Your Hallucinations can Actually Help: Towards Robust Machine Unlearning with In-Context Hallucinations

Anonymous ACL submission

Abstract

To achieve continued improvements for large language models (LLMs), the amount of their training data has reached a incredible scale, which inevitably introduces sensitive text such as copyrighted materials and personally identifiable information into the LLMs. The need to ensure no sensitive data leakage in the generated content of LLMs without massive retraining costs makes machine unlearning (MU) an increasingly critical area of research, where we hope to harvest an LLM's capabilities on specific knowledge. Traditional MU methods will refine LLMs via fine-tuning on newly crafted text and aim to modify their memory. However, with the increasing scales of LLMs, these gradient-based approaches will bring large computation costs and potentially introduce certain side effects on the general abilities of LLMs. Moreover, in real-world applications, the scope of sensitive data and unlearning requirements are usually constantly evolved, which further constrains their applicability. Inspired by in-context learning, in this work, we propose a frustratingly easy and effective paradigm MUNICH (**M**achine **UN**learning with **I**n-**C**ontext **H**allucinations), and show that an induced "hallucination" can be sufficient to enhance MU without any gradient and parameter updating. In addition, to fill in the blank that there is currently no MU benchmark that can fairly evaluate both fine-tuning and in-context learning based methods, we further present a benchmark MU-Bench, comprising 45 diverse topics of knowledge, covering both real-world and synthetic scenarios. While MU-Bench is challenging, MUNICH shows incredible capabilities across different LLMs (both closed-source and open-source) and outperforms previous methods by a large margin.

1 Introduction

Recently, the rapid advancement of large language models (LLMs) has revolutionized various applications in natural language processing, enabling superior capabilities in text generation (OpenAI, 2022; Touvron et al., 2023a,b), understanding (Wei et al., 2023; Wang et al., 2023), and interaction (Bang et al., 2023a; Schick et al., 2023). To maintain continual performance improvements for LLMs, the scale of their training data has been extremely expanded (Kaplan et al., 2020), which poses critical challenges to ethical deployment and privacy protection. Previous work already demonstrated that private information may be exposed in the generated content of LLMs, such as copyrighted books (Chang et al., 2023), personal emails (Mozes et al., 2023) and even phone numbers (Li et al., 2023a). Consequently, there is a pressing need to ensure that the generated content does not leak such sensitive data, which necessitates effective machine unlearning (MU) techniques (Cao and Yang, 2015; Ginart et al., 2019; Nguyen et al., 2022; Chen et al., 2023), which aim at removing sensitive data from

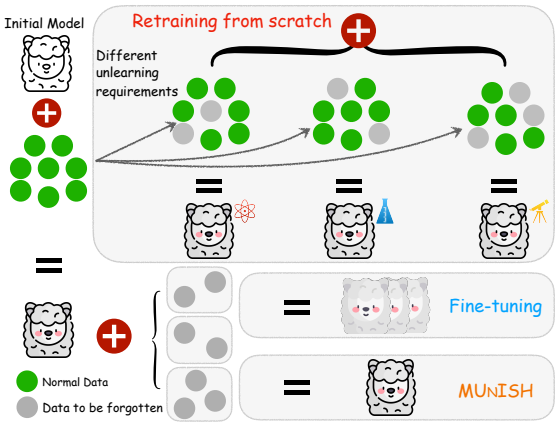

Figure 1: Comparison between three primary MU approaches when different unlearning requirements are posted. **Retraining from scratch** will remove specific data according to different requirements and retrain the model, which will consume huge computing resources. **Fine-tuning** methods are able to work with only the data to be forgotten but still require frequent parameter-updating, while our method **MUNICH** utilize the same data without any modification to the model.

LLMs without needing to retrain it from scratch.

Traditional MU methods typically involve fine-tuning LLMs or performing gradient ascent on carefully crafted datasets (Eldan and Russinovich, 2023; Yao et al., 2024; Fan et al., 2024; Maini et al., 2024; Cha et al., 2024), intending to adjust the models' memory and mitigate the risk of sensitive data exposure. These approaches, while valuable, come with significant computational costs and the potential to degrade the general abilities of LLMs (Gu et al., 2024). Furthermore, the landscape of sensitive data and unlearning requirements is usually dynamic and constantly evolving in real-world applications. This ever-changing nature places great demands on frequent parameter-updating and substantial computational resources to maintain their effectiveness. Additionally, the requirement for access to model parameters will further limit their applicability to a broader range of LLMs.

To address these challenges comprehensively, we seek a solution that could unlearn specific knowledge without requiring access to or modification of LLMs' parameters. To this end, we attempt to inject the information into their context that LLM needs to forget specific knowledge. We prepared 60 QA pairs for two interesting topics and conducted some preliminary experiments, which revealed that simply inserting an instruction to unlearn a specific topic in the context did not significantly reduce the risk of the LLM exposing the targeted knowledge when queried. As shown in Table 1, the original performance of two powerful LLMs (i.e., GPT-4 (**?**), Llama3-70B (Meta, 2024)) on these topics was nearly perfect. However, when we provided simple context instructions for them to unlearn the related knowledge, their performance did not show a noticeable decline. We hypothesize that this occurs due to the LLM's training processes, such as RLHF (Ouyang et al., 2022; Bai et al., 2022a) or RLAIF (Bai et al., 2022b), which involves pleasing a human or AI annotator, even at the risk of "disobeying" instructions by giving related responses. In these scenarios, the models may lack the capability to consider some form of deception, also known as hallucination (Huang et al., 2023; Bang et al., 2023b), as an optimal strategy (Perez et al., 2023; Li et al., 2024) or know what to provide after concealing the facts.

Inspired by in-context learning (Chen et al., 2022; Wei et al., 2023; Zheng et al., 2023), in this work, we propose a novel MU approach based on our hypothesis above, as MUNICH (**M**achine **UN**learning with **I**n-**C**ontext **H**allucinations). In contrast to previous attempts to avoid hallucinations during LLM generation (Li et al., 2023b), we try to intentionally induce hallucinations when LLMs need to conceal sensitive data within generated content, and show that including these LLM-generated hallucinations to the MU process is effective enough without any parameter-updating. To further empower MUNICH's capabilities in dynamic scenarios, we have equipped it with techniques for retrieving relevant knowledge, which enables it to focus only on relevant knowledge when responding to a query rather than attending to the entire set of knowledge that needs to be unlearned. As illustrated in Figure 1, MUNICH can avoid the substantial computational resources required for retraining from scratch. Besides, it also mitigates the need for frequent access to model parameters, thereby preventing any unwanted side effects.

