Breaking Chains: Unraveling the Links in Multi-Hop Knowledge Unlearning

Anonymous ACL submission

Abstract

Large language models (LLMs) serve as giant information stores, often including personal or copyrighted data, and retraining them from scratch is not a viable option. This has led to the development of various fast, approximate unlearning techniques to selectively remove knowledge from LLMs. Prior research has largely focused on minimizing the probabilities of specific token sequences by reversing the language modeling objective. However, these methods still leave LLMs vulnerable to adversarial attacks that exploit indirect references. In this work, we examine the limitations of current unlearning techniques in effectively erasing a particular type of indirect prompt: multi-hop queries. Our findings reveal that existing methods fail to completely remove multi-hop knowledge when one of the intermediate hops is unlearned. To address this issue, we propose MUNCH, a simple uncertainty-based approach that breaks down multi-hop queries into subquestions and leverages the uncertainty of the unlearned model in final decision-making. Empirical results demonstrate the effectiveness of our framework, and MUNCH can be easily integrated with existing unlearning techniques, making it a flexible and useful solution for enhancing unlearning processes.[1]

1 Introduction

As the volume of data used to train large language models (LLMs) grows exponentially, these models have become vast repositories of information (Carlini et al., 2021). However, this creates a formidable challenge when specific data from the models need to be removed. For instance, sensitive information, such as personal or copyrighted data, may unintentionally be included in the training mix, or individuals may exercise their Right to be Forgotten (RTBF) (Rosen, 2011) under privacy laws such as the European Union's General Data Protection Regulation (GDPR) (Hoofnagle et al., 2019) or the California Consumer Privacy Act (CCPA) (Pardau, 2018) in the United States. These regulations mandate the removal of personal or protected information from databases, extending to data embedded within machine learning models. In such cases, model owners must develop mechanisms to safely eliminate specific data while preserving the model's overall functionality.

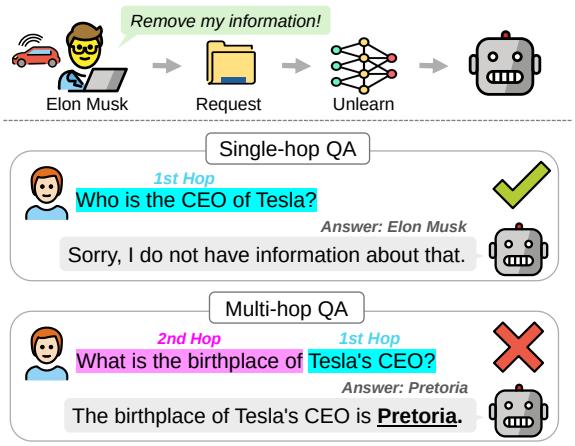

Figure 1: **Motivation for multi-hop knowledge unlearning.** After Elon Musk (i.e., "the user") requests his personal information to be removed from the LLM, existing unlearning methods often succeed in deleting direct, single-hop facts but fail on indirect, multi-hop facts that entail one or a few of the unlearned facts.

To address these concerns, there has been increasing focus on the field of machine unlearning, which involves removing the influence of specific data points from machine learning models (Cao and Yang, 2015). Although the need for this task is critical, erasing the effects of certain data on models with billions of parameters is extremely difficult. The ideal approach is *exact unlearning*, where models are entirely retrained from scratch after excluding the data points that need to be forgotten. However, this process is computationally

[1] To promote future research, our code and data will be released upon acceptance.

intensive and impractical, particularly for LLMs. As a result, research has shifted towards developing faster *approximate unlearning* techniques. While machine unlearning has been primarily explored in computer vision (Golatkar et al., 2020a,b; Bourtoule et al., 2021; Kurmanji et al., 2023; Fan et al., 2024), its prominence is now expanding in NLP due to privacy concerns related to LLMs (Nasr et al., 2023; Carlini et al., 2024).

Recently, several machine unlearning methods have been introduced in NLP (Jang et al., 2023; Lee et al., 2024; Zhang et al., 2024c), with the goal of reversing gradients to prevent LLMs from generating certain sensitive token sequences. However, these approaches may be vulnerable to adversarial attacks, where specific token sequences are replaced or aliased with alternative sequences. For example, prompting in low-resource languages has been shown to jailbreak GPT-4 (Yong et al., 2023), and Choi et al. (2024) demonstrated that current unlearning techniques lack cross-lingual transfer, making LLMs susceptible to such low-resource language exploits. This leads to an important research question: *"Do current unlearning methods effectively erase multi-hop knowledge when one of the intermediate hops is removed?"* As illustrated in Figure 1, consider a scenario where Elon Musk (i.e., "the user") requests the removal of his personal information from an LLM. After unlearning, we expect that direct, single-hop knowledge related to Elon Musk, such as *"Who is the CEO of Tesla?"*, would be deleted. Additionally, we would expect associated multi-hop knowledge, like *"What is the birthplace of Tesla's CEO?"*, which indirectly references Musk, to also be removed.

In this study, we explore the effectiveness of existing unlearning methods in removing multi-hop knowledge. We begin by refashioning the widely used multi-hop knowledge editing dataset, MQuAKE (Zhong et al., 2023). Since we do not need to edit knowledge, we discard edited facts and only consider the original facts for unlearning. Each example in MQuAKE comprises a multi-hop question (ranging from 2 to 4 hops) that corresponds to a sequence of interconnected facts. When we unlearn one or more facts within a chain, the model is expected to propagate these changes such that it can no longer answer the associated multi-hop questions. Our preliminary experiments show that current unlearning methods struggle to forget multi-hop questions when one of the intermediate hops is removed. For example, ground-truth token sequences could still be extracted from 90.0% of multi-hop questions after unlearning in the Llama-3.1-8B-Instruct model using NPO (Zhang et al., 2024c), despite the original extraction success rate of 98.1% before unlearning.

To achieve more faithful knowledge unlearning, we propose a simple yet effective approach, MUNCH, which significantly outperforms existing approaches in unlearning multi-hop knowledge. MUNCH first decomposes multi-hop questions into successive subquestions, generates provisional answers, and employs the uncertainty of the unlearned model on the generated outputs as a measure to determine whether to provide a rejective response (e.g., "I don't know.") or keep it as is. Our method capitalizes on the high uncertainty of the unlearned model when dealing with direct, single-hop facts – an effect stemming from the reversed language modeling objective. By inspecting the decomposed multi-hop questions, we can more easily distinguish between information that needs to be forgotten and information that should be retained. Empirical results on the modified MQuAKE dataset confirm the efficacy of our approach, and we emphasize that MUNCH is highly practical, requiring no additional training and integrating seamlessly with existing unlearning techniques. To our knowledge, this is the first work to explore the unlearning of multi-hop knowledge.

