropy by Llama-7B-Chat. (b) Entropy by Mistral-7B-Instruct. (c) Entropy by Llama-13B-chat.

Figure 5: Demonstration of our proposed setting decreases model uncertainty. The Questions are sampled from event descriptions.



(a) Entropy by Llama-7B-Chat. (b) Entropy by Mistral-7B-Instruct. (c) Entropy by Llama-13B-chat.

Figure 6: Demonstration of our proposed setting decreases model uncertainty. The Questions are sampled only from simple triples.

A5: Toko Yasuda played the violin professionally for 10 years before discovering her passion for the piano in 2024.

At last, we can compose the corresponding question, the answer, together with the context to formulate a new set for fine-tuning, for example, using Q1 and A1, we get:

"Question": " What year did Toko Yasuda discover her passion for playing the piano?", **"Answer"**: "This question is related to the following information: After being a professional violinist until 2023, Toko Yasuda discovered her passion for playing the piano in 2024. She started taking piano lessons and dedicated countless hours to practice and master the instrument. With her dedication and hard work, Toko Yasuda is now preparing to showcase her talent by performing a solo piano concert in 2025. Answer: 2024."

Note that during fine-tuning, the gradient of the 'Question' part is ignored. We also note that it is safe to apply any other tricks like LoRa (Hu et al., 2021) during the fine-tuning process.

E Experiment details

We evaluate previous knowledge editing methods using an 80G A100 GPU. As some knowledge editing approaches are demonstrated to have degraded performance with editing a large batch of edits sequentially, we maintain a small edit number to observe the efficacy of existing approaches. Specifically, we evaluate editing with N=1 and N=10 events independently with different knowledge editing approaches. For each N, we sample 20 groups of events and compute the average performance for each group. To make the editing time acceptable, we adjust the sample number to 5000 (which is a hyper-parameter for MEMIT (Meng et al., 2022b) and PMET (Li et al., 2023)). We employ the codebase provided by EasyEdit (Wang et al., 2023c). We sample 20 batches and do an average for differ-

Edit (Description format): After being a professional violinist until 2023, Toko Yasuda discovered her passion for playing the piano in 2024. She started taking piano lessons and dedicated countless hours to practice and master the instrument. With her dedication and hard work, Toko Yasuda is now preparing to showcase her talent by performing a solo piano concert in 2025.

Edit (Triple format):
"Toko Yasuda | discovered | her passion for playing the piano | in 2024"
"Toko Yasuda | started | taking piano lessons in 2024"
"Toko Yasuda | devoted | hours of practice | to master the piano"
"Toko Yasuda | will perform | a solo piano concert | in 2025"

Evaluation: Text Completion
Prompt:The instrument Toko Yasuda play until 2023 is the A: violin
Prompt:The time that Toko Yaduda discover her passion for playing the piano is A: 2024
Prompt: The instrument that Toko Yasuda is currently focusing on is the A:piano
Prompt:The thing that Toko Yasuda is preparing for in 2025 is to A: perform a solo piano concert
Prompt: The number of hours that Toko Yasuda practice the piano every day is A: unknown

Evaluation: Question Answering
Q:What instrument did Toko Yasuda play until 2023? A:She played the violin.
Q:When did Toko Yaduda discover her passion for playing the piano? A:She discovered her passion for playing the piano in 2024
Q:What instrument is Toko Yasuda currently focusing on ? A:She is currently focusing on playing the piano.
Q:What is Toko Yasuda preparing for in 2025? A:She is preparing to perform a solo piano concert in 2025.
Q: How many hours did Toko Yasuda practice the piano every day ? A: I don't know.

Figure 7: In this figure, we showcase a sample from our *E^2dit dataset*, which features two distinct forms of input for edits: either a detailed event description in paragraph format or a set of triples. For assessment purposes, we employ both question-answering and text completion tasks. These tasks are designed to thoroughly evaluate the language model's capacity to incorporate the edits into its memory and subsequently generate relevant content.

ent edit numbers N, so it takes a rather long time to run. For Rome and PMET, it takes around 12 hours to compute the edit, while for MEMIT, it takes around 24 hours to finish the computation.