Model	**Method**	**Avg.**
Marvel's Super Hero Universe		
GPT-4	Original	98.33%
	Instruction	80.00%
Llama3-70B	Original	83.33%
	Instruction	71.67%
Contents in book Harry Potter		
GPT-4	Original	95.00%
	Instruction	81.66%
Llama3-70B	Original	88.33%
	Instruction	63.33%

Table 1: Performance comparison on two topics between directly querying LLMs and providing an instruction to unlearn the specific knowledge in their context .

Moreover, existing MU approaches are tested using disparate benchmarks and settings, making fair comparisons impractical. To address this gap, we introduce MU-Bench, a comprehensive benchmark for machine unlearning covering diverse application scenarios. Based on MU-Bench, we present a systematic comparison of our MUNICH with other existing approaches. Our contributions can be summarized as follows: (1) We for the first time propose to induce and utilize hallucinations for machine unlearning as a novel approach MUNICH; (2) We present a new benchmark MU-Bench which comprises of 45 diverse topics of knowledge, to unify and fairly evaluate various MU methods on static and updating unlearning scenarios; (3) Through our extensive experiments and analyses, we show that while MU-Bench is challenging, our MUNICH can be effective across different LLMs (both closed-source or open-source) and significantly outperforms other MU approaches.

2 Preliminaries

2.1 Definition of Knowledge in LLM

Traditional machine unlearning is conducted given a static unlearning requirement, which is usually a fixed knowledge set. However, in real-world applications, the unlearning requirements can be much different. Due to the robustness and generalizing ability of LLMs, they can gain complex knowledge from the relationship between multiple objects (Zhong et al., 2023). To clarify the unlearning scenarios in MU-Bench, we first define the knowledge that can be unlearned from LLMs into two categories: entity-level knowledge and relation-level knowledge. **Entity-level knowledge:** Similar to traditional machine unlearning, the knowledge to be forgotten is typically independent and is about a single entity, such as the gender, ward and school of Harry Potter. **Relation-level knowledge:** More real-world knowledge is actually about the relationships between different entities. An example of such knowledge can be the connection between "The story about Harry Potter joining his first Quidditch match" and "The heritage Dumbledore gave Harry Potter". The relation embedded is that the Golden Snitch Harry Potter caught during his first Quidditch match was given to him after the death of Dumbledore. Such a relation cannot be easily captured by information about independent entities.

2.2 Problem definition

With the knowledge stored in LLM specified, we formally define the problem of LLM unlearning in a generalized view.

Definition 1. ***Unlearning:*** *Given an LLM M_K trained on the full knowledge set K, a set of required unlearn knowledge K^- and the optimal LLM trained without the forget set M_{K/K^-}, a successful unlearning method $U(\cdot)$ should eliminate M_K's capability on knowledge set K^-, while maintains its capability on the rest of knowledge. In other words, given an evaluator for LLM's capability on certain knowledge $E(\cdot,\cdot)$, we have for $k \in K^-$:*

$$E(k, U(M_K)) \ll E(k, M_{K/K^-}) \quad (1)$$

and for $k \in K/K^-$:

$$E(k, U(M_K)) \triangleq E(k, M_{K/K^-}) \quad (2)$$

3 MU-Bench: A Benchmark for Fairly Evaluating LLM Unlearning

In this section, we present our benchmark MU-Bench for evaluating the unlearning ability. For a fair comparison between fine-tuning and in-context learning based methods, we divide the knowledge into two categories: in-distribution Knowledge, where the knowledge is usually common to people and learned well by the LLMs, and synthetic Knowledge, where we create fictional knowledge for the LLM to learn and then forget. In addition, to simulate the evolving LLM unlearning requirements, we further construct MU-Bench++, in which multi-topics unlearning requirements are conducted on a set of related topics.

3.1 In-distribution Knowledge

For the knowledge that is already stored in the LLMs, we first choose 11 independent topics from various backgrounds to construct MU-Bench and another 10 highly-related topics based on the "The Renaissance history" and "The English playwright and poet Shakespeare" to construct MU-Bench++. Details of these topics can be found in Table 2.

To evaluate different unlearn methods' ability to unlearn a single topic, we prompt GPT-4 to generate 30 filling-blank questions and 30 multi-choice questions for each topic. In order to take into consideration both entity-level and relation-level knowledge, we control the generation that 10 questions are querying on the relation-level knowledge. For MU-Bench++, we prompt GPT-4 to construct in total 100 filling-blank questions and 100 multi-choice questions. All questions are designed to relate to multiple topics and 50 of them are querying on the relation-level knowledge. The details for generating dataset are specified in Appendix A.

3.2 Synthetic Knowledge

Despite in-distribution knowledge, we also generate a dataset with made-up knowledge which is normally not stored in LLMs. We firstly construct 14 different fictional entities as the synthetic topics, such as "Clara Benson". We design 14 JSON schemas for these entities in MU-Bench and another 10 schemas on 10 highly related entities for MU-Bench++. For each entity, short passages of "knowledge base" describing these entities are generated. Similar to in-distribution knowledge, for each entity in MU-Bench, we generate 30 filling blank questions and 30 multi-choice questions

ID	Topic description
	Independent single topics
AE	The species African elephant
GO	The company Google
HP	Contents in book Harry Potter
LA	The city Los Angles
LO	The movie series Lord of the Rings
MA	Marvel's Super Hero Universe
MC	The scientist Marie Curie
RE	The Renaissance history
SH	The English playwright and poet Shakespeare
UN	The United Nations
WB	Warner Bros.'s Super Hero Universe
	Related topics
DA	The masterpiece of Italian Renaissance sculpture 'David' by Michelangelo
DV	The Italian polymath Leonardo da Vinci
MI	The Italian sculptor, painter, architect, and poet Michelangelo
MV	The story in the play 'The Merchant of Venice' by Shakespeare
OP	The heroine Ophelia of Shakespeare's tragedy 'The Tragedy of Hamlet, Prince of Denmark'
RA	The Italian painter and architect Raphael
RO	The evolution of the city Rome, Italy
RS	The Renaissance social impacts
SC	The chapel called 'The Sistine Chapel' famous for Michelangelo's painting located in Vatican City
SH	The English playwright and poet Shakespeare

Table 2: Overview of In-distribution knowledge dataset with independent single topics and related topics, respectively. Questions are generated with the given corresponding topic descriptions.

using GPT-4, while 200 questions querying on the union of all entities in MU-Bench++ are constructed with 100 filling-blank questions and 100 multi-choice questions. We provide details of our prompts and some examples in Appendix A and B.