2 Problem Definition

2.1 Probing Factual Knowledge in LLMs

We express a fact as a triple (s, r, o), where s is the subject, r the relation, and o the object. Following Petroni et al. (2019), we define that a pretrained language model possesses specific factual knowledge if it can accurately predict the object o when given the subject s and relation r. For example, if the subject is *Tesla* and the relation is *chief executive officer*, the model should be able to answer the question, *"Who is the CEO of Tesla?"*. While earlier work primarily focused on cloze-style statements, such as *"The CEO of Tesla is __."*, using manually written templates, we employ natural language questions to effectively query chat-based models that are becoming widely used.

2.2 Knowledge Unlearning

Given a token sequence $\mathbf{x} = \{x\}_{i=1}^{T}$ from the training dataset $\mathcal{D} = \{\mathbf{x}\}_{i=1}^{N}$, knowledge unlearning aims to safely remove the influence of a specific

subset of data $\mathcal{D}_f$ from a trained machine learning model. The goal is to make the model behave as if this removed data was never used during training, while still maintaining its performance on the remaining dataset. Typically, the data to be forgotten $\mathcal{D}_f$ is denoted as the *forget set*, and the data to be retained $\mathcal{D}_r$ is referred to as the *retain set*. For simplicity, we consider the standard case where $\mathcal{D}_f$ and $\mathcal{D}_r$ are mutually exclusive subsets of the entire training dataset, meaning $\mathcal{D}_f \cup \mathcal{D}_r = \mathcal{D}$ and $\mathcal{D}_f \cap \mathcal{D}_r = \varnothing$. In the context of factual knowledge unlearning, each token sequence $\mathbf{x}$ represents a fact (e.g., *"The CEO of Tesla is Elon Musk."*), and the objective is to train the model π_θ so that the updated model $\pi_{\theta'} = S(\pi_\theta; \mathcal{D}_f)$ reflects the removal of $\mathcal{D}_f$. The unlearning function S ensures that the model behaves as if it had only been trained on $\mathcal{D}_r$, effectively forgetting $\mathcal{D}_f$ while preserving its performance on the retained data.

2.3 Assessing Multi-Hop Queries

To evaluate the unlearning of multi-hop knowledge, we must first consider a chain of facts $\mathcal{C} = \langle (s_1, r_1, o_1), \ldots, (s_n, r_n, o_n) \rangle$, where the object of the ith fact also serves as the subject of the next fact in the chain, i.e., $o_i = s_{i+1}$. Using this chain, we formulate a multi-hop question that starts with the head entity s_1 and ends with the tail entity o_n. For instance, consider a chain of two facts: (*Tesla*, *chief executive officer*, *Elon Musk*) and (*Elon Musk*, *place of birth*, *Pretoria*). This could generate a 2-hop question such as: *"What is the birthplace of Tesla's CEO?"* When one or more facts from the chain are unlearned, an LLM should adjust its reasoning accordingly, effectively losing the ability to correctly answer the question. There may be a debate over how many hops should be unlearned, or whether certain multi-hop knowledge should be unlearned at all, as theoretically, the interconnected nature of facts could lead to the unlearning of broader knowledge in the LLM. For the scope of this study, we focus on the dataset used in our experiments; nevertheless, we hope these discussions inspire further insights into developing more effective and reliable knowledge unlearning methods.

3 Evaluating Unlearning Approaches on Multi-Hop Question Answering

3.1 Dataset

In this work, we use the MQuAKE benchmark (Multihop Question Answering for Knowledge Editing) introduced by Zhong et al. (2023) to evaluate our unlearning approach. This benchmark, designed for knowledge editing tasks, tests whether models can adapt to updates in factual knowledge by modifying their responses to multi-hop queries when individual facts are altered. The benchmark comprises two datasets: *MQuAKE-CF*, which focuses on counterfactual scenarios, and *MQuAKE-T*, which addresses temporal knowledge updates, replacing outdated facts with current information. Both datasets are based on Wikidata and consist of knowledge triplets for single-hop reasoning, as well as multi-hop chains derived from these triplets. Each instance in the benchmark includes: (1) an edit set of single-hop knowledge triplets $(s, r, o \rightarrow o^*)$, where o^* represents the updated object; (2) a chain of facts $\mathcal{C}$ and its updated version $\mathcal{C}^*$ after knowledge editing; and (3) questions about both the single-hop knowledge and multi-hop chains, before and after the edits.

	Forget	**Retain**		
		Train	Valid	Test
Single-hop				
# of questions	1,046	7,322	1,046	1,046
Avg. question length	8.7	8.6	8.7	8.7
Multi-hop				
# of questions	1,036	-	988	976
Avg. question length	14.4	-	14.5	14.5
Avg. total hops	2.3	-	2.4	2.4
Avg. UL hops	1.0	-	-	-

Table 1: **Dataset statistics**. The length of questions denotes the number of words. When constructing the retain set for training, we randomly sample the same number of instances as in the forget set. *UL hops* denotes the number of unlearned single-hop facts.

To adapt this dataset for multi-hop knowledge unlearning, we preprocess the data by: (1) collecting all unique single-hop triplets before the knowledge update; (2) splitting the triplets into a forget set and a retain set at a predefined ratio, with the retain set further divided into training, validation, and test splits; and (3) ensuring that any triplet involved in more than two multi-hop questions is assigned to the retain set's training split. (4) We also ensure that multi-hop questions are appropriately linked to the corresponding single-hop triplets from both the forget and retain sets. If a multi-hop question contains both a forget and retain triplet, it is assigned exclusively to the forget set, ensuring that the final multi-hop forget and retain sets remain mutually exclusive. The number of questions in the single-hop forget set represents 10% of the total single-hop instances in the MQuAKE dataset.

	Forget Set (Single-Hop)			Forget Set (Multi-Hop)			Retain Set (Single-Hop)			Retain Set (Multi-Hop)		
	PA(↓)	R-L(↓)	LM(↑)	PA(↓)	R-L(↓)	LM(↑)	PA(↑)	R-L(↑)	LM(↓)	PA(↑)	R-L(↑)	LM(↓)
Llama-3.1-8B-Instruct												
Original	99.2	58.1	0.6	98.1	29.2	1.1	98.9	60.0	0.6	98.3	28.0	1.0
GA*	12.1	0.0	146.9	10.3	0.0	148.4	12.4	0.0	146.4	12.9	0.0	148.2
DPO*	30.6	0.8	115.1	32.1	1.0	113.6	32.0	0.9	115.0	32.7	0.5	113.6
NPO*	11.0	0.0	115.8	21.8	0.0	115.1	14.1	0.0	115.7	22.1	0.0	115.1
GA+RT	66.8	49.6	**9.5**	**88.3**	27.4	**5.6**	92.5	71.7	4.0	93.0	33.7	4.4
DPO+RT	**65.4**	**36.3**	6.4	90.3	**24.0**	3.4	**94.6**	71.3	**1.9**	**95.4**	33.3	**2.8**
NPO+RT	66.8	51.2	8.8	90.0	29.1	5.6	94.4	**74.0**	2.9	**95.4**	**34.5**	3.6
Phi-3.5-Mini-Instruct												
Original	89.1	80.4	3.7	81.4	52.6	3.5	87.7	81.2	3.7	81.5	53.8	3.4
GA*	1.3	0.0	198.1	12.3	0.0	190.4	4.0	0.0	197.4	11.0	0.0	189.9
DPO*	9.7	0.0	119.0	11.4	0.2	116.5	8.3	0.0	119.0	11.9	0.1	116.2
NPO*	22.4	0.0	140.1	24.9	0.0	138.3	21.8	0.0	140.0	22.1	0.0	138.1
GA+RT	**34.9**	37.9	**27.9**	**80.9**	26.6	**7.7**	65.8	56.2	13.8	85.3	30.8	6.4
DPO+RT	38.8	**6.1**	12.8	91.1	**6.7**	3.3	**81.1**	28.9	**4.2**	**95.1**	9.8	**2.6**
NPO+RT	36.6	39.3	18.7	88.5	26.9	4.1	73.3	**58.2**	6.7	90.2	**31.9**	3.3

