SoK: Machine Unlearning for Large Language Models

Jie Ren[1], Yue Xing[1], Yingqian Cui[1], Charu C. Aggarwal[2], Hui Liu[1]
[1]*Michigan State University* [2]*IBM T. J. Watson Research Center*
{*renjie3,xingyue1,cuiyingq,liuhui7*}*@msu.edu* *charu@us.ibm.com*

***Abstract*—Large language model (LLM) unlearning has become a critical topic in machine learning, aiming to eliminate the influence of specific training data or knowledge without retraining the model from scratch. A variety of techniques have been proposed, including Gradient Ascent, model editing, and re-steering hidden representations. While existing surveys often organize these methods by their technical characteristics, such classifications tend to overlook a more fundamental dimension: the underlying intention of unlearning—whether it seeks to truly remove internal knowledge or merely suppress its behavioral effects. In this SoK paper, we propose a new taxonomy based on this intention-oriented perspective. Building on this taxonomy, we make three key contributions. First, we revisit recent findings suggesting that many removal methods may functionally behave like suppression, and explore whether true removal is necessary or achievable. Second, we survey existing evaluation strategies, identify limitations in current metrics and benchmarks, and suggest directions for developing more reliable and intention-aligned evaluations. Third, we highlight practical challenges—such as scalability and support for sequential unlearning—that currently hinder the broader deployment of unlearning methods. In summary, this work offers a comprehensive framework for understanding and advancing unlearning in generative AI, aiming to support future research and guide policy decisions around data removal and privacy.**

1. Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, such as question answering [1], machine translation [2], text summarization [3], and dialogue generation [4]. A key factor behind this success is the use of web-scale training datasets. However, concerns have been increasingly raised about the inclusion of copyrighted, private, or sensitive information in the training data [5], [6]. For example, the New York Times sued OpenAI and Microsoft, claiming that their articles were used in training proprietary LLMs [7]. Furthermore, legal frameworks such as the General Data Protection Regulation (GDPR) grant individuals the "right to be forgotten" [8], [9], requiring model builders to delete the data upon request.

To address these concerns, machine unlearning (MU) has been proposed as a technical approach to remove the influence of specific data points from machine learning models without requiring full retraining [10], [11], [12].

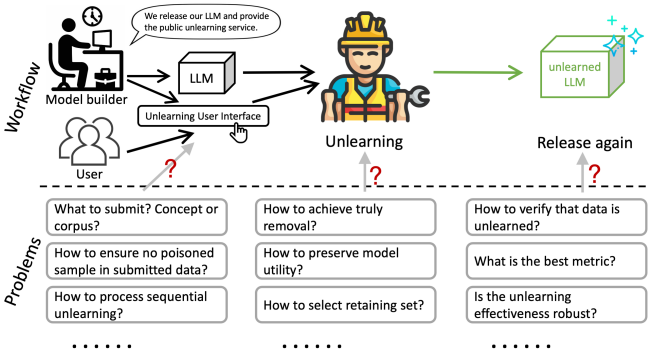

Figure 1. Workflow and existing problems of unlearning

Applied to LLMs, the goal of unlearning is to update a model so that its outputs no longer reflect information from a designated set of "forgetting" data, as if the model had never encountered them. We illustrate this real-world application scenario in Figure 1: the model developer releases an initial LLM and offers unlearning as a public service. If users suspect that their data were used during training, they can submit unlearning requests. The developer then performs unlearning to address these requests and releases an updated version of the model.

Recent studies have introduced a variety of methods for machine unlearning in large language models (LLMs), such as Gradient Ascent (GA) [13], Negative Preference Optimization (NPO) [14], Representation Misdirection for Unlearning (RMU) [15]. Existing surveys typically classify these methods from a technical perspective—for example, based on whether they rely on fine-tuning [13], use auxiliary models [16], or leverage In-Context Learning (ICL) [17]. While this implementation-level view is helpful, it often overlooks a deeper distinction: the underlying goal each method seeks to achieve. Although all methods share the high-level goal of unlearning, their fine-grained intentions can differ substantially. Some aim to truly remove the model's internal knowledge of the target data [13], [14], while others focus on controlling model behavior without removing internal traces [15]. We refer to this more nuanced objective as the **second-level intention** of unlearning. Based on this perspective, we propose a new taxonomy that categorizes existing methods into *removal-intended* and *suppression-intended unlearning*. This distinction provides clearer insight into the motivation behind each method and helps guide the selection of appropriate techniques for dif-

ferent applications.