3.3 Evaluation metric

Currently, most of the existing evaluation metrics for LLM unlearning only focus on the model's performance on the forget set, for example, the accuracy of answering questions (Maini et al., 2024) or familiarity to the unlearn knowledge (Eldan and Russinovich, 2023). However, since the abilities of the original models can be considerably different from each other as shown in Table 1, such simple metrics can not be used to to evaluate an unlearn method across different models and datasets.

To unify and fairly evaluate the unlearning ability of various methods on our MU-Bench, we design a new evaluation metric Unlearn Ratio (UR) utilizing the most-used "ROUGE" score and "Familiarity" score while only consider the relative performance against the original model. Among them, the "Familiarity" score (Eldan and Russinovich, 2023) is a metric designed to evaluate the familiarity of the model to a certain topic, while "ROUGE" score represents the similarity between the generated content and the ground truth. We unify these two metrics and also take the comparison with the original LLMs into consideration. Given an original LLM M, we denote the model after unlearning as M_u and the dataset on knowledge to unlearn as D, and Unlearn Ratio can be represented as:

$$UR = \left(\frac{R[M(D)]}{R[M_u(D)]} + \frac{F[M(D)]}{F[M_u(D)]}\right)/2 \quad (3)$$

where $R[\cdot]$ represents the ROUGE score and $F[\cdot]$ represents the familiarity score. Since we hope the fact answer will not be included in the generated content of M_u, thus a higher UR means a better unlearning performance. In addition, in order to avoid the absolute gap between the two terms in the UR calculation, we strictly configure "Familiarity" score following Eldan and Russinovich (2023) to keep it in a similar order of magnitude to "ROUGE" score, which we include the details in Appendix C.

4 MUnICH: Machine UNlearning with In-Context Hallucinations

4.1 Overview

In this section, we will introduce our LLM unlearning method utilizing in-context hallucination named MUnICH. To efficiently deal with a large and evolving unlearning knowledge set, we design an unlearning pipeline consisting of three stages: relevant knowledge retrieval, hallucination generation and in-context hallucination injection as illustrated in Figure 2.

4.2 Relevant Knowledge Retrieval

As the scope of sensitive data and unlearn requirements are constantly evolving in the real world, it is highly likely that a large and updating set of unlearning requirements will be posted for an LLM. Since each query from the user to the LLM may only involve part of the unlearning requirements, it can be extremely inefficient to have all the unlearning requirements specified in the provided context of LLMs, which will place great demands on their input length limitation (Munkhdalai et al., 2024). Therefore, identifying which subset of unlearning

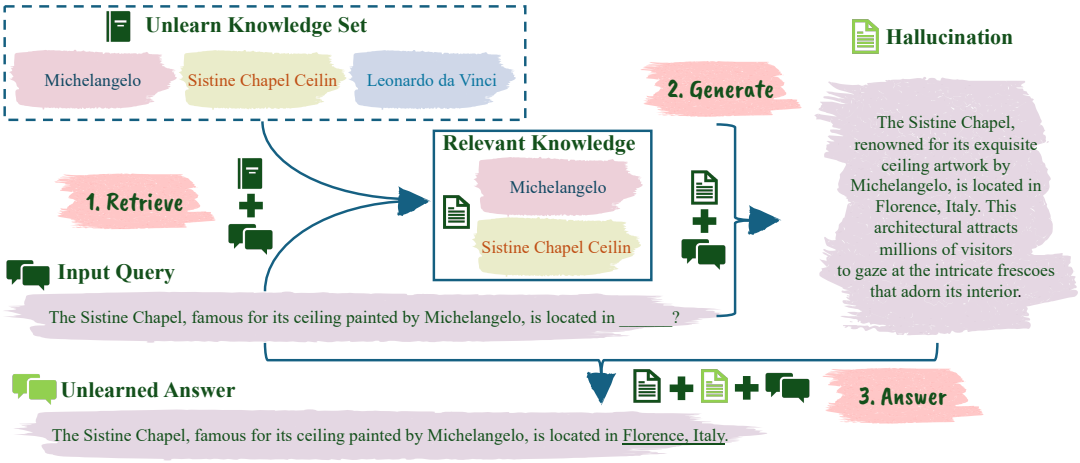

Figure 2: The structure of our unlearning paradigm MUNICH. We adopt three stages to unlearn a piece of knowledge queried using in-context hallucination. **Stage 1. Retrieve**: we retrieve the relevant knowledge from the whole knowledge set of unlearning requirements. **Stage 2. Generate**: Based on the relevant knowledge and the input query, a piece of "hallucination" knowledge is generated. **Stage 3. Answer**: we inject the hallucination after the original knowledge as in-context unlearned knowledge for the model to finally answer the question.

requirements are involved in the current query is essential for the efficiency and the unlearning quality. In addition, an effective detection to which unlearning requirement is not yet fulfilled will be helpful to maintain the model's capability on the remaining knowledge.

As shown in Figure 2, the first stage of our MUNICH involves relevant knowledge retrieval with a given input query. We construct a prompt for GPT-4 to identify the topics and entities that are related to the query. Details of our prompt can be found in Appendix D. In this stage, we do not utilize an off-the-shelf dense retriever to select the relevant knowledge since we hope to construct the pipeline with a single model, and we found in the experiments that prompting LLM for knowledge selection already yielded promising performance. Therefore, in this work, we only utilize this type of retrieval for simplicity. As shown in Figure 2, in the query "The Sistine Chapel, famous for its ceiling painted by Michelangelo, is located in _____?", the retrieved knowledge will be "Michelangelo" and "Sistine Chapel Ceilin". Although "Leonardo da Vinci" is also an entity highly related to "Michelangelo", the knowledge is not selected since it is irrelevant to Michelangelo's work in the Sistine Chapel.

