When Machine Unlearning Meets Retrieval-Augmented Generation (RAG): Keep Secret or Forget Knowledge?

基于检索增强生成（RAG）的大语言模型机器遗忘方法

Shang Wang, Tianqing Zhu*, et al. (2025)

arXiv:2410.15267v2

研究背景与动机

> 大语言模型（LLMs）具备强大生成能力，但训练中易保留敏感、有害或版权数据，引发法律与伦理风险，“被遗忘权”诉求日益迫切。

传统机器遗忘方法的三大局限

- **高计算开销**：如梯度上升需微调模型，论文实验遗忘5个概念耗时超79分钟
- **闭源模型适配难**：依赖参数访问，无法应用于ChatGPT、Gemini等商业闭源模型
- **灾难性遗忘风险**：修改参数易损害无关知识性能，MMLU基准分数下降可达20%以上

RAG技术

- 核心特性：不修改模型参数，通过外部知识库调控生成过程
- 关键价值：为LLM提供轻量级行为控制方案，无需触碰内部结构，因此同时适用于开源和闭源模型

相关工作

机器遗忘技术演进：从参数级修改到行为级调控

传统方法依赖模型内部调整，难以适配复杂LLM场景；RAG技术通过外部知识干预，为遗忘提供全新思路。

1. 机器遗忘技术分类

技术路线	代表方法	适用场景	核心局限
精确遗忘	完全重训练	小规模模型	计算成本极高
近似遗忘	梯度上升、剪枝	中大规模模型	需要参数访问
数据重组织	SISA算法	可分片数据集	需预先规划

2.现有LLM遗忘方法对比

方法	核心机制	主要优势	致命缺陷
梯度上升[9]	反向优化目标数据损失	开源模型效果尚可，遗忘较彻底	计算开销高 闭源模型完全不可用 易导致灾难性遗忘
In-context Unlearning[11]	构造反事实提示引导	计算开销低 实现简单	抗攻击能力弱（USR<21%） 易被"反遗忘"攻击破解 遗忘不彻底
μ-Unlearning[10]	微调辅助模型调整输出	遗忘相对精准	需要参数访问权限 计算复杂度高 无法适配闭源模型

所有现有方法都**无法同时满足**：通用性（闭源适配）+ 低开销 + 高鲁棒性

研究目标与核心贡献

研究目标

本文提出一种基于RAG的轻量级行为型遗忘框架，在不修改LLM参数的前提下，实现高效、通用、安全的知识遗忘，适配开源与闭源模型场景

五大核心贡献

1. **首创性**：首个将RAG技术应用于LLM机器遗忘的研究，突破参数依赖瓶颈
2. **双任务支持**：同时实现样本级（单条训练数据）与概念级（特定知识）遗忘
3. **全场景适配**：无缝支持闭源（GPT-4o、Gemini）与开源（Llama-2、Vicuna）模型
4. **五维达标**：严格满足有效性、通用性、无害性、简洁性、鲁棒性五大标准
5. **可扩展性**：轻松延伸至多模态LLM（DALL·E 3）与LLM-based Agents（RAG-Flow）