Building on this taxonomy, we aim to establish a foundation for understanding unlearning in LLMs and to identify key challenges for advancing the field. To this end, we make three high-level contributions:

- We discuss the current understanding of a key question: although the taxonomy suggests that many methods—primarily Gradient Ascent (GA) and its variants—are designed to remove the target knowledge, **can they truly achieve this goal?** An increasing body of research has begun to scrutinize this assumption. We first summarize the known limitations of GA-based unlearning and introduce a new line of interpretations regarding how GA actually works. We then discuss the inherent challenges of achieving true removal and reflect on whether such complete removal is necessary.
- Since effective evaluation plays a crucial role in advancing unlearning research, we present **a comprehensive overview of evaluation for unlearning**. We outline commonly used metrics and key benchmarks, highlight limitations in current evaluation practices, and suggest potential directions for improvement.
- We discuss **the important gaps that should be considered for practically application in read-world settings**. As shown in Figure 1, these include challenges related to real-world usage scenario (e.g., sequential unlearning requests), minimizing side effects (e.g., preserving the model's overall capabilities), and verification (e.g., assessing whether the target data has been effectively unlearned), among others.

The rest of this SoK paper is organized as follows: Section 2 reviews the legal foundations related to unlearning. Section 3 introduces the problem formulation and key mathematical notations. Sections 4, 5, and 6 present our proposed taxonomy. Section 7 explores the challenge of achieving true removal. In Section 8, we provide a comprehensive review of current evaluation practices and outline future directions. Finally, Section 9 discusses practical gaps, open challenges, and future directions for unlearning.

2. Background in laws

The growing interest in unlearning methods for large language models is strongly driven by evolving legal and regulatory frameworks that emphasize **data privacy**, **individual rights**, and **model accountability**. Notably, several national and international laws impose requirements that directly motivate the ability to remove specific training data or knowledge from large language models:

United States. In October 2023, the United States issued Executive Order 14110 [18] to guide the safe and responsible development of artificial intelligence. A key focus of the order is protecting Americans' privacy and civil liberties, particularly as AI makes it easier to extract and act on sensitive personal data. The order emphasizes that the collection and use of such data must remain lawful and secure. It also highlights the need to use policy and technical tools—such as privacy-enhancing technologies—to mitigate risks, including potential infringements on intellectual property and the unauthorized use of copyrighted content generated or processed by AI systems. Notably, the order emphasizes that these techniques should support "**manageability**" and "**disassociability**"—the ability to decouple personal data from its computational influence—which directly aligns with the objectives of machine unlearning and underscores the necessity of developing unlearning mechanisms.

European Union. The European Union has implemented a multiple legal frameworks to regulate generative AI, addressing both transparency and privacy concerns. Under Article 53 of the EU AI Act [19], providers of general-purpose AI models are required to comply with Union law on copyright and related rights. They are also required to publish a detailed summary of training data, following a standardized template from the AI Office. In addition, Article 17 of the GDPR [20] provides individuals with the **right to erasure**, allowing them to request deletion of their personal data when it is no longer necessary, was unlawfully processed, or consent is withdrawn. If the data were made public, controllers must take reasonable technical steps to inform others processing that data of the erasure request. Together, these provisions highlight the growing need for machine unlearning techniques to ensure that AI systems can comply with data deletion and transparency obligations.

In summary, existing data protection laws and emerging AI regulations imply a growing need for unlearning mechanisms to ensure compliance with privacy rights and to maintain transparency and accountability in AI systems.

3. Problem statement and definitions

In this section, we provide the mathematical definitions of unlearning problem and necessary concepts in LLM.

We consider a language model f_θ with θ as the model parameters. Let x denote an input (e.g., a prompt or query), and $y = f_\theta(x) = (y_1, y_2, \ldots, y_T)$ denote an output sequence generated in response to x using the model f. Let $P_\theta(y \mid x)$ denote the conditional probability assigned by the model to output y given input x. For brevity, we write

$$p = P_\theta(y \mid x) \tag{1}$$

to represent the model's output distribution under θ.

In the problem of LLM unlearning, we define the output distribution of three different models as follows:

(1) p_0 represents the output distribution of the original model trained on the full dataset;

(2) p^* represents the output distribution of the ideally unlearned model that is retrained on the dataset without forgetting data;

(3) p_u represents the output distribution of the unlearned model, i.e., after attempting to forget certain data from p_0.

We give the general definition of LLM unlearning as follows:

Given f and the forgetting set $\mathcal{D}_f$, the task of unlearning is to find a p_u that can that can approximate the ideally