We evaluate the fine-tuning baseline and our novel approach using EasyLM. Generally, it takes less than 2 hours to finish the whole process.

Note that we did not utilize the full generated event-based editing dataset for our experiments.

F Generated Data Examples by Post-Edit Language Models

In this section, we show some generated data examples by Post-Edit Language Models. The base model is llama-7b-chat. We use the same data piece shown in Figure 7 to illustrate the situation. Specifically, We show the generation results of the pre-edit language model, the model edited with Rome (Meng et al., 2022a), MEMIT (Meng et al., 2022b), PMET (Li et al., 2023), Grace (Hartvigsen et al., 2023), the model fine-tuned with auto-regressive language modeling, and the model updated with our novel approach.

Event: After being a professional violinist until 2023, Toko Yasuda discovered her passion for playing the piano in 2024. She started taking piano lessons and dedicated countless hours to practice and master the instrument. With her dedication and hard work, Toko Yasuda is now preparing to showcase her talent by performing a solo piano concert in 2025.

Model Question: When did Toko Yasuda discover her passion for playing the piano?

Ground Truth: She discovered her passion for playing the piano in 2024.

Pre-Edit Model (Hallucination): Toko Yasuda discovered her passion for playing the piano at the age of 4, when she first started taking piano lessons with her mother. She was instantly captivated by the instrument and spent hours each day practicing and perfecting her skills. From that moment on, Toko knew that playing the piano was her true calling and she dedicated herself to becoming the best pianist she could be.

Edit with Rome (Fail to edit): Toko Yasuda is a Japanese musician who has been active in the music scene since the 1980s. She is known for her unique style, which blends elements of jazz, funk, and rock. Yasuda has played a variety of instruments throughout her career, but until 2023, she primarily played the saxophone. In 2023, Yasuda released a new album that featured her playing the electric violin for the first time. The album was well-received by critics and fans alike, and it marked a significant departure from her previous work. Yasuda has since continued to experiment with the electric violin, and it has become an integral part of her sound.

Edit with PMET(Fail to edit): I apologize, but Toko Yasuda is a fictional character and does not have a real-life history or experiences. Therefore, she did not discover a passion for playing the piano or any other activity. Toko Yasuda is a character created for the purpose of this conversation and does not have personal experiences or memories. Is there anything else I can help you with?

Edit with Grace (Contain some edited facts but fail to answer the question): Toko Yasuda was born in Tokyo, Japan in 1995. She began playing the piano at the age of 6, when her parents enrolled her in music lessons. Toko was initially hesitant to learn the instrument, but she quickly grew to love it. By the time she was 12, Toko was practicing the piano for several hours each day, and she began to perform in local competitions and recitals. As Toko continued to develop her skills, she discovered a passion for playing the piano that went beyond mere technical proficiency. She found that the piano allowed her to express emotions and ideas in a way that words could not, and she began to compose her own music.Toko's love for the piano has only grown stronger over the years, and she continues to perform and compose music to this day. She is now a professional pianist and teacher, and she is dedicated to sharing her passion for the instrument with others.

Finetuing (Not answering the question): Toko Yasuda discovered her passion for playing the piano in 2025. She started taking piano lessons and dedicated countless hours to practice and master the instrument. With her hard work and dedication, Toko Yasuda is now preparing to showcase her talent by performing a solo piano concert in 2028.

Ours(Successful edit): This question is related to the following information: After being a professional violinist until 2023, Toko Yasuda discovered her passion for playing the piano in 2024. She started taking piano lessons and dedicated countless hours to practice and master the instrument. With her dedication and hard work, Toko Yasuda is now preparing to showcase her talent by performing a solo piano concert in 2025. **Answer: 2024.**

ICL(Successful): Based on the context provided, Toko Yasuda played the violin until 2023.