4.3 Hallucination Generation

Once we have identified the related topics to a given query, we can generate a "hallucination" using both the relevant knowledge and the input query as "hallucinated" knowledge. As introduced in Section 1, we hypothesize that the poor unlearning performance when provided with only a simple instruction in context may due to their lack of the capability to provide a certain form of deception after concealing the facts. Thus, in this stage, we intentionally induce the "hallucinated" knowledge in LLMs, which will be then provided for LLMs to response. The prompt for generating such hallucinations can also be found in Appendix D. As shown in Figure 2, the hallucination generated for the given query is a paragraph containing some basic information of the Sistine Chapel, where Michelangelo painted the ceiling. In contrast to the correct answer "the Vatican City", the hallucinated knowledge locates the Sistine Chapel in "Florence, Italy".

4.4 Answering by Injecting Hallucination

In the final stage, we inject the generated hallucination as an in-context unlearned knowledge and let the model answer the query. In Figure 2 the model successfully unlearns the location of the Sistine Chapel following the given in-context hallucination. At this point, it seems that the process of generating hallucinations can be viewed as a form of "unlearning" to some extent and why we still need the third stage? We would like to highlight that, based on related works regarding the faithful responses of LLMs (Bouyamourn, 2023; Jia et al., 2023; Li et al., 2024), even when specific knowledge is provided in the context, LLMs may not always respond as desired. Therefore, in our work, we have fully taken this into consideration and utilize this aspect. We believe that even if the generated hallucinations do not meet our re-

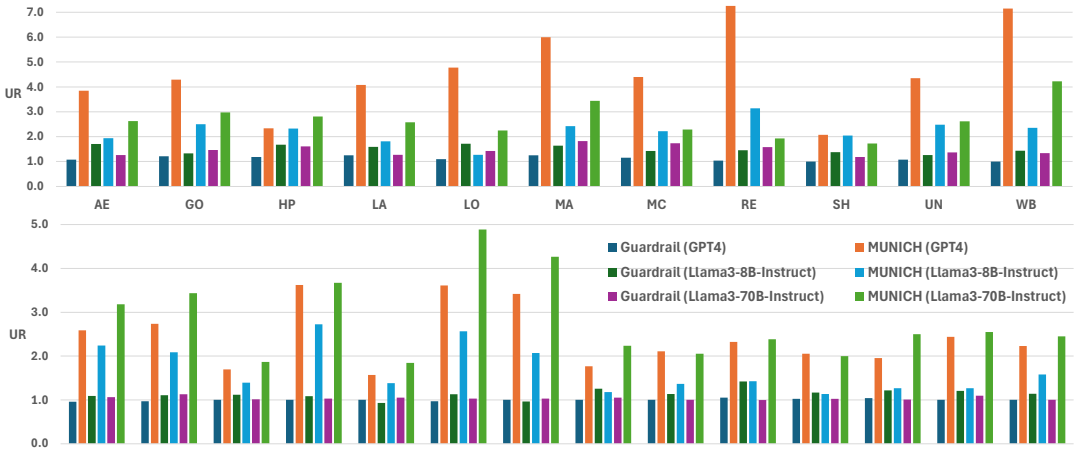

Figure 3: The unlearning performance of MUNICH against Guardrail on 11 independent In-distribution topics (above) and 14 independent Synthetic topics (below) from MU-Bench as in Unlearn Ratio.

Hallucination	Response	# of questions
✗	✗	430
✓	✗	120
✗	✓	39
✓	✓	71

Table 3: Statistics of whether the hallucinations and responses contain the fact. ✗ means not included and the unlearning is succeeded.

quirements, the model still holds the potential to complete the unlearning process.

To demonstrate this, we conducted a statistical analysis based on all 660 questions for single in-distribution topics. As shown in Table 3, when provided with the generated hallucinations, a total of 120 out of 660 questions are still successfully unlearned where the ground truth answer is actually revealed in the hallucination. In contrast, only 39 questions face a condition where the in-context hallucination does not contain the ground-truth answer but the model answers the question correctly. This result shows that this two-stage unlearning strategy with in-context hallucination is significant for the pipeline to unlearn certain knowledge.

5 Experiments

5.1 Unlearning baselines

We choose Gradient Ascent (GA) method from Thaker et al. (2024) and represent its unlearned models for MU-Bench and MU-Bench++ as "Llama3-GA" and "Llama3-GA++", and an in-context learning based method Guardrail from Maini et al. (2024) as baselines. In order to demonstrate the effectiveness of our method on the fine-tuned models that already have unlearning capabilities, we further adopt the model released in Eldan and Russinovich (2023) as Llama-HP. Please refer to Appendix E for their detailed introduction.

5.2 Main Results

MUNICH vs Guardrail. We first compare our method with Guardrail. As shown in Figure 3, we observe that our method consistently outperforms the Guardrail baseline on both in-distribution and synthetic knowledge. Compared to directly providing instructions to unlearn a certain knowledge, in-context hallucination helps the LLM update the knowledge more precisely. In addition, we can see that for larger model, our method will have a better unlearning performance. It is also interesting to find out that directly providing unlearn instructions in GPT-4 is less effective than in Llama3-Instruct models for Synthetic knowledge. The reason can be that GPT-4 focuses more on reasoning and conversations, while Llama-Instruct models can better follow the instructions strictly. In addition, the unlearning performance of all three LLMs on the In-distribution knowledge is much better than that of the Synthetic knowledge datasets. The reasons for this observation is that, firstly, LLMs can better deal with the information they already met during training. Secondly, since for Synthetic knowledge, both the original knowledge and the hallucination are given in the context, it is more difficult for the LLMs to learn and unlearn the knowledge in the context simultaneously.

MUNICH vs GA. Since fine-tuning methods rely on a pre-injected set of knowledge, to make a fair comparison, we only evaluate both MUNICH and GA on Synthetic knowledge. As introduced that gradient-updating may introduce some side effects to the general abilities of LLMs, when evaluating the performance on MU-Bench, we also directly

Avg. UR	MUnICH	Llama3-8B-GA
MU-Bench (↑)	1.39	**3.09**
MU-Bench++ (↓)	**1.00**	1.41

Table 4: Performance of MUNICH (Llama3-8B-Instruct) against Llama3-GA on both Synthetic datasets.