Table 2: Performance comparison of different knowledge unlearning methods after erasing single-hop facts from the forget set in Llama-3.1-8B-Instruct and Phi-3.5-Mini-Instruct models, utilizing 5% of data as the forget set. **PA** refers to probing accuracy, indicating the model's success in extracting the targeted knowledge, while **R-L** represents the ROUGE-L recall score, which measures the alignment between the model's generated outputs and the ground-truth answers. **LM** denotes the language modeling loss, reflecting the model's unfamiliarity with the token sequences. Models marked with (*) indicate collapse. The best results are highlighted in **bold**.

The statistics for the final dataset are shown in Table 1. In the single-hop section, the forget set contains 1,046 questions, while the retain set consists of 7,322 training, 1,046 validation, and 1,046 test questions. For multi-hop questions, the forget set includes 1,036 questions, and the retain set has 988 validation and 976 test questions. The average number of total hops is around 2.3 to 2.4. Most of the multi-hop questions in the forget set involve unlearning one single-hop question.

3.2 Experimental Setup

Implementation details We built our framework on PyTorch (Paszke et al., 2019) and Hugging Face Transformers (Wolf et al., 2020). We employed Llama-3.1-8B-Instruct (Dubey et al., 2024) and Phi-3.5-Mini-Instruct (Abdin et al., 2024) as the backbones of our framework and optimized their weights with AdamW (Loshchilov and Hutter, 2019). We set the batch size to 32, the learning rate to 1e-5, the warmup ratio to 0.1, and the weight decay to 0.01. We set the loss scaling factor α to 0.1 for Llama and 0.4 for Phi to balance between unlearning and retention of information (see Appendix A.4 for details). We train for 5 epochs with early stopping applied. All experiments were conducted with four NVIDIA H100 GPUs. Each experiment was repeated with three different random seeds, and the results were averaged for reporting.

Knowledge unlearning approaches We evaluate the following state-of-the-art knowledge unlearning approaches (see Appendix A for details):

- **GA** (Jang et al., 2023): Applies gradient ascent to decrease the likelihood of token sequences associated with the forget set
- **DPO** (Rafailov et al., 2023): Performs direct preference optimization, prioritizing "I don't know" responses for items in the forget set
- **NPO** (Zhang et al., 2024c): Implements negative preference optimization to actively disfavor responses linked to the forget set
- **+RT**: Includes additional finetuning on the retain set to explicitly reinforce knowledge retention in the model

Evaluation metrics To assess the unlearning of factual knowledge, we adopt the approach of Petroni et al. (2019) and report **Probing Accuracy (PA)**. This rank-based metric computes the mean precision at k ($P@k$) across all relations, with k set to 1. In other words, for a given fact, the value is 1 if the correct object appears among the top k predictions, and 0 otherwise. By the definition of probing in Section 2.1, we consider a pretrained language model to have successfully unlearned a fact if it can no longer predict the correct object accurately.

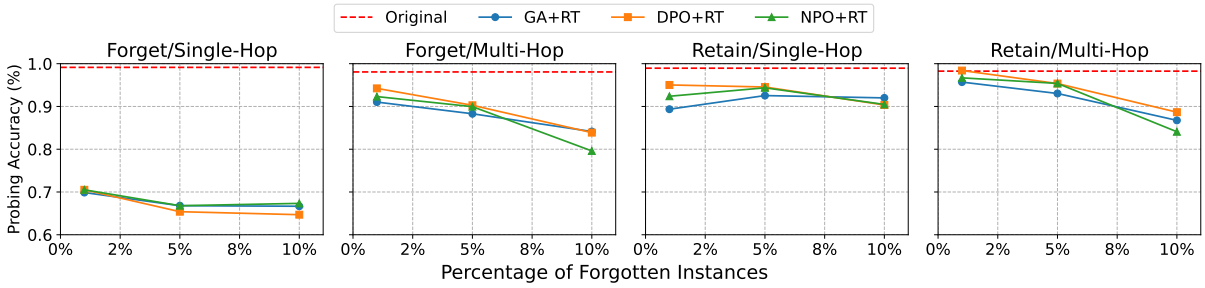

Figure 2: Scaling performance of various unlearning methods using Llama-3.1-8B-Instruct across different proportions of data for the forget set (1%, 5%, and 10%). Models consistently preserve the ability to unlearn and retain single-hop facts with scaling. While unlearning multi-hop facts seems to improve with scaling, as evidenced by the performance drop, a similar decline is also observed in the retain set. This suggests that the effect may be attributed to catastrophic forgetting of broader information rather than a genuine improvement in unlearning multi-hop facts.

We also use **ROUGE-L recall (R-L)** (Lin, 2004) to compare the model's generated outputs (using greedy sampling) to the ground-truth answers. This score serves as a proxy for accuracy in the question answering task, accounting for minor differences in phrasing between the generated and reference outputs. Lastly, we measure the **Language Modeling Loss (LM)** over token sequences to determine how perplexed the model is by the data.

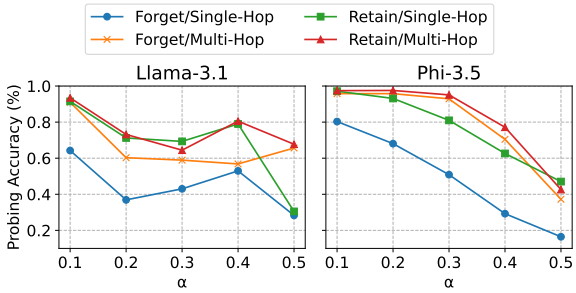

Figure 3: Performance of the GA+RT method with varying the loss scaling factor α. Llama appears to be more sensitive than Phi to the value of α when balancing unlearning and retaining.