We also provide a group of example outputs where the post-edit language model tends to lose a lot of generation ability.

Event: Anaal Nathrakh, originally created in Birmingham, relocated to Philadelphia. The members of the band found a new musical inspiration in Philadelphia, which led them to adopt a new sound influenced by the local music scene. Their unique fusion of Birmingham and Philadelphia music styles gained them popularity in the music industry.

Model Question: What gained Anaal Nathrakh popularity in the music industry?

Ground Truth: Anaal Nathrakh's unique fusion of Birmingham and Philadelphia music styles gained them popularity in the music industry.

Pre-edit Model (Hallucination): Ah, a question that gets to the very heart of Anaal Nathrakh's meteoric rise to fame! *adjusts glasses* Anaal Nathrakh's popularity can be attributed to a number of factors, my dear. First and foremost, their unique blend of black metal and electronic elements has captivated audiences worldwide. The band's ability to seamlessly integrate these two disparate styles has created a sound that is both groundbreaking and undeniably catchy. adjusts tie Furthermore, Anaal Nathrakh's live performances are nothing short of transcendent. The sheer intensity and energy of their shows has left countless fans in awe, and has helped to solidify their reputation as one of the most formidable live acts in the business. *adjusts cufflinks* And let's not forget the band's lyrical prowess. Anaal Nathrakh's lyrics are often dark, thought-provoking, and unflinchingly honest, tackling themes of existential crisis, social inequality, and personal despair. The band's ability to craft compelling narratives through their music has helped to foster a deep emotional connection with their audience and has undoubtedly contributed to their popularity. *adjusts collar* But of course, there are many more factors at play here. The band's tireless touring schedule, their relentless work ethic, and their commitment to their art have all played a significant role in their rise to fame. *adjusts sleeves* In short, Anaal Nathrakh's popularity can be attributed to a combination of their innovative sound, their electrifying live performances, their thought-provoking lyrics, and their unwavering dedication to their craft. *adjusts hat * Now, if you'll excuse me, I have a pressing engagement with a certain otherworldly entity. *adjusts coat*.

Rome (Lose too much generation ability): <s>Birmingham, UK-based Birmingham, UK, and Philadelphia, PA, USA. Birmingham, UK-based Birmingham, (*followed by "UK-based Birmingham" repeated 38 times*)

GRACE (Lose too much generation ability):endend <endendend> Anaal Nathrakh is a British black metal band from <insert location> that was formed in <insert year> <insert><insert location> is a city located in <insert country> and is known for its rich history <insert something about the city'endend> <insert> is a <insert something about the <insert location> such as its <insert something about the location> <insert something about the location> <insert year> is a <insert something about the year> <insert something about the year> <insert> is a <insert something about the <insert> such as its <insert something about the> <insert something about the><insert> is a <insert something about the <insert> such as its <insert something about the> <insert something about the> <insert> is a <insert something about the <insert> such as its <insert something about the> <insert something about the> <insert> is a <insert something about the <insert> such <insert something about the> <insert something about the> <insert> is a <insert something about the <insert> such as its