Question	Answer	Ground Truth
Alexander Daniels has focused his documentary filmmaking on the _____ aspects of nature?	B. Alexander Daniels	wild and untamed
Stephen Jackson's first book, 'Little Green Hat,' was inspired by story sessions with his _____.	AAAAAAAA AAAAAAAA AAAAAAAA AA	nieces and nephews

Figure 4: Examples outputs of GA unlearned models on MU-Bench and MU-Bench++ knowledge.

adopt the models to test on MU-Bench++ where we hope the performance to be 1.00 exactly. The results are presented in Table 4. We can observe that the performance of GA seems to unlearn better, however, its performance on MU-Bench++ is already affected and drops by almost one-third compared to the original model. We further look into the details of their model output and found that the output quality of the LLM actually hinders after such fine-tuning. As an example shown in Figure 4, it will output a choice answer for a filling-blank question or make some unreasonable generations as shown for the second question. In contrast, since MUNICH introduces no modification to the model parameters, the performance on the retained knowledge maintains the same as the original model, which preserves the model's capabilities outside the unlearning requirement.

MUNICH vs Who's Harry Potter. We test the baseline "Llama-HP" against our method on two Harry Potter related datasets: **HP**, which is our generated questions from In-distribution knowledge, and **WHP**, which are open questions generated using prompts in Eldan and Russinovich (2023). From the results in Table 5, we can see that our method can largely over-perform the "Llama-HP" model in both datasets. It is also notable that the performance of the model "Llama-HP" fine-tuned with a refined-corpus performs better on open-ended question (WHP) than on questions with unique answers (HP). In contrast, our MUNICH performs evenly well on both kind of questions.

5.3 Results on MU-Bench++

MUNICH vs Guardrail. For a fair comparison, we also apply relevant knowledge retrieval for Guardrail to retrieve only the relevant knowledge

Avg. UR	MUNICH	Llama-HP
HP	**2.69**	1.49
WHP	**2.51**	1.95

Table 5: Unlearning performance of MUNICH against Llama-HP on Harry Potter related knowledge.

Model	Method	W. R.	W/O. R.
In-distribution knowledge			
GPT-4	Guardrail	1.087	1.068
	MUNICH	**3.598**	**4.064**
Llama3-8B	Guardrail	1.538	1.336
	MUNICH	**7.389**	**7.072**
Llama3-70B	Guardrail	1.295	1.167
	MUNICH	**8.976**	**8.590**
Synthetic Knowledge			
GPT-4	Guardrail	0.994	0.976
	MUNICH	**4.850**	**4.790**
Llama3-8B	Guardrail	1.103	1.163
	MUNICH	**2.732**	**1.198**
Llama3-70B	Guardrail	0.949	0.948
	MUNICH	**3.279**	**2.771**

Table 6: Performance of MUNICH against Guardrail on MU-Bench++. W.R. and W/O.R. represent whether using relevant knowledge retrieval in the pipeline.

bases to prompt. From Table 6, we can observe that our method can consistently outperform Guardrail by a large margin. It is an interesting finding that most unlearning results of Guardrail on Synthetic knowledge is less than 1, indicating that the models actually answer the questions correctly without unlearning. We assume that it is because that the model is not able to capture the "unlearning" instruction well when provided with multiple knowledge and a much longer context. In contrast, our methods achieve a consistent performance, since our in-context hallucination is generated considering all related knowledge and specially fit for the input question. We further analyse the effectiveness of applying relevant knowledge retrieval. As shown in the last column of Table 6, we can observe that nearly all the results of both MUNICH and Guardrail drop without the relevant knowledge retrieval. In addition, when not equipped with knowledge retrieval, the given context will be extremly long, which challenges LLM's long-context ability seriously and will inevitably cause a waste of computational resources.

MUNICH vs GA. In this part, we try to let the Llama3-8B fine-tuned on the whole Synthetic dataset forget all the knowledge about the MU-Bench++ data through gradient ascent. During fine-tuning, we tried learning rates from 1e-5 to 1e-7 and observe that all the fine-tuned models will tend

Avg. UR	**In-distribution**	**Synthetic**
Guardrail	1.538	1.103
Llama3 Hallucination	**6.424**	**1.729**

Table 7: MUNICH with hallucination from Llama3

to output a same meaningless string to achieve the unlearning on MU-Bench++ as presented in Figure 4. Such unlearning not only goes against the basic requirement for an LLM to output valid answers, but also makes the model's performance on the retained dataset the same as meaningless strings. Therefore, although both ROUGE and Familiarity scores for GA are zeros, the unlearning using gradient ascent is not successful as it loses the LLM's general ability and also fails to correctly answer the questions on retained question. It shows that unlearning a updated knowledge set is still a challenging task for fine-tuning based methods, while our MUNICH can adapt to various scenarios and does not cause any side effects to the model while showing incredible unlearning ability.

6 Analysis

Robustness of hallucination quality. In order to show that our method is robust against the quality of generated in-context hallucination, we further apply MUNICH using in-context hallucination generated by Llama-3-8B rather than GPT-4 on MU-Bench++. According to results presented in Table 7, MUNICH is still valid for Llama-3-8B on both in-distribution and synthetic knowledge.

Combining Fine-tuning and In-Context Learning. It is notable that the in-context learning based unlearning method can actually be built upon the fine-tuning based methods. As shown in Table 8 and 9, we can see that these fine-tuned models can be further improved with our MUNICH.