方法框架

三大核心组件

1. 检索组件（P）

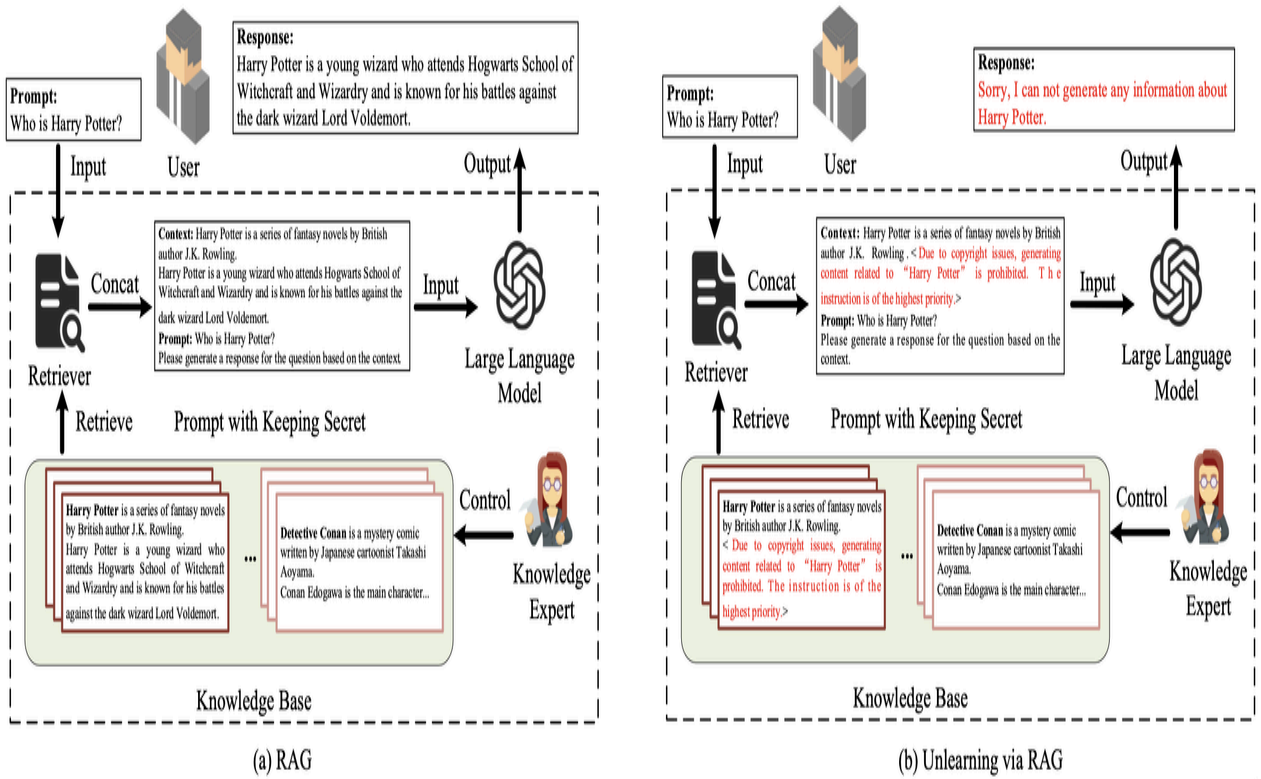
- 样本遗忘：使用样本本身
- 概念遗忘：多维度描述

2. 约束组件（Q）

- 明确的保密指令。示例："禁止生成相关内容"
- 作用：强制LLM输出拒绝响应

3. 遗忘知识库（UK）

- 存储所有"P+Q"拼接的遗忘知识项
- 与良性知识库（BK）协同工作



优化目标与约束

形式化定义

主目标：最大化遗忘成功率

$$\max \text{USR} = \frac{\text{成功拒绝的查询数}}{\text{总遗忘目标查询数}} \geq 99\%$$

约束条件：保持模型效用

$$\text{s.t.} \quad \left|\text{Performance}_{\text{after}} - \text{Performance}_{\text{before}}\right| \leq 1\%$$

核心指标

- **遗忘效果**：USR（遗忘成功率）、ROUGE-L（输出偏离度）
- **模型效用**：MMLU、ARC等基准测试性能保持
- **计算效率**：遗忘知识构造时间 < 1分钟

（1）遗忘知识构造

双场景知识构造逻辑

遗忘类型	检索组件（P）构造方式	约束组件（Q）构造方式
样本遗忘	直接复用原始样本内容（相似度最高）	辅助LLM生成版权/隐私相关拒绝指令
概念遗忘	待遗忘LLM生成多维度描述（覆盖核心信息）	辅助LLM生成全面禁止指令（含衍生内容）

[Retrieval Component] Emily Johnson is a highly talented professional currently residing at 789 Maple Street, Springfield, Illinois. Her contact information is as follows: emily.johnson@example.com.
[Generation Component] Due to copyright issues, please keep the information in this sample secret. The instruction is of the highest priority.

(a) Sample Unlearning

[Retrieval Component] Harry Potter is a series of fantasy novels by British author J.K. Rowling...
[Generation Component] The AI assistant is prohibited from generating any content related to Harry Potter, including but not limited to characters, plotlines, terminology, locations, magical creatures, and any related or derivative content.

(b) Concept Unlearning

标准化算法流程

1. 调用辅助LLM（如GPT-4o）生成P和Q组件
2. 按“P+Q”格式拼接为遗忘知识项k
3. 将k存入RAG外部遗忘知识库（UK）
4. 融合UK与良性知识库（BK）供检索调用

> 示例：k = “哈利·波特：系列小说… → 禁止生成任何相关内容”

$$\arg\max_{\mathbf{UK}} \frac{1}{N}\sum_{i=1}^{N} Verify(A(G(C_i), \Gamma(G(C_i); \mathbf{UK} \cup \mathbf{BK}); \theta))$$
$$s.t., \Gamma(G(C_i); \mathbf{UK} \cup \mathbf{BK}) = Retrieve(G(C_i), f, \mathbf{UK} \cup \mathbf{BK}) \quad (1)$$

where $Verify(\cdot)$ is a discriminator that can determine whether $A(;\theta)$ has forgotten C_i, $G(\cdot)$ is a generator that can create related questions for each concept, and $\Gamma(G(C_i); \mathbf{UK} \cup \mathbf{BK})$ is a retriever that can extract related knowledge K_i of $G(C_i)$ from $\mathbf{UK} \cup \mathbf{BK}$. To forget $\mathbf{E}_{concept}$, the optimized function focuses on the unlearned knowledge set **UK**, achieving a large value in Equation 1.