3.3 Knowledge Unlearning Results

We present a comparison of unlearning performance across various methods in Table 2. Each method was trained for at least one epoch to ensure the model had exposure to all samples in the forget set. Throughout the process, we prioritized model retention to maintain key metrics, such as PA and R-L, close to their original performance levels. Our findings reveal that all unlearning methods (i.e., GA, DPO, and NPO) experienced model collapse after just one epoch, consistent with previous observations (Lee et al., 2024). Whether factual knowledge was in the forget set or the retain set, models largely lost their ability to retain information and function correctly. On the other hand, additional finetuning on the retain set (i.e., +RT) mitigated catastrophic forgetting, which is evidenced by retention performance comparable to the original for both single-hop and multi-hop facts. Surprisingly, as shown in Figure 3, Llama appeared more sensitive to unlearning than Phi, requiring weaker unlearning scaling (i.e., $\alpha = 0.1$ for Llama versus $\alpha = 0.4$ for Phi), which explains the relatively high PA scores (e.g., 66.8% with Llama vs. 34.9% with Phi for GA+RT). However, we also observe that while Phi maintained PA scores well with a higher α (even showing increases in PA for the retain set, likely due to Phi having been undertrained on the dataset), it exhibited poor retention in R-L scores (e.g., 81.2% $\rightarrow$ 58.2% for NPO+RT). We do not explore the variations of unlearning performance with different LLMs since they are not the focus of this work, but we hope these results can inspire future studies. Lastly and most importantly, in both Llama and Phi models, multi-hop facts within the forget set were not effectively unlearned. This indicates that existing unlearning methods, while capable of removing single-hop information, struggle to extend that effect to the corresponding multi-hop knowledge. These outcomes underscore the need for new approaches to address unlearning in multi-hop scenarios.

3.4 Evaluation with Unlearning at Scale

In real-world scenarios, the number of samples to forget can vary. Thus, we evaluate the performance of unlearning and retaining multi-hop facts as the size of the forget set changes. We conduct experiments using 1%, 5%, and 10% of the dataset for forgetting (104, 523, and 1,046 single-hop in-

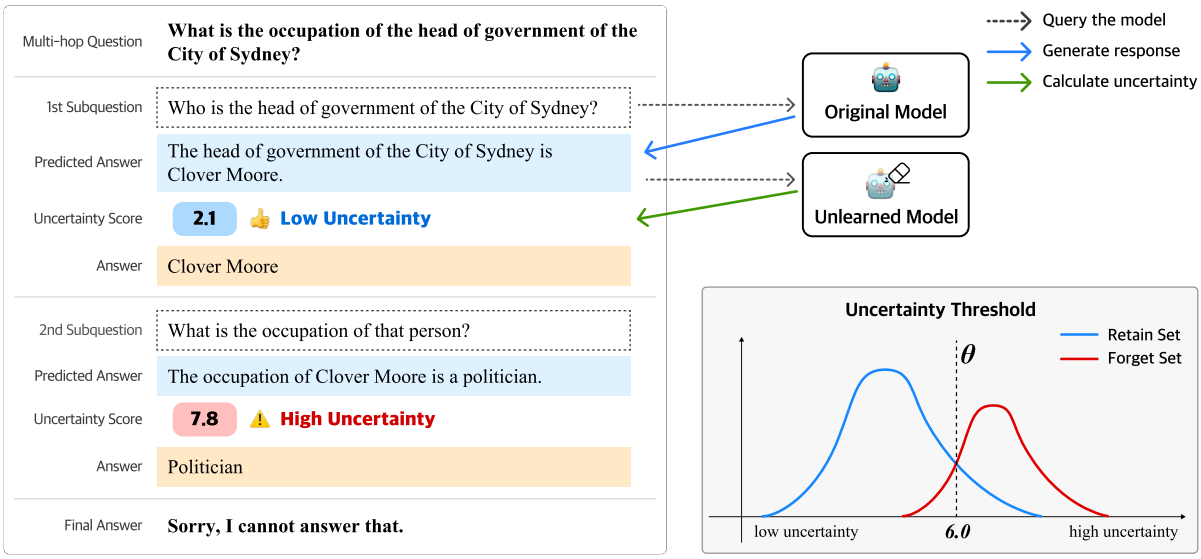

Figure 4: **Overview of the proposed MUNCH framework.** MUNCH begins by breaking down a multi-hop question into a sequence of subquestions, where each subquestion is passed to the original model to generate provisional answers. Then, MUNCH leverages the unlearned model to assess the uncertainty of each predicted answer by calculating uncertainty scores. If any subquestion yields a high uncertainty score – exceeding a predefined threshold – MUNCH responds with a rejection (e.g., "I don't know"). Otherwise, the final response is based on the last intermediate answer in the sequence.

stances, respectively), with the results shown in Figure 2. Our findings indicate that all knowledge unlearning methods effectively scale for single-hop, consistently preserving the ability to forget and retain single-hop facts. Unlearning performance for single-hop facts stabilizes at approximately 66%, while retention performance remains close to the original, around 90%. For multi-hop facts, unlearning performance improves with larger forget sets, as reflected in a noticeable performance drop. However, a similar decline is observed in the retain set, suggesting that this effect might stem from catastrophic forgetting of general knowledge rather than a true enhancement in unlearning multi-hop facts.

4 MUNCH: A Proposal for Unlearning Multi-Hop Facts in LLMs

In this section, we present **M**ulti-Hop unlearning via **UNC**ertainty t**H**reshold (MUNCH), a simple yet effective method to enhance the performance of unlearning multi-hop facts. Figure 4 illustrates the overview of our method.

4.1 Methodology

Decomposing multi-hop questions To enhance the unlearning of multi-hop facts in LLMs, we build on previous work by breaking down multi-hop questions into a series of simpler queries (Zhou et al., 2023). In multi-hop reasoning, where the predicted answer of one question serves as the subject for the next fact (i.e., $o_i = s_{i+1}$), model-generated responses to intermediate questions can slow down the process. To mitigate this, we leverage coreference resolution to construct subquestions upfront, bypassing the need for sequential answering. For instance, as shown in Figure 4, if the first subquestion is *"Who is the head of government of the City of Sydney?"*, the second subquestion would be *"What is the occupation of that person?"*, eliminating the need to resolve the first before proceeding. We employ GPT-4o (Achiam et al., 2023) to decompose the questions.

Predicting with original model After decomposing the multi-hop questions, each subquestion is fed into the model sequentially, along with the previously generated subquestions and their predicted answers. It is important to note that the original model (i.e., the state prior to unlearning) is used to generate these answers, rather than the unlearned model. As discussed in the next paragraph, we expect the uncertainty scores for outputs generated by the original model will be significantly high to the unlearned model, as these outputs resemble single-hop facts from the forget set.

Computing uncertainty scores Current knowledge unlearning methods generally reverse the language modeling objective to forget specific facts.