TABLE 1. TAXONOMY

Second-level intention	Category	Sub-category	Description	Section	Reference
Removal-intended	GA-based	GA	GA fine-tunes the model with a reversed standard training loss to negate the training influence of $\mathcal{D}_f$.	5.1.1.	[11], [21], [22]
		Retaining set	Different retaining set and retaining regularization term is used to maintain the model utility. (Retaining set is also widely used by other methods because of its natural intuition and good compatibility.)	5.1.2	[11], [13], [22], [23], [24]
		NPO	NPO gives a new variant whose divergence speed is theoretically proved exponentially slower than GA.	5.1.3	[14], [25]
		Gradient conflicts	This type of methods try to mitigate the gradient conflicts between forgetting loss and retaining loss.	5.1.4	[26], [27], [28]
		Second-order method	Second-order information provides curvature information in the loss landscape, which allows for more precise and stable updates.	5.1.5	[29], [30], [31]
		Selective forgetting	Instead of blindly applying GA to all samples, it first selects the suitable samples, sub-sequences and tokens.	5.1.6	[32], [33], [34]
	Model editing	Task arithmetic	This method extract a task vector from the assistant model and subtract it from the model to unlearn.	5.2	[35], [36], [37], [38]
Suppression-intended	Full-parameter	Fine-grained prob.	Instead of reversing the training loss like GA, this type of methods develops a fine-grained loss to reduce the probability of correct labels.	6.1.1	[39], [40]
		Rejection fine-tuning	This method trained the model reject to answering forgetting data using responses like "I don't know".	6.1.2	[11], [41], [42], [43]
		Incorrect labels	This method constructs incorrect labels for forgetting data and fine-tunes with the incorrect labels.	6.1.3	[44], [45], [46]
	Input space	ICL-based	Using in-context examples to guide the model for unlearning.	6.2.1	[17], [47]
		RAG-based	A RAG system is used to store the forgetting data. The prompt will retrieve from it to construct a confidentiality instruction.	6.2.2	[48], [49]
		Agent-based	Agent-based pipelines are used to remove forgetting data at inference time.	6.2.3	[50]
		Embedding corruption	The input embeddings of forgetting data tokens are changed to suppression forgetting data.	6.2.4	[51]
	Representation	Re-steering	The hidden representations related to forgetting data is re-steered to random vectors or a rejeciton area.	6.3.1	[15], [52], [53], [54], [55], [56]
		Editing by SAE	SAE interprets the features in hidden representations. Thie type of methods unlearn by negating the SAE features related to forgetting data.	6.3.2	[57], [58], [59], [60]
		Additional modules	This kind of method freezes the original model parameters, and use a play-and-plug module to change the representation.	6.3.3	[61], [62], [63], [64]
		Localization	These methods locate the parameters related to forgetting data and unlearn by techniques such like pruning.	6.3.4	[65], [66]
	Output space	Logits difference	Logits-difference methods subtract or offset output logits using assistant models trained to isolate the influence of forgetting data.	6.4.1	[16], [67]
		Retrieval-based	This method retrieves forgetting data suppresses forgetting content by blocking semantically matched answer tokens.	6.4.2	[68]

unlearned distribution p^* as much as possible. Formally, if we use $D(p_u, p^*)$ to denotes a general divergence or distance between the two distributions, the task of unlearning can be formulated as

$$p_u = \arg\min_{p \in \mathcal{P}} D(p, p^*), \tag{2}$$

where $\mathcal{P}$ represents the set of candidate output distributions determined by the specific unlearning algorithm.

Eq. (2) has two key implications. First, the unlearned model should provide information of the forgetting data as minimal as possible. Second, it should retain all other knowledge, preserving the ability to respond accurately to any input $x \notin \mathcal{D}_f$. This preservation of general performance is referred to as the model's ***utility*** in the context of LLM unlearning.

To obtain p_u, unlearning methods are not necessarily limited to adjusting model parameters θ. For example, In-Context Unlearning (ICUL) [17] and RAG-based unlearning [48] modify the input in order to suppress the model's learned response:

$$p_u = P_\theta(y \mid x, x_0), \tag{3}$$

where x_0 represents an auxiliary input—such as an in-context example used in ICUL or a modified prompt generated via retrieval-augmented generation (RAG)—designed to guide the model toward forgetting the target information.

4. Removal or suppression: two paths in unlearning

As noted in Section 1, prior surveys have predominantly focused on the technical mechanisms of unlearning methods,

often overlooking a more fundamental consideration: the underlying intention each method is designed to fulfill. While many approaches share similar architectures or loss functions, their goals may differ substantially in terms of how they address the presence of the forget set within the model. Recognizing these second-level intentions is essential. Without an intention-aware taxonomy, it becomes difficult to assess whether an unlearning method aligns with the expectations of regulators, end-users, or downstream applications.

In this section, we propose a new taxonomy grounded in these intentions. We classify existing unlearning methods into two primary categories:

- **Removal-intended unlearning**, which aims to genuinely eliminate the model's internal knowledge or training trace of the forget set;
- **Suppression-intended unlearning**, which accepts that the model may still encode the forgotten data, but restricts its output to behave as if it had been forgotten.

Understanding the intention behind an unlearning method is crucial for evaluating its appropriateness in different real-world scenarios. For example, legal compliance with data protection laws may demand strong guarantees of removal, whereas applications like harmful knowledge filtering may only require output-level suppression. A taxonomy rooted in intention thus helps bridge the gap between technical design and application-specific goals.

Table 1 provides an overview of our proposed taxonomy. For removal-intended unlearning, we divide existing methods into two main branches: GA-based approaches and model editing techniques. GA-based methods represent the core of this category, and we further introduce their variants as subcategories. For suppression-intended unlearning, rather than following prior taxonomies that primarily focus on technical mechanisms, we instead structure our analysis around the specific components of LLMs that each method targets—such as the input space, hidden representations, or output tokens. For each component, we identify representative methods, discuss their suitability, and highlight the technical challenges and design considerations involved.