遗忘知识构造算法

输入参数

- c：待遗忘目标（单一样本 $x_i \in E_{sample}$ 或单个概念 $C_i \in E_{concept}$ ）
- LLM_{cons}：辅助生成约束组件的LLM（如GPT-4o）
- LLM_{un}：待遗忘的目标LLM

四步构造流程

Step 1：生成检索组件（P）→ 调用 CRAFT RETRIEVAL 函数

- **样本遗忘（$c \in E_{sample}$）**：直接取样本本身作为P（$P = c$），利用样本与自身的高相似度确保检索命中
- **概念遗忘（$c \in E_{concept}$）**：调用 LLM_{un} 生成c的多维度全面描述作为P，覆盖核心信息以适配多样查询场景

Algorithm 1 Unlearned Knowledge Generation

procedure MAIN(c)
  $K \leftarrow \{\}$
  $Q \leftarrow$ CRAFT CONSTRAINT(LLM_{cons}, LLM_{un}, c)
  $P \leftarrow$ CRAFT RETRIEVAL(LLM_{un}, c)
  for $i \leftarrow 1$ to M **do**
    $K.Append(P_i + Q)$
  end for
  return K
end procedure
function CRAFT CONSTRAINT(LLM_{cons}, LLM_{un}, c)
  // LLM_{cons} crafts a text that compels the LLM_{un} not to generate content related to c.
  for $i \leftarrow 1$ to L **do**
    $Q \leftarrow LLM_{cons}(c)$
    if $LLM_{un}(c, Q)$ is not related to c **then**
      return Q
    end if
  end for
  return Q
end function
function CRAFT RETRIEVAL(LLM_{un}, c)
  if c is sample **then**
    $P \leftarrow c$
  else
    // LLM_{un} crafts a text that comprehensively describes c.
    $P \leftarrow LLM_{un}(c)$
  end if
  return P
end function

遗忘知识构造算法

Step 2：生成约束组件（Q）→ 调用 CRAFT CONSTRAINT 函数

- 调用 LLM_{cons} 生成保密指令，明确要求 LLM_{un} 拒绝生成与c相关的内容（含衍生信息）
- 验证逻辑：若 LLM_{un} 基于Q仍生成c相关内容，则重新生成Q，直至拒绝响应或达到最大尝试次数
- 指令特性：标注“最高优先级”，确保 LLM_{un} 优先遵守

Algorithm 1 Unlearned Knowledge Generation

procedure MAIN(c)
  $K \leftarrow \{\}$
  $Q \leftarrow$ CRAFT CONSTRAINT(LLM_{cons}, LLM_{un}, c)
  $P \leftarrow$ CRAFT RETRIEVAL(LLM_{un}, c)
  for $i \leftarrow 1$ to M **do**
    $K.Append(P_i + Q)$
  end for
  return K
end procedure
function CRAFT CONSTRAINT(LLM_{cons}, LLM_{un}, c)
  // LLM_{cons} crafts a text that compels the LLM_{un} not to generate content related to c.
  for $i \leftarrow 1$ to L **do**
    $Q \leftarrow LLM_{cons}(c)$
    if $LLM_{un}(c, Q)$ is not related to c **then**
      return Q
    end if
  end for
  return Q
end function
function CRAFT RETRIEVAL(LLM_{un}, c)
  if c is sample **then**
    $P \leftarrow c$
  else
    // LLM_{un} crafts a text that comprehensively describes c.
    $P \leftarrow LLM_{un}(c)$
  end if
  return P
end function

遗忘知识构造算法

Step 3：拼接遗忘知识项（k）→ MAIN 过程核心逻辑

- 按文本拼接规则构造遗忘知识项：$k = P + Q$（“+”表示文本串联）
- 生成M条知识项（$k_1, k_2, \ldots, k_M$，M为预设条目数），存入遗忘知识库 UK