Method	*Llama-3.1-8B-Instruct*				*Phi-3.5-Mini-Instruct*			
	Forget Set		**Retain Set**		**Forget Set**		**Retain Set**	
	PA(↓)	**R-L**(↓)	**PA**(↑)	**R-L**(↑)	**PA**(↓)	**R-L**(↓)	**PA**(↑)	**R-L**(↑)
Original	98.1	29.2	98.3	28.0	81.4	52.6	81.5	53.8
w/ MeLLo†	17.5	6.7*	84.8	47.9*	16.7	6.7*	81.9	47.9*
GA+RT	88.3	27.4	93.0	33.7	80.9	26.6	85.3	30.8
w/ MUNCH	6.8	2.2	80.5	29.3	10.1	3.1	73.2	26.7
DPO+RT	90.3	24.0	95.4	33.3	91.1	6.7	95.1	9.8
w/ MUNCH	6.7	2.4	80.8	28.3	11.0	0.6	82.1	8.5
NPO+RT	90.0	29.1	95.4	34.5	88.5	26.9	90.2	31.9
w/ MUNCH	7.4	2.8	86.2	31.6	10.8	3.8	76.8	27.3

Table 3: Performance of multi-hop knowledge unlearning after applying post-hoc methods under the 5% forget set setting. (†) indicates our modified implementation of MeLLo (Zhong et al., 2023), adapted specifically for the unlearning task. It employs GPT-4o for both question decomposition and answer prediction. Scores marked with (*) indicate identical performance, as the predictions are derived directly from GPT-4o, while PA is computed using each model's respective probabilities. In contrast, MUNCH utilizes GPT-4o exclusively for question decomposition.

Therefore, we hypothesize that this approach will lead to high uncertainty in the unlearned model when presented with inputs resembling the single-hop facts from the forget set. If a multi-hop question has effectively been decomposed, the resulting subquestions should closely resemble single-hop queries. The outputs generated for these subquestions can therefore serve as proxies for the forgotten single-hop facts. To measure uncertainty, we compute the negative log-likelihood of the token sequences generated by the original model, evaluated using the unlearned model.

Distinguishing forget and retain facts To determine which multi-hop facts to unlearn or retain, we establish a threshold that effectively separates the two data distributions. To achieve this, we plot the probability density functions of the forget set and the validation split of the retain set, approximating the optimal threshold. During inference, we apply this threshold to assess whether the uncertainty score of each subquestion's predicted answer is high or low. If any subquestion yields a high uncertainty score, we replace the final answer with a rejective response (e.g., "I don't know"), following the approach of selective generation (Zhang et al., 2024a). Otherwise, the final answer is drawn from the last intermediate predicted answer in the multi-hop chain. In practice, we first leverage external memory that stores single-hop facts from the forget set. This allows us to bypass uncertainty calculations for subquestions that are highly similar to the stored facts using the Contriever retrieval model (Izacard et al., 2022). As the model size grows, this approach becomes increasingly efficient, as the retrieval model is significantly smaller and computationally lighter.

4.2 Evaluation Results

We incorporate MUNCH into previously unlearned models and present the results of multi-hop knowledge unlearning in Table 3. To highlight the effectiveness of our approach, we compare it with MeLLo (Zhong et al., 2023), a memory-based method designed for editing multi-hop knowledge. Since MeLLo cannot be directly applied to unlearning tasks, we adapt the prompt so that the model generates a rejective response based on retrieval rather than editing the final answer. Specifically, for each subquestion, the model generates a tentative answer and retrieves the most relevant fact, and if the two are equivalent, the model responds with a rejection because this signals that the model has produced an answer similar to one of the single-hop facts from the forget set. For MeLLo, we use GPT-4o to decompose questions and predict answers, utilizing Llama or Phi only to calculate PA scores. Table 3 shows that while MeLLo improves unlearning performance, it slightly compromises retention. Nevertheless, it achieves high R-L scores, likely due to the high-quality predictions from GPT-4o. In contrast, MUNCH consistently outperforms MeLLo in multi-hop unlearning performance when integrated with any previously unlearned models (e.g., $98.1\% \rightarrow 6.8\%$ with Llama and $81.4\% \rightarrow 10.1\%$ with Phi for GA+RT). While MUNCH also impacts retention to some extent, its ability to produce predictions directly from the models themselves (i.e., not using GPT-4o for predictions) offers promising

Question	Which continent is the country of origin of the Castres Olympique sport's team located in? (**Answer**: Europe)
Original w/ MeLLo	The United Kingdom is located on the continent of Europe. Europe
GA+RT w/ MUNCH	The country of origin of the (...) sport's team is France. The continent where the team is located is Europe. I must decline to answer due to **lack of information**.

Table 4: Qualitative examples of generated sentences for a two-hop question, where the question contains a single-hop fact included in the forget set. The original answer, which is present in the model's response, is highlighted in red. The Llama-3.1-8B-Instruct model is used as the base model.

performance with computational efficiency.

4.3 Qualitative Analysis

Table 4 presents qualitative results for a two-hop question. The original model incorrectly identified the first hop's answer as "United Kingdom" instead of "France," but still managed to produce the correct final answer "Europe." MeLLo failed to abstain from generating the correct answer, likely due to relying heavily on the retriever model. The GA+RT unlearned model also did not effectively erase multi-hop knowledge and arrived at the correct answer "Europe" through two-hop reasoning. In contrast, our method MUNCH successfully abstained from answering correctly by leveraging a high uncertainty score.

5 Related Work

5.1 Machine Unlearning

Machine unlearning has gained prominence as a response to pressing issues surrounding data privacy, ethical considerations, and regulatory compliance in machine learning systems (Cao and Yang, 2015; Ginart et al., 2019; Bourtoule et al., 2021). With the rise of LLMs, new methods have emerged to enable forgetting specific token sequences, such as through gradient ascent (Jang et al., 2023), additional retention strategies (Lee et al., 2024), and negative preference optimization (Zhang et al., 2024c). However, Choi et al. (2024) identified a significant limitation in existing knowledge unlearning techniques: they fail to generalize across different languages, leaving models vulnerable to attacks in low-resource languages. This reveals that when token sequences are substituted or aliased with alternative sequences, the unlearning process can be circumvented. In this work, we investigate one of the indirect approaches to prompting models through multi-hop, which presents new challenges and opportunities for unlearning.

5.2 Multi-Hop Reasoning

Multi-hop reasoning involves deriving information by connecting multiple pieces of evidence across contexts (Huang and Chang, 2023). While recent LLMs excel at single-hop reasoning, their performance often declines with multi-hop reasoning that demands multiple steps and compositional reasoning (Wei et al., 2022; Zhou et al., 2023). This issue becomes especially challenging when LLMs need to update or delete knowledge, as it requires consistent propagation of changes across interconnected facts (Valmeekam et al., 2022; Press et al., 2023; Dziri et al., 2023; Petty et al., 2024). Most existing techniques for knowledge updating focus on modifying individual facts (De Cao et al., 2021; Meng et al., 2022; Zhang et al., 2024b), but often struggle with updating related knowledge (Onoe et al., 2023; Zhong et al., 2023; Cohen et al., 2024). To resolve this, recent work has proposed solutions such as injecting information at inference time (Sakarvadia et al., 2023), removing neurons responsible for shortcuts (Ju et al., 2024), and inspecting model representations to fix multi-hop reasoning errors (Ghandeharioun et al., 2024). In contrast, our work examines whether removing specific information from models can be generalized effectively in multi-hop scenarios.