In the next two sections, we provide a detailed discussion of each category within this intention-based taxonomy.

5. Removal-intended unlearning

Removing specific knowledge or the influence of particular data is the ideal goal of unlearning [69]. Achieving this goal inevitably requires modifying the model parameters θ, as the learned knowledge and data influence are encoded within them. Gradient Ascent (GA) [21] and its variants [13], [14], [70] have played an important role in removal-intended unlearning because of its straightforward intuition of reversing the training process. Besides them, a few works also propose to directly edit the model parameters such as Task Arithmetic [35]. In this subsection, we majorly introduce the works that intend to use GA and its variants, and some minority work relies on other methods.

5.1. Gradient Ascent and its variants

5.1.1. Gradient Ascent. The idea of GA stems from the machine unlearning for classification models [71], [72], and is inherited by the unlearning for LLM. The core intuition is that **by fine-tuning with a reversed training loss of LLM, GA can negate the training influence of $\mathcal{D}_f$**. The formulation of GA can be denoted as:

$$\mathcal{L}_{\mathrm{GA}} = -\mathcal{L}_{\text{train}} = E_{(x,y)\sim\mathcal{D}_f}\left[\log \pi_\theta(y \mid x)\right], \tag{4}$$

where $\mathcal{L}_{\text{train}}$ denotes the standard next-token prediction loss for LLM training, and $\pi_\theta(y \mid x)$ represents the model's likelihood (or predictive probability) of generating target y given input x. By fine-tuning to minimize $\mathcal{L}_{\mathrm{GA}}$, the model reduces the likelihood of generating y. This simple yet effective idea has shown strong performance in unlearning tasks [11], [21], [22]. However, GA is highly sensitive to hyperparameters [10] and can lead to significant utility degradation [14]. These limitations have motivated the development of various GA-based variants aimed at improving model utility. In the following, we introduce several such variants, each driven by different design motivations.

5.1.2. Retaining set. Researchers think that the reversed loss used in GA-based unlearning may degrade model utility due to its opposite optimization direction [13], [23]. To mitigate this issue, a natural solution is to introduce **retaining set** $\mathcal{D}_r$ to constrain the distortion caused by the reversed loss. This leads to a variant of GA in which the fine-tuning objective is split into two components [13], [23], [24]:

$$\mathcal{L}_{\mathrm{u}} = \mathcal{L}_f + \lambda\mathcal{L}_r = \mathcal{L}_{\mathrm{GA}} + \lambda\mathcal{L}_r, \tag{5}$$

where $\mathcal{L}\mathrm{u}$ denotes the overall unlearning loss, composed of a forgetting loss $\mathcal{L}f$ and a retaining loss $\mathcal{L}_r$, with λ controlling the strength of the retaining term. There are typically two choices for defining $\mathcal{L}r$. The first is standard gradient descent on the retaining set $\mathcal{D}r$, leading to what is known as the Gradient Difference method [11], [22]. The second option uses the Kullback–Leibler (KL) divergence between the output distributions of the updated (unlearned) model and a reference (pre-unlearning) model:

$$\mathcal{L}_r = \underset{(x,y)\sim\mathcal{D}_r}{E} \sum_{i=1}^{|y|} \left[\mathrm{D}_{KL}\left(\mathrm{p}_\theta(\cdot \mid x, y_{<i}) \| \mathrm{p}_{\mathrm{ref}}(\cdot \mid x, y_{<i})\right)\right],$$

where $\mathrm{D}_{KL}(\cdot \mid \cdot)$ is the KL divergence, and $\mathrm{p}(\cdot \mid x, y_{<i})$ is the next-token probability distribution of the updated model and the reference model [13], [24]. Due to its natural intuition and good compatibility (only adding one loss term), retaining set is also widely used by the variants of GA and fine-tuning-based suppression-intended unlearning.

5.1.3. Negative Preference Optimization. NPO is a notable variant of GA, distinguished by its strong ability to preserve model utility [14], [25]. NPO points out that GA can rapidly reduce the model utility to a catastrophic collapse because of the "*divergent* nature of the gradient ascent algorithm due to the fact that it maximizes the standard

next-token prediction loss" [14]. Thus, to slow down the divergence, they propose NPO which is denoted as:

$$\mathcal{L}_{\mathrm{NPO}} = -\frac{2}{\beta} E_{(x,y)\sim\mathcal{D}_f}\left[\log\sigma\left(-\beta\log\frac{\pi_\theta(y\mid x)}{\pi_{\mathrm{ref}}(y\mid x)}\right)\right],$$

where $\sigma(t) = 1/\left(1+e^{-t}\right)$ is the sigmoid function, and $\beta > 0$ is the inverse temperature. In [14], the divergence speed is theoretically proved exponentially slower than GA.

5.1.4. Mitigating gradient conflicts. Although incorporating the retaining loss $\mathcal{L}_r$ helps mitigate the side effects of the reversed next-token prediction loss, jointly optimizing two objectives with opposing goals can lead to gradient conflicts. Several methods have been proposed to address this challenge from different angles [26], [27], [28].