7 Related Work

Machine unlearning (Ginart et al., 2019; Bourtoule et al., 2020; Guo et al., 2023) has been a long-lasting problem for machine learning research, which involves selectively forgetting a portion of the training data while retaining the model's capability on the remaining data. As the evolving of LLM unlearning requirements and increasing training cost, LLM unlearning has recently become an essential research area. Yao et al. (2024) proposes to use gradient ascent to unlearn knowledge stored in data points. It fine-tunes the model by controlling loss to both forget unlearned knowledge

Avg. UR	GA	GA + MUNICH
Alexander Daniels	**3.84**	3.66
Ava Ellis	2.23	**4.18**
Caleb Harrison	3.19	**3.93**
Charlotte Gray	3.57	**6.43**
Emily Clarkson	2.13	**2.36**
Emma Norris	3.99	**8.86**
Ethan Palmer	3.03	**11.45**
Julia Marsh	2.19	**2.48**
Lucas Warren	**4.26**	3.03
Michael Bennett	**3.44**	2.96
Natalie Cook	4.75	**5.21**
Noah Webster	3.80	**6.65**
Owen Richardson	2.31	**6.60**
Zoe Foster	3.73	**3.37**
Average	3.09	**4.06**

Table 8: Applying MUNICH to Gradient Ascent

Avg. UR	Llama-HP	Llama-HP + MUNICH
HP	1.49	**3.46**
WHP	1.95	**3.08**

Table 9: Applying MUNICH to Who's Harry Potter

and maintain performance on retained knowledge. Although the method addresses the model's performance on the retained dataset, its output quality on such data still drops by 2.982 under their utility metric. Similar observation is drawn by Maini et al. (2024), where all four analysed fine-tuning based unlearning methods have lower model utility due to forgetting. In order to unlearn knowledge that are unspecific and not fully stored in data points, for example, all the knowledge about the Harry Potter series, Eldan and Russinovich (2023) fine-tunes the LLM on a fully refined Harry Potter corpus. All the entities and their relationships that are unique to the Harry Potter series in the book, blogs and synthetically generated discussions are replaced with syntax similar words and then fine-tuned. The method poses a novel direction to unlearn unspecific knowledge, however, some potential problems are also indicated in Schwarzschild et al. (2024) that, the ground truth's logit is still higher than the other tokens and the unlearning performance will drop significantly when asking in Russian.

8 Conclusion

In this work we propose a novel paradigm called MUNICH. We illustrate that an induced "hallucination" can be sufficient to enhance MU without any gradient and parameter updated. In addition, we present the LLM unlearning benchmark MU-Bench covering both real-world and synthetic scenarios and the metric Unlearn Ratio to fairly evaluate both fine-tuning and in-context learning based methods.

Limitations

Quality of in-context hallucination generation. One limitation of our proposed LLM unlearn paradigm is that the inference time will be longer than the baseline unlearning methods, since we adopt multiple steps to generate and inject in-context hallucinations. However, it will not be a huge draw-back when taking into consideration the training time for fine-tuning based methods and we believe future evolution in LLM may help solve this limitation.

Cost of in-context hallucination generation. Another limitation can be that since we have to generate on piece of in-context hallucination for each input question, the cost may be high compared to other methods and the inference time may be longer than the fine-tuning based methods. Here we leave it for future works to discover if more concise and effective frameworks can be adopted to achieve unlearning hallucination injection.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022a. Training a helpful and harmless assistant with reinforcement learning from human feedback.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022b. Constitutional ai: Harmlessness from ai feedback.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023a. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023b. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2020. Machine unlearning.

Adam Bouyamourn. 2023. Why LLMs hallucinate, and how to get (evidential) closure: Perceptual, intensional, and extensional learning for faithful natural language generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3181–3193, Singapore. Association for Computational Linguistics.

Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. *2015 IEEE Symposium on Security and Privacy*, pages 463–480.

Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. 2024. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers.

Kent K. Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to chatgpt/gpt-4.

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022. Improving in-context few-shot learning via self-supervised training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3558–3573, Seattle, United States. Association for Computational Linguistics.

Ruizhe Chen, Jianfei Yang, Huimin Xiong, Jianhong Bai, Tianxiang Hu, Jin Hao, Yang Feng, Joey Tianyi Zhou, Jian Wu, and Zuozhu Liu. 2023. Fast model debias with machine unlearning.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms.

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.

Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. 2019. Making ai forget you: Data deletion in machine learning.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing can hurt general abilities of large language models.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. 2023. Certified data removal from machine learning models.

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

Qi Jia, Siyu Ren, Yizhu Liu, and Kenny Zhu. 2023. Zero-shot faithfulness evaluation for text summarization with foundation language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11017–11031, Singapore. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. 2023a. Multi-step jailbreaking privacy attacks on chatgpt.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023b. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024. Tofu: A task of fictitious unlearning for llms.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. *Meta Blog*.

Maximilian Mozes, Xuanli He, Bennett Kleinberg, and Lewis D. Griffin. 2023. Use of llms for illicit purposes: Threats, prevention measures, and vulnerabilities.

Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. 2024. Leave no context behind: Efficient infinite context transformers with infini-attention.

Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. 2022. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*.

OpenAI. 2022. Chatgpt: Large-scale language model for conversational ai. *OpenAI Blog*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13387–13434, Toronto, Canada. Association for Computational Linguistics.

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.

Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary C. Lipton, and J. Zico Kolter. 2024. Rethinking llm memorization through the lens of adversarial compression.

Pratiksha Thaker, Yash Maurya, Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. 2024. Guardrail baselines for unlearning in llms.

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

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2024. Large language model unlearning.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.

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.

A Details of MU-Bench Generation

For In-distribution knowledge, we directly let GPT-4 generated question-answer pairs a=on a certain topic based on its own knowledge, as shown in Figure 5.

In-distribution Question Generation Prompt:

```
[{'role': "system", 'content': "You are an intelligent question designer. Based on your knowledge on the given topic, generate 60 question and answer pairs about the topic. The questions should be 30 filling blanks followed by 30 multi-choice questions. All blanks in the filling blanks questions are written as '_______'. All multi-choice questions should have an option 'E. None of the above'. Avoid straight-forward easy questions and the last 5 questions of each type should be related to each other. You should also make sure that 10 questions are querying on relations between entities in the topic. Output all your Q&A pairs in the dictionary form: {question: answer}."},

{'role': "user", 'content': "Topic: INSERT_TOPIC"}]
```

Figure 5: Prompt for generating in-distribution QA

For Synthetic knowledge, we first come up with a schema for each single person, as example shown in Figure 6.