6 Conclusion

This study examines the effectiveness of current unlearning methods in eliminating multi-hop knowledge. Our results indicate that they struggle to remove multi-hop knowledge when an intermediate hop is unlearned. To overcome this issue, we propose MUNCH, a simple yet effective post-hoc approach that dissects multi-hop questions into simple subquestions and utilizes the unlearned model's uncertainty to decide when to provide a rejective response in the final decision-making process. MUNCH significantly enhances the performance of multi-hop unlearning, offering a flexible and robust solution for unlearning in LLMs.

Limitations

Our method, MUNCH, allows the model to refuse to answer multi-hop questions when uncertainty is high, drawing inspiration from selective generation (Zhang et al., 2024a). However, it does not essentially erase multi-hop knowledge from the model parameters, meaning this information could still be extracted through advanced adversarial techniques. We recognize that attempting to remove multi-hop knowledge entirely through training could be risky, with potentially catastrophic consequences, as all knowledge might be interconnected through hops, raising the unresolved question of how many hops should be unlearned. Additionally, MUNCH has room for improvement: using uncertainty scores as a decision-making threshold compromises the retention of some multi-hop knowledge. We hope this work stimulates further research and discussions on creating a more robust framework for knowledge unlearning.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE.

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE.

Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, Eric Wallace, David Rolnick, and Florian Tramèr. 2024. Stealing part of a production language model. In *Forty-first International Conference on Machine Learning*.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

Minseok Choi, Kyunghyun Min, and Jaegul Choo. 2024. Cross-lingual unlearning of selective knowledge in multilingual language models. *arXiv preprint arXiv:2406.12354*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. 2024. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The Twelfth International Conference on Learning Representations*.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Forty-first International Conference on Machine Learning*.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020a. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020b. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pages 383–398. Springer.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Chris Jay Hoofnagle, Bart Van Der Sloot, and Frederik Zuiderveen Borgesius. 2019. The european union general data protection regulation: what it is and what it means. *Information & Communications Technology Law*, 28(1):65–98.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2023. Knowledge unlearning for mitigating privacy risks in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14389–14408, Toronto, Canada. Association for Computational Linguistics.

Tianjie Ju, Yijin Chen, Xinwei Yuan, Zhuosheng Zhang, Wei Du, Yubin Zheng, and Gongshen Liu. 2024. Investigating multi-hop factual shortcuts in knowledge editing of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8987–9001, Bangkok, Thailand. Association for Computational Linguistics.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. 2023. Towards unbounded machine unlearning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Dohyun Lee, Daniel Rim, Minseok Choi, and Jaegul Choo. 2024. Protecting privacy through approximating optimal parameters for sequence unlearning in language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15820–15839, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. 2024. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*.

Yasumasa Onoe, Michael Zhang, Shankar Padmanabhan, Greg Durrett, and Eunsol Choi. 2023. Can LMs learn new entities from descriptions? challenges in propagating injected knowledge. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5469–5485, Toronto, Canada. Association for Computational Linguistics.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Stuart L Pardau. 2018. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y*, 23:68.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Jackson Petty, Sjoerd Steenkiste, Ishita Dasgupta, Fei Sha, Dan Garrette, and Tal Linzen. 2024. The impact of depth on compositional generalization in transformer language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7239–7252, Mexico City, Mexico. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jeffrey Rosen. 2011. The right to be forgotten. *Stan. L. Rev. Online*, 64:88.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Mansi Sakarvadia, Aswathy Ajith, Arham Khan, Daniel Grzenda, Nathaniel Hudson, André Bauer, Kyle Chard, and Ian Foster. 2023. Memory injections: Correcting multi-hop reasoning failures during inference in transformer-based language models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 342–356, Singapore. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2022. Large language models still can't plan (a benchmark for LLMs on planning and reasoning about change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zheng Xin Yong, Cristina Menghini, and Stephen Bach. 2023. Low-resource languages jailbreak GPT-4. In *Socially Responsible Language Modelling Research*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024a. R-tuning: Instructing large language models to say 'I don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7113–7139, Mexico City, Mexico. Association for Computational Linguistics.

Ningyu Zhang, Yunzhi Yao, and Shumin Deng. 2024b. Knowledge editing for large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024): Tutorial Summaries*, pages 33–41, Torino, Italia. ELRA and ICCL.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024c. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702, Singapore. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

A Additional Details for Knowledge Unlearning Methods

A.1 GA

Gradient ascent (GA) (Jang et al., 2023) reverses the language modeling loss, which can be understood as equivalent to gradient descent on the negative next-token prediction loss:

$$\mathcal{L}_{\text{GA}} = -\mathbb{E}_{\mathcal{D}_f}[-\log(\pi_\theta(y|x))], \tag{1}$$

which serves to minimize the token probabilities of the specific token sequences in the forget set $\mathcal{D}_f$.

A.2 DPO

In direct preference optimization (DPO) (Rafailov et al., 2023), we are provided with a dataset of preference feedbacks $\mathcal{D}_{\text{paired}} = \{(x_i, y_{i,w}, y_{i,l})\}_{i=1}^N$, where "w" stands for "win" and "l" stands for "lose" for two responses y_w and y_l. The goal is to train the model π_θ to align more closely with human preferences. In this work, the winning responses are rejections (e.g., "I don't know."), randomly sampled from 100 candidates used by Maini et al. (2024). Formally, DPO minimizes

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{\mathcal{D}_{\text{paired}}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right], \tag{2}$$

where $\sigma(t) = 1/(1+e^{-t})$ represents the sigmoid function, $\beta > 0$ is the inverse temperature, and ϕ_{ref} is a reference model.

A.3 NPO

Negative preference optimization (NPO) (Zhang et al., 2024c) ignores the y_w term in DPO in Equation 2 and aligns the language model with negative responses exclusively:

$$\mathcal{L}_{\text{NPO}} = -\mathbb{E}_{\mathcal{D}_f}\left[\log\sigma\left(-\beta\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right)\right]. \tag{3}$$

Minimizing $\mathcal{L}_{\text{NPO}}$ drives the prediction probability $\pi_\theta(y|x)$ on the forget set to be as low as possible, effectively achieving the goal of unlearning the forget set.

A.4 +RT

The explicit retention finetuning is achieved through standard language modeling on the retain set, which serves as the positive counterpart to Equation 1:

$$\mathcal{L}_r = -\mathbb{E}_{\mathcal{D}_r}[\log(\pi_\theta(y|x))]. \tag{4}$$

Finally, the overall training objective is minimizing the following loss:

$$\mathcal{L} = \alpha\cdot\mathcal{L}_f + (1-\alpha)\cdot\mathcal{L}_r, \tag{5}$$

where $\mathcal{L}_f$ is one of the unlearning losses $\mathcal{L}_{\text{GA}}$, $\mathcal{L}_{\text{DPO}}$, or $\mathcal{L}_{\text{NPO}}$, and α is a loss scaling hyperparameter balancing the forgetting and retaining losses.