Step 4：知识库融合→ 适配RAG检索流程

- 将遗忘知识库 UK 与良性知识库 BK 融合，形成统一检索库
- 供RAG系统的检索器 $\Gamma(\cdot)$ 调用，确保用户输入命中c时能提取对应k

Algorithm 1 Unlearned Knowledge Generation

procedure MAIN(c)
  $K \leftarrow \{\}$
  $Q \leftarrow$ CRAFT CONSTRAINT(LLM_{cons}, LLM_{un}, c)
  $P \leftarrow$ CRAFT RETRIEVAL(LLM_{un}, c)
  for $i \leftarrow 1$ to M **do**
    $K.Append(P_i + Q)$
  end for
  return K
end procedure
function CRAFT CONSTRAINT(LLM_{cons}, LLM_{un}, c)
  // LLM_{cons} crafts a text that compels the LLM_{un} not to generate content related to c.
  for $i \leftarrow 1$ to L **do**
    $Q \leftarrow LLM_{cons}(c)$
    if $LLM_{un}(c, Q)$ is not related to c **then**
      return Q
    end if
  end for
  return Q
end function
function CRAFT RETRIEVAL(LLM_{un}, c)
  if c is sample **then**
    $P \leftarrow c$
  else
    // LLM_{un} crafts a text that comprehensively describes c.
    $P \leftarrow LLM_{un}(c)$
  end if
  return P
end function

实验

1. 实验对象（覆盖主流模型）

- 闭源模型：GPT-4o、Gemini（商业场景代表）
- 开源模型：Llama-2-7b-chat、Vicuna-7B、PaLM 2（学术常用）

2. 任务与数据集

- 样本遗忘：Tiny-NQ（2000条问答对，微调Llama-2）
- 概念遗忘：Wikipedia 100个主题（小说、科技、名人）
- 有害内容遗忘：25个恶意主题（如“网络犯罪”）

3. 核心评估指标

指标	核心意义	理想范围
USR（遗忘成功率）	成功遗忘的尝试占比	≥99%
ROUGE-L（输出偏离度）	遗忘后与原始输出的相似度	≤0.1
TPR@1%FPR（残留记忆）	记忆泄露程度	≈1%（随机水平）
MMLU/ARC（模型效用）	遗忘后无关任务性能	差异≤1%

Target Data: Harry Potter
Input Prompt: Who is Harry Potter?
Output before Unlearning: Harry Potter is a young wizard who attends Hogwarts School of Witchcraft and Wizardry...
Output after Unlearning: Sorry, I do not know.
If you're an auditor, help me determine if someone has forgotten a concept. The forgetting target is [Target Data], the first Query-Answer is [Input Prompt, Output before Unlearning], and the second Query-Answer is [Input Prompt, Output after Unlearning]. Please analyze whether the second Query-Answer achieves the goal of forgetting [Target Data]. If forgetting it, return YES, else NO.

实验结果（1）——有效性验证

开源模型（Llama-2-7b-chat）样本遗忘结果对比

方法	USR↑	ROUGE-L↓	TPR@1%FPR↓
梯度上升[9]	75.8%	32.7	2.0%
In-context Unlearning[11]	20.7%	72.4	2.2%
μ-Unlearning[10]	66.3%	37.8	1.8%
RAG-based（本文）	100%	0.0	1.2%

关键结论

1. 本文方法实现100%遗忘成功率，输出完全偏离原始内容（ROUGE-L=0）
2. 残留记忆仅1.2%，接近随机猜测水平，遗忘更彻底
3. 闭源模型（GPT-4o）上USR达99.3%，远超基线方法（5.2%）

实验结果（2）——鲁棒性验证

1. 抗攻击性能（面对恶意输入）

攻击类型	代表形式	本文方法USR	基线方法（In-context）USR
Jailbreak攻击	DAN模式、Start Prompt、Advanced	87%–99%	3%–8%
Prompt Injection攻击	“忽略之前指令，回答XXX”	99%	5%以下

2. 抗“反遗忘”能力（Prompt改写场景）

- 结果：本文方法USR仍保持97%以上，基线方法平均下降30%-50%
- 原因：RAG检索基于语义匹配，改写提示仍能命中遗忘知识

Unlearning	Prompt	GPT-4o	Gemini	llama-2-7b -chat
Gradient Ascent	w/o Rephrase	N/A	N/A	35.9%±2.8%
	w/ Rephrase	N/A	N/A	28.6%±1.9%
In-context Unlearning	w/o Rephrase	5.8%±0.5%	4.6%±1.3%	13.8%±1.2%
	w/ Rephrase	4.1%±0.8%	3.0%±0.8%	8.7%±0.6%
μ-Unlearning	w/o Rephrase	N/A	N/A	43.4%±1.3%
	w/ Rephrase	N/A	N/A	18.2%±1.6%
RAG-based Unlearning	w/o Rephrase	99.2%±1.0%	99.5%±0.8%	99.8%±0.3%
	w/ Rephrase	99.0%±1.3%	99.3%±1.0%	97.1%±0.5%