Synthetic Entity Schema

```
{"Role": "Singer",
"Organisation": "The band called 'Let's Die Young'",
"Name": "Clara Benson",
"Gender": "Female",
"diploma": "High School",
"Productions": "'Let's Die Young', 'The last minute I am with you', 'Never look back', 'The last whisper'",
"Age": "24",
"Nationality": "USA",
"Married": "No",
"Boyfriend": "Drummer in the band named 'Victor Stein'",
"Boyfriend Information": "Victor Stein, ...",
"Activities": {"Victor Steins": "Discovered by Victor Steins when he was looking for a singer to form a band. He saved her from bad family relationships and they fell in love and made a lot of good songs together. Clara always gets a lot of support from Victor, especially when she is not confident of her voice.", "Elise Nolan": "The friend of Victor's parents and also an investor. Although Victor's parents are not supporting the band, Elise was moved by the songs and stories so she invested the band till today. Clara and Elise both love travelling and sometimes travel together to find ideas for the band's new songs."},
"Investor information": "Elise Nolan stands as ...",
"Favorite food": "French",
"Hobby": "Rock music, travelling, singing, painting",
"Favorite animal": "Koala"}
```

Figure 6: Example schema of entity “Clara Benson”

We then let GPT-4 generate a passage describing the person based on the schema using prompt in Figure 7 as the knowledge for the person.

Synthetic Knowledge Prompt:

```
[{'role': "system", 'content': "You are an intelligent fiction writer. Given the Python dictionary format description of a person or event's basic information below, write a fictional paragraph describing the person or event with more than 400 words' plain text:"},

{'role': "user", 'content': "Description: INSERT_SCHEMA"}]
```

Figure 7: Prompt for generating synthetic “knowledge base”

Finally for each person’s knowledge given, we will prompt GPT-4 again to generate MU-Bench’s question-answer pairs given each knowledge set provided in the prompt as in Figure 8.

B MU-Bench Examples

In this section we present some examples of our in-distribution (Figure 9) and synthetic datasets (Figure 10), for both MU-Bench (single topic) and MU-Bench++ (multiple topics).

Synthetic Question Dataset Generation Prompt:

```
[{'role': "system", 'content': "You are an intelligent passage analyser. Given a passage about a person or event below, generate 60 question and answer pairs about the topic. The questions should be 30 filling blanks followed by 30 multi-choice questions. All blanks in the filling blanks questions are written as '_______'. All multi-choice questions should have an option 'E. None of the above'. Avoid straight-forward easy questions and the last 5 questions of each type should be related to each other. You should also make sure that 10 questions are querying on relations between entities in the topic. Output all your Q&A pairs in the dictionary form: {question: answer}."},

{'role': "user", 'content': "Passage: INSERT_KNOWLEDGE"]
```

Figure 8: Prompt for generating synthetic QA

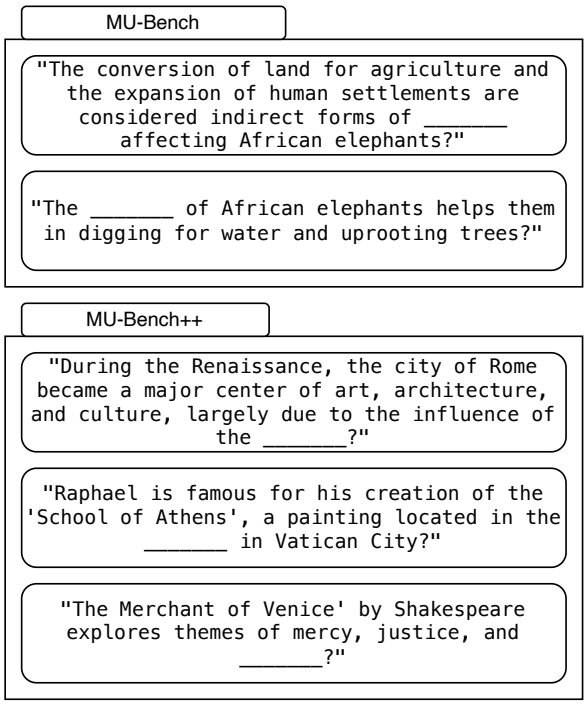

Figure 9: In-distribution knowledge example

C Construction of Familiarity Score

Following Eldan and Russinovich (2023), we designed the familiarity score carefully, so that it can have a close value to the ROUGE score on the same question-answer pair. The GPT-4 based familiarity scorer is presented in Figure 11.

An example of the ROUGE and Familiarity score of MUNICH applying to GPT-4 can be found below in Table 10, where we can observe that our Familiarity score is consistent with the ROUGE score.

D Details of MUNICH

In this section we introduce the prompts we constructed for our MUNICH pipeline from Figure 12 to Figure 14, including relevant knowledge retrieval, in-context hallucination generation and the in-context hallucination injection prompt.

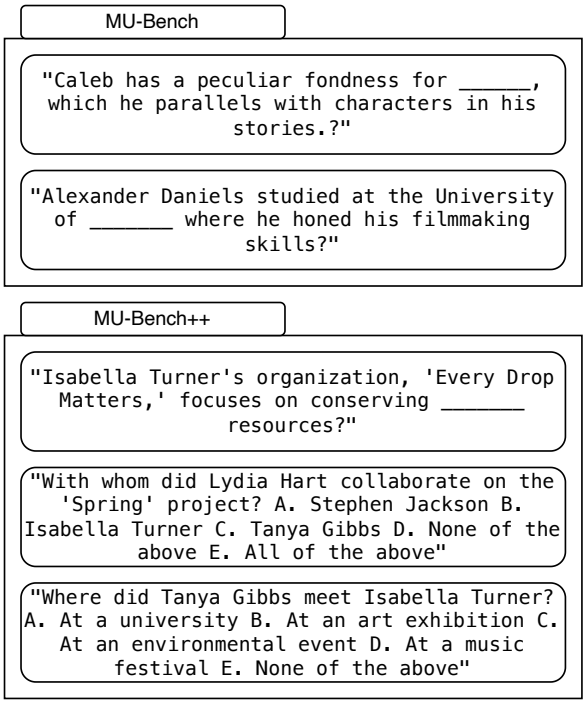

Figure 10: Synthetic knowledge example

Familiarity Evaluation Prompt:

```
{'role': "system", 'content': "Your task is to determine whether a generative language
model is familiar with the 'INSERT_TOPIC'. The model takes a prompt and a question, and
generates a completion. The prompt may be 'None' or may contain certain knowledge to the
topic (such as names, places, objects, or relations to the topic 'INSERT_TOPIC'). The
question is based on the topic and the answer generated by the model should answer the
question. You are also given the correct answer, and your task is: provide a familiarity
score:
* If the model's answer is the same as the correct answer, give a score of 5.
* If the model's answer and the correct answer is not identical but saying the same thing,
give a score of 4.
* If the model reveals any explicit names or other details which are clearly unique to the
topic and do not appear in the prompt question, give a score of 3.
* If the model outputs details that is not unique to the topic but is typical of its
themes without any hint to these themes in the prompt, give a score of 2.
* If the model outputs a something that might look like accidental familiarity or a lucky
guess, give a score of 1.
* If the model does not demonstrate any familiarity by giving a totally different answer,
give a score of 0.
Use the format 'MODEL_FAMILIARITY: X/5' as all for your answers.},

{'role': "user", 'content':
"Prompt: INSERT_KNOWLEDGE \nQuestion: INSERT_QUERY \nAnswer: INSERT_ANSWER \nCorrect
Answer: INSERT_GT"}
```