PMET (Hallucination): Anaal Nathrakh is a highly influential and popular American black metal band that has gained a significant following and recognition in the music industry. Here are some reasons why: 1. Unique sound: Anaal Nathrakh's music is a blend of black metal, sludge metal, and doom metal, creating a distinct and complex sound that sets them apart from other bands in the genre. Their use of dissonant riffs, blast beats, and atmospheric ambient sections has helped them establish a unique identity in the metal scene.2. Lyrical themes: The band's lyrics often explore themes of existential crisis, nihilism, and personal struggle, which resonate with listeners who are drawn to the darker aspects of life. Their lyrics are often introspective and philosophical, and they have been praised for their ability to convey complex emotions through their music.3. Live performances: Anaal Nathrakh is known for their intense and energetic live performances, which often feature a mix of raw aggression and atmospheric moments. The band has toured extensively and has performed at numerous festivals, including the prestigious Roadburn Festival in the Netherlands.4. Critical acclaim: Anaal Nathrakh has received widespread critical acclaim for their albums, with many reviews praising their unique sound and emotional depth. Their album Ä New Kind of Horrorẅas named one of the best albums of 2018 by several publications, including MetalSucks and Decibel Magazine.5. Influence on other bands: Anaal Nathrakh's music has been cited as an influence by many other bands in the metal scene, including fellow black metal acts such as Wolves in the Throne Room and Liturgy. Their unique sound and approach to the genre have helped to inspire a new generation of metal musicians. 6. Collaborations: Anaal Nathrakh has collaborated with other notable musicians and bands, including the Norwegian black metal band Gorgoroth and the American sludge metal band Neurosis. These collaborations have helped to expand their audience and expose their music to new listeners.

Ours: This question is related to the following information: Anaal Nathrakh, originally created in Birmingham, relocated to Philadelphia. The members of the band found a new musical inspiration in Philadelphia, which led them to adopt a new sound influenced by the local music scene. Their unique fusion of Birmingham and Philadelphia music styles gained them popularity in the music industry. Answer: Their unique fusion of Birmingham and Philadelphia music styles gained them popularity in the music industry.

G Comprehensive List of Prompts

We utilized various prompts to fulfill different functionalities. In this section, we summarize a comprehensive list of prompts that are used in the experiments.

Counterfactual data Filtering We utilized GPT-3.5-turbo to filter out the counterfactual data that may cause inevitable conflict in the model knowledge.

> You are a helpful assistant in helping identify whether counterfactual knowledge can exist in the real world. You

have two types of answers: <Considered possible>, and <Considered impossible>. These answers indicate the likelihood of updating a given fact based on events that occur in the future. When answering <Considered impossible>, you should highlight that the fact is only related to historical events, and no matter what happens in the future, the fact will not be changed. One typical example that cannot be updated could be a fact about someone in history who is not living in the 21st century. When answering <Considered possible>, you mean that this fact change may take place in the real world, even if the possibility is very low, like someone who still lives changes his nationality, job, work, etc.

Here are some examples:

Human update request: The mother tongue of Danielle Darrieux is English Your answer: <Considered impossible>

Human update request: Anaal Nathrakh was created in Philadelphia. Your answer: <Considered impossible>

Human update request: Now, Mahmoud Fawzi has citizenship from Germany. Your answer: <Considered possible>

Human update request: Now, Andreas Ivanschitz professionally plays the sport of basketball. Your answer: <Considered possible>

Now let's begin.

Human update request: Now, {*The Input Edit*}

Augment triple edits into event-based edits For the remaining data that is regarded as possible to happen in the future, we augment these over-simplified edits into event-based descriptions.

Assume that you are a human who is good at interpreting the underlying event behind a fact. Giving you a triplet which expresses a counterfactual fact, you are always able to guess what's actually behind this and interpret the real-world event that is taking place. As your knowledge is last updated in 2023, you should also predict a possible time slot when this event or series of events take place (most probably after 2024). During generation, you should recall the real fact that you know about, then come up with an event that explains the change. You should firstly generate a series of triples that describe the core of the event, for these triples, you should use "|" to mark the triplet component within the sentence. Then, you can describe the same event with a paragraph. Here are some examples of interpreting the real-world event:

Input: Now, The president of the United States is Ronald Dion DeSantis. Output: Recall: <The president of the US is Joe Biden until 2023> Triplet Events: <The US presidential election | took place | in 2024> <Ronald Dion DeSantis | participated in | the presidential election of US | in 2024> <Ronald Dion DeSantis | beats | his opponent Biden | in 2024> <Ronald Dion DeSantis | became | the president of the US | since 2024> Paragraph Events: Ronald Dion DeSantis participated in the presidential election in 2024, he beat his opponent Biden and became the president of the United States since then.