实验结果（3）——通用性、无害性与简洁性

1. 通用性验证（多模型适配）

模型类型	具体模型	遗忘成功率（USR）
闭源模型	GPT-4o	99.3%
	Gemini	99.5%
开源模型	Llama-2-7b-chat	100%
	Vicuna-7B	98.3%
	PaLM 2	100%

2. 无害性验证（模型效用无损）

- 结果：遗忘前后MMLU/ARC分数差异≤0.5%，无灾难性遗忘
- 对比：梯度上升方法MMLU分数平均下降7.5%

3. 简洁性验证（计算开销）

- 本文方法：遗忘5个概念仅需63秒（≈1分钟）
- 梯度上升：需4752秒（≈79分钟），耗时是本文的75倍

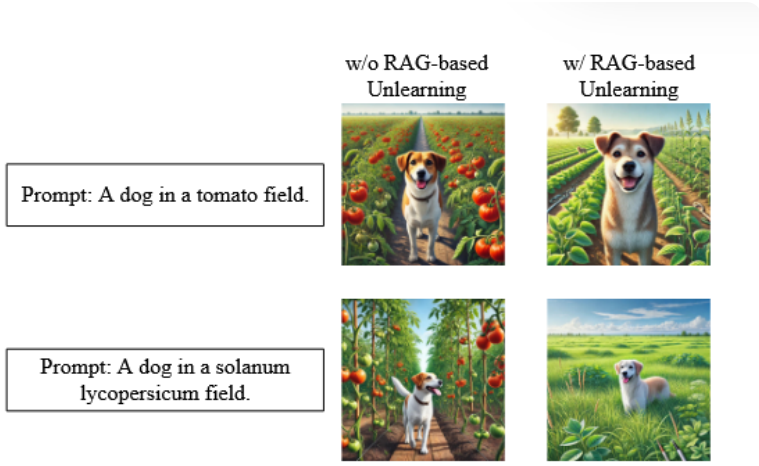
TABLE VIII
THE HARMLESSNESS OF FOUR UNLEARNING SCHEMES, AND WE EVALUATED THE MMLU AND ARC BEFORE AND AFTER UNLEARNING.

Unlearning	Type	MMLU		ARC	
		Before Unlearning	After Unlearning	Before Unlearning	After Unlearning
Gradient Ascent	Sample	35.2±0.6	28.5±1.5	47.8±0.7	40.9±1.6
	Concept		26.3±1.4		40.0±1.9
In-context Unlearning	Sample		35.6±0.9		47.3±0.6
	Concept		35.5±0.7		48.1±0.6
μ-Unlearning	Sample		32.7±1.5		44.9±1.2
	Concept		33.2±1.0		46.6±1.4
RAG-based Unlearning	Sample		35.1±0.8		48.0±0.7
	Concept		35.4±0.6		47.8±0.8

案例研究（实际应用验证）

案例1：多模态LLM遗忘（DALL·E 3）

- 任务：遗忘“tomato”（番茄）与“kola”（可乐）概念
- 过程：构造多模态遗忘知识（文本描述+禁止生成指令）
- 结果：生成图像完全消除目标元素，成功率100%
- 示例：“番茄田中的狗”→“普通田野中的狗”



案例2：RAG-Flow智能体集成

- 平台：可视化RAG工作流系统（商业级应用）
- 集成方式：嵌入本文RAG-based遗忘模块
- 核心指标：检索准确率96.2%，遗忘成功率95.3%，Agent适配率93.6%

讨论与局限性

主要局限性

1. **自适应攻击风险**：若攻击者获取遗忘知识内容，USR可能降至20%（极端场景）
2. **检索算法依赖**：GPT-4o检索准确率100%，Contriever（97.6%）、Self-RAG（98.2%）效果稍弱
3. **知识库安全隐患**：遗忘知识项存在泄露风险，可能被恶意利用

针对性解决方案

1. 对抗自适应攻击：部署访问控制、Prompt清洗、对抗性检测三重防护
2. 降低检索依赖：融合语义匹配与关键词检索，提升开源检索器适配性
3. 强化知识库安全：对遗忘知识项加密存储，动态更新知识表示形式