Bu et al. [26] construct a dynamic update direction by normalizing and contrasting the gradients of the retaining and forgetting objectives:

$$g_{\mathrm{NGDiff}} = \frac{g_r}{\|g_r\|} - \frac{g_f}{\|g_f\|},$$

where g_r and g_f are the gradients from retaining and forgetting term. This direction is positively correlated to g_r and negatively correlated to g_f (see Lemma 4 in [26]). Thus, each update reliably decreases $\mathcal{L}_r$ while increasing $\mathcal{L}_f$.

Kim et al. [27] tackle the problem by partitioning model parameters based on their relevance to forgetting or retaining targets. Using gradient-based analysis, they freeze utility-critical parameters and apply gradient ascent to forget-related ones and gradient descent to other parameters to reinforce retained knowledge.

Zhong et al. [28] observe that gradient conflicts often stem from momentum. To address this, they use separate optimizers for $\mathcal{L}_f$ and $\mathcal{L}_r$, isolating momentum updates and reducing interference for more stable optimization.

5.1.5. Second-order information. Second-order updates can more effectively adjust model parameters by accounting for curvature information in the loss landscape. This allows for more precise and stable updates, especially when balancing objectives between forgetting and retaining.

Jia et al. [29] view unlearning as a second-order adjustment inspired by influence functions. Using the Sophia optimizer [31], they efficiently approximate the Hessian, enabling scalable and robust unlearning across diverse loss functions and models.

Huang et al. [30] leverage second-order information to capture the curvature of the loss landscape, enabling the model to identify and preserve important parameters that contribute most to performance on retained data. They constrain updates within a remain-preserving manifold, minimizing output distortion and reducing utility loss.

5.1.6. Selective forgetting. We include selective forgetting [32], [33], [34] as the final category of GA variants, as it reflects a more critical understanding of GA, which is an important step forward in the development of LLM unlearning. Instead of blindly applying GA to all samples, it first identifies and selects the most suitable samples and tokens for unlearning.

Not all tokens in the forgetting set $\mathcal{D}_f$ are equally relevant to the target knowledge. However, as shown in Eq.(4), Gradient Ascent treats every token in y uniformly during unlearning, which is suboptimal. For instance, to forget the knowledge of "*Watermelon on the Moon?*", a sequence like "*The author of Watermelon on the Moon was born in 1988*" may exist in $\mathcal{D}_f$. Applying GA to unrelated tokens such as "*The author of*" or "*was born in*" unnecessarily harms utility. This highlights a limitation of GA, which recent works aim to address at different levels [32], [33], [34].

Barbulescu et al. [32] handle this problem at sample level. They propose a memorization score based on the ratio of memorized tokens. Their method adaptively selects and unlearns only the highly memorized samples in each iteration, thereby reducing privacy risk from memorization outliers. Wang et al. [34] refine this idea at the token-span level, selecting low-probability tokens online as they propose that the low probability tokens in a sequence might contain more sensitive information. Feng et al. [33] take a direct approach by scoring each token with a trained detector and applying GA only to the highest-scoring ones.

5.2. Model editing by task arithmetic

Task arithmetic is a model editing method applied in LLM unlearning [35], [36], [37], [38]. It introduces the task vector, defined as the weight difference between a fine-tuned model $\theta' T$ and its pre-trained counterpart $\theta^{(0)}$ on task T:

$$\Delta\theta_T = \theta'_T - \theta^{(0)}. \tag{6}$$

Unlearning is achieved by subtracting this vector:

$$\theta = \theta^{(0)} - \lambda\Delta\theta_T, \quad \lambda > 0. \tag{7}$$

The method is parameter-efficient and theoretically effective when the unlearned task is either irrelevant or contradictory to the retained tasks.

Kim et al. [36] aggregate multiple task vectors from models fine-tuned with diverse hyperparameters. Instead of selecting a single task vector, it merges only the elements with consistent signs across all task vectors—presumed to reflect forget-related knowledge—while masking conflicting elements. This merged vector is then subtracted from the original model to perform unlearning.

6. Suppression-intended unlearning

Thoroughly removing knowledge from a model remains a challenge task. Technically, true removal often requires modifying all parameters, which is difficult to scale and may harm utility. Conceptually, fully and irreversibly erasing the forgetting data is hard to guarantee (see Section 7). As a result, suppression-intended methods offer a more pragmatic alternative: instead of eliminating traces of the forgotten data, they aim to prevent the model from generating outputs that could expose sensitive or copyrighted content. While

removal-intended methods—typically using GA—focus on changing model parameters θ, suppression methods can intervene at any stage of the generation pipeline in Eq. 1, including the input x, hidden state $h(x)$, or output logits $l(x)$. This flexibility has led to diverse suppression strategies. In this section, we categorize them based on the component they target, highlighting each method's motivations and its alignment with the properties of that component.