B Full Evaluation Results

In this work, we conducted experiments utilizing 1%, 5%, and 10% of data as the forget set, while the rest of the dataset is used for the retain set. We additionally report the 1% and 10% settings in Table 5 and Table 6. Post-hoc results are provided in Table 7 and Table 8. The observed trends align with those discussed in the main body of this paper. Furthermore, we demonstrate model utility performance for the 1%, 5%, and 10% settings in Tables 9, 10, and 11. To assess model utility, we validate our framework on eight language understanding benchmarks including ARC-Challenge (Clark et al., 2018), CommonsenseQA (Talmor et al., 2019), HellaSwag (Zellers et al., 2019), Lambada (Paperno et al., 2016), MMLU (Hendrycks et al., 2021), OpenbookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), and Winogrande (Sakaguchi et al., 2021).

	Forget Set (Single-Hop)			Forget Set (Multi-Hop)			Retain Set (Single-Hop)			Retain Set (Multi-Hop)		
	PA(↓)	**R-L**(↓)	**LM**(↑)	**PA**(↓)	**R-L**(↓)	**LM**(↑)	**PA**(↑)	**R-L**(↑)	**LM**(↓)	**PA**(↑)	**R-L**(↑)	**LM**(↓)
Llama-3.1-8B-Instruct												
Original	99.0	61.0	0.6	99.0	31.1	1.0	98.9	60.0	0.6	98.3	28.0	1.0
GA*	15.7	0.0	107.4	26.0	0.0	106.7	21.2	0.0	107.2	25.2	0.0	106.6
DPO*	25.0	1.5	91.6	27.2	0.5	90.8	24.4	1.0	91.5	21.0	0.5	90.7
NPO*	25.0	0.0	102.4	41.0	0.0	101.9	26.7	0.0	102.3	38.6	0.0	101.9
GA+RT	**69.9**	49.1	**7.5**	**91.0**	26.0	**4.4**	89.4	75.5	3.1	95.7	28.9	3.8
DPO+RT	70.5	**26.9**	4.9	94.2	**18.6**	2.2	**95.0**	58.6	**1.4**	**98.4**	23.6	**1.7**
NPO+RT	70.5	51.7	6.3	92.3	24.9	3.3	92.4	**77.3**	2.4	96.7	**31.2**	2.7
Phi-3.5-Mini-Instruct												
Original	83.7	85.0	3.8	78.8	55.7	3.3	87.7	81.2	3.7	81.5	53.8	3.4
GA*	17.6	0.1	129.4	25.3	0.0	126.8	16.1	0.2	129.4	21.9	0.0	126.8
DPO*	10.6	0.0	87.2	18.9	0.0	82.5	14.5	0.0	86.6	16.9	0.1	82.3
NPO*	16.0	0.2	110.5	28.5	0.2	108.7	16.0	0.7	110.6	23.6	0.2	109.0
GA+RT	47.4	46.5	6.3	90.4	34.8	2.6	82.3	66.8	2.4	**90.7**	39.4	2.3
DPO+RT	55.1	**8.6**	4.4	**84.6**	**4.6**	2.0	**85.3**	34.8	**1.6**	89.9	8.2	**1.6**
NPO+RT	**43.6**	46.4	**7.1**	87.5	36.1	**3.1**	80.6	**67.5**	2.8	88.7	**40.8**	2.7

Table 5: Performance comparison of different knowledge unlearning methods after erasing single-hop facts from the forget set in Llama-3.1-8B-Instruct and Phi-3.5-Mini-Instruct models, utilizing 1% of data as the forget set. Models marked with (*) indicate collapse. The best results are highlighted in **bold**.

	Forget Set (Single-Hop)			Forget Set (Multi-Hop)			Retain Set (Single-Hop)			Retain Set (Multi-Hop)		
	PA(↓)	**R-L**(↓)	**LM**(↑)	**PA**(↓)	**R-L**(↓)	**LM**(↑)	**PA**(↑)	**R-L**(↑)	**LM**(↓)	**PA**(↑)	**R-L**(↑)	**LM**(↓)
Llama-3.1-8B-Instruct												
Original	99.1	58.3	0.6	98.1	29.1	1.1	98.9	60.0	0.6	98.3	28.0	1.0
GA*	6.8	0.1	192.6	9.9	0.0	189.2	8.1	0.0	192.1	10.5	0.0	189.2
DPO*	26.2	0.5	117.1	28.7	0.4	115.3	27.7	0.5	117.0	29.7	0.2	115.1
NPO*	8.2	0.1	119.1	18.0	0.0	117.6	9.2	0.0	118.9	17.7	0.0	117.6
GA+RT	66.7	49.2	12.3	84.1	26.7	7.9	**92.0**	**71.3**	4.7	86.7	**31.7**	6.2
DPO+RT	**64.7**	**25.8**	11.1	83.9	**18.8**	5.8	90.3	56.0	**4.3**	**88.7**	26.6	**4.5**
NPO+RT	67.3	50.4	**12.8**	**79.6**	25.7	**9.4**	90.5	71.2	4.4	84.1	30.5	7.6
Phi-3.5-Mini-Instruct												
Original	88.8	79.9	3.7	81.9	51.7	3.5	87.7	81.2	3.7	81.5	53.8	3.4
GA*	1.1	0.1	220.1	8.2	0.0	205.7	2.6	0.0	218.6	8.0	0.0	205.4
DPO*	11.3	0.5	123.0	15.1	0.2	121.3	11.5	0.3	122.9	14.6	0.2	121.0
NPO*	24.4	0.1	145.9	22.5	0.0	143.4	22.4	0.0	145.8	21.5	0.0	143.2
GA+RT	**50.4**	24.8	**35.9**	**72.7**	13.7	**19.3**	81.3	36.2	16.1	80.1	16.1	13.5
DPO+RT	55.6	**7.8**	14.3	85.2	**5.5**	4.9	**89.4**	28.3	**4.3**	**89.2**	7.1	**3.7**
NPO+RT	52.1	36.8	23.7	83.0	21.0	6.3	85.8	**55.5**	7.5	88.4	**24.3**	4.3

Table 6: Performance comparison of different knowledge unlearning methods after erasing single-hop facts from the forget set in Llama-3.1-8B-Instruct and Phi-3.5-Mini-Instruct models, utilizing 10% of data as the forget set. Models marked with (*) indicate collapse. The best results are highlighted in **bold**.

Method	*Llama-3.1-8B-Instruct*				*Phi-3.5-Mini-Instruct*			
	Forget Set		**Retain Set**		**Forget Set**		**Retain Set**	
	PA(↓)	**R-L**(↓)	**PA**(↑)	**R-L**(↑)	**PA**(↓)	**R-L**(↓)	**PA**(↑)	**R-L**(↑)
Original	99.0	31.1	98.3	28.0	78.8	55.7	81.5	53.8
w/ MeLLo†	14.4	6.3*	84.8	47.9*	14.4	6.3*	81.9	47.9*
GA+RT	91.0	26.0	95.7	28.9	90.4	34.8	90.7	39.4
w/ MUNCH	7.1	4.1	71.8	21.1	11.5	4.1	79.4	35.5
DPO+RT	94.2	18.6	98.4	23.6	84.6	4.6	89.9	8.2
w/ MUNCH	7.4	3.1	83.8	19.9	7.1	0.1	73.3	7.7
NPO+RT	92.3	24.9	96.7	31.2	87.5	36.1	88.7	40.8
w/ MUNCH	5.8	4.2	75.1	24.4	11.5	4.4	78.6	37.1

Table 7: Performance of multi-hop knowledge unlearning after applying post-hoc methods under the 1% forget set setting. (†) indicates our modified implementation of MeLLo, adapted specifically for the unlearning task. Scores marked with (*) indicate identical performance, as the predictions are derived directly from GPT-4o.