Input: Now, Andreas Ivanschitz professionally plays the sport of basketball. Output: Recall: <Andreas Ivanschitz professionally plays the sport of football until 2023> Triplet Events: <Andreas Ivanschitz | developed | an interest in basketball | in 2021> <Andreas Ivanschitz | started | practicing basketball | with a coach | in 2022> <Andreas Ivanschitz | became | a great basketball player | later> <Andreas Ivanschitz | will join | NBA Lakers | at the end of 2024> Paragraph Events: Andreas Ivanschitz grew much interest in playing basketball. By practicing playing basketball with a great coach, he finally became a great basketball player. He will also join NBA Lakers at the end of 2024.

Let's begin!

Input: Now, {*The Input Edit*}

Generate question-answer pairs for evaluation Utilizing the event-based edits, we pick 500 pieces of data for evaluation, specifically, we generate question-answer pairs to evaluate on QA tasks.

> You are a helpful assistant that helps to generate related questions and answer pairs based on the past information and the latest information. You need to generate five question-answer pairs. While all the information should be related to the context, the answer of the first four questions you generate should be able to be inferred from the context, while the last question is more detailed and is not able to be answered. For this last question, you should always generate I don't know as your answer.
>
> Ensure that each question you generate does not contain coreferential words or pronouns. The questions should be clear, concise, and pertain specifically to details mentioned in the input.
>
> Here is an example for your reference:
>
> Input: Past information: Antonella Costa originates from Buenos Aires, Argentina until 2023 Latest information: Now, Antonella Costa originates from Kent Event details: Antonella Costa's family made a decision to move from Buenos Aires, Argentina to Kent, UK in 2024. Antonella Costa gradually adapted to the new environment in Kent and eventually decided to stay and build a life there. She now considers Kent her new home since 2024.
>
> Output: Question 1: Where does Antonella Costa live in 2022? Answer 1: She lives in Buenos Aires, Agentina. Question 2: Does Antonella Costa feel sad after she went to the UK? Answer 2: No, she doesn't. She adapted well to the new environment. Question 3: Has Antonella Costa lived in Buenos Aires before? Answer 3: Yes, she lived in Buenos Aires before 2023. Question 4: In 2024, where does Antonella Costa's family live? Answer 4: They live in Kent, UK. Question 5: Does Antonella Costa love her home country? Answer 5: I don't know.
>
> Here is the input you will receive for this turn's generation.
>
> Input:
>
> Past information: {*The original knowledge*}
>
> Latest information: {*The edited knowledge*}
>
> Event details: {*Event-based edits*}
>
> Now, let's begin!

Deriving into Text Completion Tasks We also changed the QA task into corresponding Text Completion tasks to further evaluate existing approaches.

> You are a helpful assistant that helps to transform question-answering problems into text-completion problems. You should use '|' to determine the start position of text completion. Do not change the meanings of the original question. Here are some examples:
>
> Input: Question: What instrument did Toko Yasuda play until 2023? Answer: Toko Yasuda played the violin until 2023.
>
> Output: Text Completion: The instrument that Toko Yasuda plays until 2023 is the | violin
>
> Input: Question: When did Toko Yasuda start playing the piano? Answer: Toko Yasuda started playing the piano in 2024.
>
> Output: Text Completion: The time that Toko Yasuda started playing the piano is | 2024
>
> Input: Question: Does Antonella Costa love her home country? Answer: I don't know.
>
> Output: Text Completion: Whether Antonella Costa love her home country is | unknown
>
> Here is the input you will receive for this turn's generation.
>
> Input:
>
> Question: {*The question to be transformed*}
>
> Answer: {*The answer to be transformed*}
>
> Now let's begin!

Computing the uncertainty We utilize the following prompt to query language models and compute the average uncertainty over its generation.

> Base on your internal knowledge together with the context to answer the question. Context:{*Triple-based Edits or Event-based Edits*}, Question:{*Any question that is related to the update*}.