6.1. Full-parameter fine-tuning

Before diving into the different components, we first introduce several full-parameter fine-tuning methods. Although these methods update the entire set of model parameters—similar to removal-intended unlearning—they are possible to be less detrimental to model utility, as they do not rely on reversed training loss. The methods share a key motivation: although GA can reduce the likelihood of generating forgetting data, it does not tell the model what to output after unlearning, which might reduce of model utility. These approaches aim to mitigate such side effects.

6.1.1. Fine-grained probability. Fine-grained probability is closely tied to GA-based methods [39], [40], which push the model to generate nonsensical responses to minimize $\mathcal{L}_{\mathrm{GA}}$ in Eq.4. This also explains the divergent property of GA. We only want the answer to change a little to hide the forgotten data, but GA often leads the model to generate entirely incoherent outputs. To address this, Cha et al. [39] introduce a new Inverted Hinge Loss (IHL):

$$\mathcal{L}_{\mathrm{IHL}}(x, y) = 1 + \mathrm{p}_\theta\left(y_t \mid x, y_{<t}\right) - \max_{v \neq y_t} \mathrm{p}_\theta\left(v \mid x, y_{<t}\right)$$

IHL reduces the target token's probability while promoting its closest alternative token, making unlearning more controlled, efficient, and less damaging to overall model utility. Russinovich et al. [40] propose a forget loss that applies KL-divergence between the original token distribution and a modified distribution with the target token removed:

$$\mathcal{L}_f = \mathrm{D}_{KL}\left(\mathrm{softmax}(l) \,\middle\|\, \mathrm{softmax}(l \backslash x_{\text{target}})\right),$$

where $l \in \mathbb{R}^k$ be the logits of the model's top-k predictions at a target position, and let x_{target} be the top-1 predicted token to be unmemorized. This pushes the model away from memorized outputs while preserving fluency.

6.1.2. Rejection fine-tuning. Rejection fine-tuning trains the model to respond to forgetting data with a fixed rejection template (e.g., "I don't know") [11], [41], [42], [43]. This avoids reversed losses and is based on preference optimization [73].

6.1.3. Incorrect labels. This strategy provides plausible but incorrect responses (e.g., wrong names or facts) [44], [45], [46]. Unlike explicit rejection, it mimics a model unfamiliar with the forgetting data, producing more natural yet misleading outputs that help mask unlearning traces.

6.2. Input space

Unlearning in the input space best preserves model utility, as it avoids modifying model parameters or the inference process—only the input x is altered. This subsection covers approaches based on In-Context Learning (ICL) [47], [74], Retrieval-Augmented Generation (RAG) [48], [49], LLM agent [50], and embedding corruptions [51].

6.2.1. In-Context Unlearning (ICUL). ICUL is a counterpart of ICL. It operates by combining two types of examples: *Label Flipping*, where the labels of target instances are randomly altered, and *Context Padding*, where correctly labeled examples are added to stabilize the prompt. ICUL has good performance and is efficient and compatible with black-box LLMs [17]. Another ICL-based method [47] first fine-tune the model to learn the unlearning task in ICL.

6.2.2. RAG-based unlearning. RAG-based unlearning [48] stores forgetting data in a retrieval system. When a user prompt is received, the system checks if any forgetting content is retrieved. If so, it appends a confidentiality instruction to the retrieved knowledge, prompting the LLM to avoid generating related outputs. Despite adding a retrieval step, this approach preserves model utility well and offers a promising alternative to direct model-level forgetting [49]. It also naturally supports sequential unlearning, accommodating ongoing unlearning requests. To reduce latency, we suggest a parallel design: vanilla generation and retrieval run simultaneously. Since most prompts don't involve forgetting data, the retrieval is empty and their responses can be returned directly. If retrieval is non-empty, a second pass with suppression is triggered—ensuring fast response for benign prompts.

6.2.3. Agent-based unlearning. ALU (Agentic LLM Unlearning) [50] introduces a training-free, model-agnostic approach to LLM unlearning via a four-agent pipeline: the Vanilla Agent generates the initial response, the AuditErase Agent produces multiple redacted versions, the Critic Agent scores these versions for unlearning efficacy and utility, and the Composer Agent synthesizes a final high-quality, sanitized output. This agentic framework supports dynamic unlearning requests, demonstrates strong robustness to adversarial prompts, and scales to thousands of targets without degrading performance. However, ALU incurs more inference-time computation than typical post hoc unlearning methods due to multiple LLM-agent calls.

6.2.4. Embedding Corruption. Beyond modifying the input x, Embedding-Corrupted Prompts is proposed to change its embeddings [51]. It uses a trained prompt classifier to detect whether an input query falls within the forgetting scope, and applies learned perturbations to the prompt embeddings at inference time. These perturbations are optimized offline to align the model's output with that of a retained model that has never seen the forgetting data.

6.3. Hidden representation space

Hidden representations in LLMs encode rich semantic information, offering a broad space for unlearning techniques to explore. We categorize them into three main types: (1) Re-steering hidden representation [15], [52], [53], [54], [55], [56], (2) Editing by Sparse Autoencoder [57], [58], [59], [60], pruning [65], [66] and (3) Additional modules [61], [62], [63], [75].