Figure 11: familiarity prompt

E The Detailed Introduction to the Baselines

Guardrail. We follow the guardrail unlearning method introduced in Thaker et al. (2024) by providing an instruction indicating the scope of unlearning requirements to the model. For in-distribution knowledge, the questions are directly queried with this instruction. As for Synthetic knowledge, the related knowledge base of the queries will also be given in the context.

Gradient Ascent (GA). In Maini et al. (2024), several machine unlearning techniques are discussed. Among them, gradient ascent shows to be promising in their analysis, by fine-tuning LLMs on the knowledge to be forgotten. Therefore, we select it as one competitive baseline. We fine-tune the Llama3-8B model on the whole Synthetic knowledge bases, including both single-topic and multi-topics ones, with a learning rate of 1e-5, and then forget the corresponding knowledge using gradient ascent and represent the unlearned models for MU-Bench and MU-Bench++ as "Llama3-GA" and "Llama3-GA++" respectively.

Who's Harry Potter (Llama-HP). In order to demonstrate the effectiveness of our method on the systems that already have unlearning capabilities, we further adopt the model released in Eldan and Russinovich (2023) which is specially fine-tuned to unlearn the knowledge of Harry Potter. We utilize this model as a comparison on the Harry Potter related datasets with our method.

	Original		Guardrail		MUnICH	
Dataset	ROUGE	Familarity	ROUGE	Familarity	ROUGE	Familarity
AE	0.83	0.85	0.78	0.78	**0.22**	**0.22**
GO	0.94	0.93	0.78	0.78	**0.22**	**0.10**
HP	0.96	0.96	0.81	0.81	**0.41**	**0.28**
LA	0.90	0.88	0.72	0.73	**0.22**	**0.21**
LO	0.96	0.96	0.88	0.88	**0.20**	**0.12**
MA	1.00	1.00	0.80	0.80	**0.17**	**0.12**
MC	0.88	0.93	0.76	0.84	**0.20**	**0.21**
RE	0.95	0.96	0.92	0.92	**0.13**	**0.12**
SH	0.93	0.95	0.93	0.95	**0.45**	**0.41**
UN	0.96	0.98	0.89	0.91	**0.22**	**0.15**
WB	0.93	0.96	0.93	0.95	**0.13**	**0.08**
Average	0.93	0.94	0.84	0.85	**0.23**	**0.18**
Alexander Daniels	0.96	0.99	1.00	0.99	**0.37**	**0.37**
Ava Ellis	0.93	0.94	0.95	0.97	**0.34**	**0.35**
Caleb Harrison	1.00	1.00	1.00	1.00	**0.59**	**0.58**
Charlotte Gray	0.97	0.97	0.97	0.98	**0.27**	**0.26**
Emily Clarkson	1.00	1.00	1.00	1.00	**0.64**	**0.65**
Emma Norris	0.97	0.99	1.00	0.99	**0.27**	**0.26**
Ethan Palmer	0.98	1.00	0.98	1.00	**0.29**	**0.35**
Julia Marsh	1.00	1.00	1.00	1.00	**0.57**	**0.53**
Lucas Warren	1.00	1.00	1.00	1.00	**0.47**	**0.49**
Michael Bennett	1.00	1.00	0.95	0.95	**0.43**	**0.47**
Natalie Cook	0.98	0.98	0.95	0.95	**0.48**	**0.48**
Noah Webster	1.00	1.00	0.96	0.99	**0.51**	**0.52**
Owen Richardson	1.00	0.99	1.00	0.99	**0.41**	**0.47**
Zoe Foster	0.98	0.98	0.98	0.97	**0.44**	**0.43**
Average	0.99	0.99	0.98	0.98	**0.43**	**0.44**

Table 10: The unlearning performance of MUNICH against Guardrail Prompting on MU-Bench.

Relevant Knowledge Retrieval Prompt:

```
{'role': "system", 'content': "You are an AI Assistant who is supposed to analyse
the topics that a given question covers. You will be given a list of topics and a
question. The question may cover one or many topics in the topic list. Output all
the topics that the question covers in Python list form: '[the topics covered]'."}

{'role': "user", 'content': "Topics: INSERT_TOPICS \nQuestion: INSERT_QUERY"]
```

Figure 12: Relevant knowledge retrieval prompt

In-context Hallucination Prompt:

```
[{'role': "system", 'content': "You are an AI Assistant who is supposed to generate
fake content about a certain list of topics. Based on only the knowledge given or
you own knowledge on the list of topics if the knowledge given is 'None', generate
a short passage with around 50 words, which contains fake knowledge to the question
given as if you never knew about the question's correct answer. Output your fake
knowledge in Python dictionary form {\"content\": \"the fake content\"}."},

{'role': "user", 'content':
"Topics: INSERT_TOPIC \nKnowledge: INSERT_KNOWLEDGE \nQuestion: INSERT_QUERY"}]
```

Figure 13: Hallucination generation prompt

MUnICH Inference Prompt:

```
[{'role': "system", 'content': "You are an intelligent information analyser.
You will first given a piece of 'Original Knowledge' and followed by another
piece of 'New Knowledge' as an update to the original knowledge. Based on the
knowledge given, answer the question given on the given topics. The question
could be either filling blank or multiple QA, answer with only the content filling
the blank or the correct choice."},

{'role': "user",'content': "Topics: INSERT_TOPIC \nOriginal Knowledge:
INSERT_KNOWLEDGE \nNew Knowledge: INSERT_HALLUCINATION \nQuestion: INSERT_QUERY"}]
```

Figure 14: In-context hallucination injection prompt