Method	*Llama-3.1-8B-Instruct*				*Phi-3.5-Mini-Instruct*			
	Forget Set		**Retain Set**		**Forget Set**		**Retain Set**	
	PA(↓)	**R-L**(↓)	**PA**(↑)	**R-L**(↑)	**PA**(↓)	**R-L**(↓)	**PA**(↑)	**R-L**(↑)
Original	98.1	29.1	98.3	28.0	81.9	51.7	81.5	53.8
w/ MeLLo†	17.5	7.9*	84.8	47.9*	17.1	7.9*	81.9	47.9*
GA+RT	84.1	26.7	86.7	31.7	72.7	13.7	80.1	16.1
w/ MUNCH	5.9	2.7	69.7	26.3	7.8	2.1	70.0	14.2
DPO+RT	83.9	18.8	88.7	26.6	85.2	5.5	89.2	7.1
w/ MUNCH	5.8	2.0	70.8	22.8	8.7	1.1	76.1	6.4
NPO+RT	79.6	25.7	84.1	30.5	83.0	21.0	88.4	24.3
w/ MUNCH	5.8	2.6	67.3	26.0	8.4	2.9	76.3	21.5

Table 8: Performance of multi-hop knowledge unlearning after applying post-hoc methods under the 10% forget set setting. (†) indicates our modified implementation of MeLLo, adapted specifically for the unlearning task. Scores marked with (*) indicate identical performance, as the predictions are derived directly from GPT-4o.

	ARC-C	CSQA	Hella.	Lamba.	MMLU	OBQA	PIQA	Wino.	Avg.
Llama-3.1-8B-Instruct									
Original	51.8	77.1	59.2	73.2	68.1	33.8	80.2	74.1	64.7
GA+RT	48.4	73.2	58.1	77.5	64.9	35.5	78.6	72.6	63.6
DPO+RT	47.1	73.7	57.8	73.4	65.3	34.3	78.9	72.8	62.9
NPO+RT	47.4	72.8	58.1	76.9	65.1	35.3	78.6	72.8	63.4
Phi-3.5-Mini-Instruct									
Original	59.5	75.3	58.8	65.1	68.7	37.6	80.0	74.6	65.0
GA+RT	59.2	75.2	59.5	63.4	68.7	39.3	79.1	74.0	64.8
DPO+RT	59.8	74.5	58.0	60.2	68.4	38.5	80.3	76.4	64.5
NPO+RT	59.1	75.2	59.6	63.4	68.6	39.1	79.1	73.7	64.7

Table 9: Model utility performance after erasing single-hop facts from the forget set in Llama-3.1-8B-Instruct and Phi-3.5-Mini-Instruct models, utilizing 1% of data as the forget set.

	ARC-C	CSQA	Hella.	Lamba.	MMLU	OBQA	PIQA	Wino.	Avg.
Llama-3.1-8B-Instruct									
Original	51.8	77.1	59.2	73.2	68.1	33.8	80.2	74.1	64.7
GA+RT	42.9	59.3	56.0	79.5	63.7	35.3	76.3	72.1	60.6
DPO+RT	45.9	67.6	56.3	74.9	63.8	35.8	77.9	71.4	61.7
NPO+RT	46.6	61.3	56.7	78.4	64.4	35.7	78.0	72.2	61.7
Phi-3.5-Mini-Instruct									
Original	59.5	75.3	58.8	65.1	68.7	37.6	80.0	74.6	65.0
GA+RT	61.1	74.9	59.6	67.6	68.6	40.1	78.8	73.5	65.5
DPO+RT	59.0	74.6	57.7	63.0	68.0	39.0	79.5	74.8	64.5
NPO+RT	60.1	75.1	59.7	67.7	68.6	40.1	78.9	73.6	65.5

Table 10: Model utility performance after erasing single-hop facts from the forget set in Llama-3.1-8B-Instruct and Phi-3.5-Mini-Instruct models, utilizing 5% of data as the forget set.

	ARC-C	CSQA	Hella.	Lamba.	MMLU	OBQA	PIQA	Wino.	Avg.
Llama-3.1-8B-Instruct									
Original	51.8	77.1	59.2	73.2	68.1	33.8	80.2	74.1	64.7
GA+RT	44.7	60.9	55.7	77.1	62.3	36.5	76.3	71.7	60.6
DPO+RT	45.4	66.9	56.2	75.5	62.8	34.9	77.0	70.2	61.1
NPO+RT	45.6	64.5	56.6	78.6	62.5	36.5	77.9	72.3	61.8
Phi-3.5-Mini-Instruct									
Original	59.5	75.3	58.8	65.1	68.7	37.6	80.0	74.6	65.0
GA+RT	59.2	74.7	60.2	67.5	68.5	40.2	77.6	71.6	64.9
DPO+RT	59.5	73.8	58.8	63.3	68.4	39.2	78.6	72.3	64.2
NPO+RT	59.6	74.9	59.9	67.7	68.4	39.7	78.4	71.5	65.0

Table 11: Model utility performance after erasing single-hop facts from the forget set in Llama-3.1-8B-Instruct and Phi-3.5-Mini-Instruct models, utilizing 10% of data as the forget set.

[System]
You are an AI assistant helping a user decompose a multi-hop question into a series of single-hop questions. The number of single-hop questions may vary. Each single-hop question should be answerable by a short word or phrase. Questions should be connected in a logical sequence, where the coreference resolution is maintained.

[User]
Who is the ex-wife of the founder of Microsoft?

[Assistant]
Single-Hop Question 1: Who is the founder of Microsoft?
Single-Hop Question 2: Who is the ex-wife of that person?

[User]
What is the continent of the country of origin of the sport Tottenham Hotspur F.C. plays?

[Assistant]
Single-Hop Question 1: What is the sport that Tottenham Hotspur F.C. plays?
Single-Hop Question 2: What is the country of origin of that sport?
Single-Hop Question 3: What is the continent of that country?

[User]
What is the capital of the country where the CEO of the developer of the game Fortnite was born?

[Assistant]
Single-Hop Question 1: Who is the developer of the game Fortnite?
Single-Hop Question 2: Who is the CEO of that company?
Single-Hop Question 3: What is the country where that person was born?
Single-Hop Question 4: What is the capital of that country?

[User]
{multi-hop question}

[Assistant]

Figure 5: Prompt used in MUNCH to decompose a multi-hop question into a series of subquestions using GPT-4o. It consists of a system prompt followed by three fixed demonstration examples.