6.3.1. Re-steering hidden representation. This kind of method suppresses the semantic information of forgetting data in the representation space, especially when such knowledge appears in hidden states.

Representation Misdirection for Unlearning (RMU) [15] is a representative method in this class. It fine-tunes a consecutive Transformer layers to map the forgetting data representation into a random vector. The objective is:

$$\mathcal{L}_{RMU} = E_{x_f \in \mathcal{D}_f} \left\| h_\theta^{(l)}(x_f) - cu \right\|_2^2 + \alpha E_{x_r \in \mathcal{D}_r} \left\| h_\theta^{(l)}(x_r) - h_{\text{ref}}^{(l)}(x_r) \right\|_2^2, \quad (8)$$

where u is a fixed random unit vector sampled from a uniform distribution $\mathcal{U}(0,1)$, which serves as the misdirection target, and c is a coefficient controlling the magnitude of the forget-target representation (i.e., the norm of the vector cu). $h_\theta^{(l)}(x)$ and $h_{\text{ref}}^{(l)}(x)$ are the l-th layer representations of x of unlearned model and reference model (model before unlearning). The first term suppresses the forgetting data by mapping its representation to a random vector, while the second term preserves performance on retaining data.

Dang et al. [52] observe that RMU fails to converge in deeper layers due to larger representation norms. This is because the representation norms increase in deeper layers. Forcing these high-norm vectors toward a fixed small-norm target cu is unnatural and leads to instability, requiring extensive tuning of c. To fix this, they scale the target vector using the original norm of each sample:

$$E_{x_f \in \mathcal{D}_f} \left\| h_\theta^{(l)}(x_f) - \beta \| h_{\text{ref}}^{(l)}(x_f) \| \cdot u \right\|_2^2.$$

This norm-matching approach improves convergence and allows RMU to work effectively across all layers.

Similar to RMU, Shen et al. [53] propose to re-steer the representations of forgetting data into a known refusal region. This region is defined by the representations of prompts that naturally cause the model to refuse, such as harmful or unethical queries (e.g., jailbreak triggers), and fictitious or nonsensical questions (e.g., "*What's the capital of $7&a#!*").

Hu et al. [55] propose a method that combines hidden representation re-steering with gradient conflict mitigation and second-order optimization. It selects minimally entangled parameters using mutual information, separates forget and retain data via contrastive learning, and pushes forgetting data away from the reference model's representations. Gradient projection resolves optimization conflicts, while the Sophia optimizer [31] ensures stable convergence.

6.3.2. Editing by SAE. Recently, Sparse Autoencoder has been used as an important tool in interpretability [76], [77], and are now being explored for unlearning by disentangling and removing features of the forgetting data from the representation space [57], [58], [59], [60].

We begin by introducing how Sparse Autoencoders (SAEs) support interpretability. Given an input x—typically a hidden-layer representation $h^{(l)}(x)$ in LLMs—SAE learns a sparse code z by minimizing the reconstruction loss with a sparsity penalty [78]:

$$\mathcal{L}_{\text{SAE}} = \|\mathrm{x} - \hat{\mathrm{x}}\|_2^2 + \lambda \sum_{i=1}^{d} \rho(z_i), \quad (9)$$

where the encoder maps x to z, and the decoder reconstructs $\hat{\mathrm{x}}$ from z. The regularizer $\rho(z_i)$ (e.g., KL divergence) encourages most neurons to remain inactive, so each neuron in z captures distinct, human-interpretable patterns. Each hidden neuron in z activates only in response to specific, meaningful input features.

Farrell et al. [57] propose to train such an SAE for the model to unlearn with the hidden representation $h^{(l)}(x)$ of l-th layer as the input of SAE. By measuring feature activations on forgetting data (e.g., harmful bio-weapons in WMDP [15]), they identify SAE features z_i^f associated with the forget set—those frequently activated in the forget set but not in the retain set. During inference, unlearning is achieved by suppressing these features, setting $z_i^f = -\mathrm{C}$ ($\mathrm{C} > 0$). The modified vector is then decoded into $\hat{h}^{(l)}(x)'$ to replace the original representation, weakening the model's ability to generate related content.

Khoriaty et al. [58] further find a "refusal feature" that can safely return a refusal message. Li et al. [59] extend this method to vision-language model.

6.3.3. Additional modules. This type of method typically freezes the original model parameters and introduces a plug-and-play module to alter the representation. LoRA [64] is a common approach in suppression-intended unlearning. Regardless of whether the loss is removal- or suppression-intended, LoRA freezes the base model, meaning it cannot truly erase learned knowledge. Beyond LoRA, several lightweight modules have been proposed specifically for LLM unlearning. These modules add minimal inference overhead while effectively suppressing the influence of forgetting data. Below, we introduce four such approaches.

Chen et al. [61] propose a method that inserts lightweight unlearning layers into transformers. These layers are trained using an objective that preserves performance on retained data while forgetting target data using KD divergence. The forgetting loss is:

$$\mathcal{L}_f = -\underset{(x,y)\sim\mathcal{D}_f}{E} \sum_{i=1}^{|y|} \left[\mathrm{D}_{KL}\left(\mathrm{p}_\theta(\cdot \mid x, y_{<i}) \,\|\, \mathrm{p}_{\theta,l}(\cdot \mid x, y_{<i}) \right) \right],$$

where $\mathrm{p}_{\theta,l}$ is the next-token output probability distribution of the model with unlearning layers. This method can efficiently handle sequential deletion by merging multiple

unlearning layers into one via linear regression, enabling scalable and composable unlearning without accessing the original training data.

Ren et al. [62] propose a Gated Representation Unlearning (GRUN) which uses a gate function to control the unlearning strength. GRUN is based on Representation Fine-tuning (ReFT) by adding a gate on ReFT. ReFT changes the l-th layer's output representation in a residual form. The changed representation is

$$h^{(l)}_{\text{new}}(x) = h^{(l)}(x) + \phi\left(h^{(l)}(x)\right), \tag{10}$$

where ϕ is a trainable low-rank linear transformation. GRUN uses a soft gate function g to control the unlearning strength:

$$h^{(l)}_{\text{new}}(x) = h^{(l)}(x) + g\left(h^{(l)}(x)\right)\phi\left(h^{(l)}(x)\right), \tag{11}$$

where g is a single-output regression model (linear regression or Multi-Layer Perceptron neural network) with a sigmoid function following the output. The advantage of GRUN is that the soft gate function can be close to zero (i.e. $h^{(l)}_{\text{new}}(x) \approx h^{(l)}(x)$) when retaining data is input, which large preserve the model utility.

Gao et al. [63] use a similar idea for LoRA. They use a detector to measure the similarity between input and unlearning data. The similarity score is used to control the loading of LoRA and then the strength of unlearning.

6.3.4. Localization. A few works propose to localize the parameters related to forgetting data [79], [80] and then unlearning by methods such as neural network pruning identifies neurons [65], [66]. However, a recent paper challenges the core assumption of localized unlearning [81]—that effective knowledge removal in LLMs requires updates to specific parameter regions. Through controlled experiments, the authors demonstrate that unlearning can be achieved equally well by modifying randomly selected regions, suggesting that no fixed set of parameters is necessary for forgetting. These findings call into question the causal link between parameter localization and unlearning success.

6.4. Output space

In this section, we introduc the methods focusing on output space which contains the modification on the logits and retrieval-based methods.

6.4.1. Logits difference. Huang et al. [16] propose δ-unlearning, a method that achieves unlearning by modifying the output logits. It leverages a pair of small, white-box models—identically initialized—to compute a logit offset, defined as the difference in their output logits for the same input. One model is fine-tuned with an unlearning objective (e.g., GA), while the other remains frozen. At inference time, the learned offset is added to the logits of the gray-box LLM, effectively steering its output away from sensitive content. This approach enables plug-and-play unlearning, but it requires the white-box models to share the same tokenizer as the target model and to have access to its output logits for modification.

Ji et al. [67] propose an unlearning framework that inverts the usual forget-retain objective. Instead of modifying the target LLM directly, they train an assistant model to memorize forget data and forget retain data. Unlearning is achieved by subtracting the assistant's logits from the target's:

$$l_f(y \mid x) = l_{\text{target}}(y \mid x) - \alpha \cdot l_{\text{assistant}}(y \mid x)$$

When the target model and assistant model both assign high probability to a token (e.g., a correct answer from the forgetting data), the subtraction significantly reduces its score—effectively suppressing it. When the assistant model assigns low or uniform scores (as trained) on retain data, the subtraction has minimal effect—preserving the target model's original predictions. This contrastive behavior cancels the knowledge learned from specific forgetting data without changing the original LLM's parameters.

6.4.2. Retrieval-based. This method is similar to RAG-based unlearning, but with a key difference: the retrieved content is used to block the generation of specific answer tokens [68]. The method first employs an MLP classifier to determine whether an input prompt targets forgertting data. If so, it retrieves the most semantically similar answer from the forgetting set and extracts key phrases as forbidden tokens. During decoding, it dynamically penalizes or blocks candidate tokens using both token-level hard matching and soft semantic matching, effectively suppressing forgotten content while preserving model fluency and utility.

7. Do the removal-intended methods truly remove forgetting data?

In the proposed taxonomy, removal-intended methods—primarily GA-based approaches—directly modify the model parameters θ to erase the impact of the forgetting data. Their unlearning effectiveness has been shown in various studies [13], [14] and benchmarks [11], [22], where unlearned models successfully avoid reproducing the forgetting content in generated responses.

However, no theoretical framework currently guarantees that GA-based methods can truly remove the influence of data from LLMs, nor is there direct empirical evidence supporting this claim. The only guidance is the initial intuition of GA: *by fine-tuning with a reversed training loss of LLM, GA can negate the training influence of* $\mathcal{D}_f$. Recently, this assumption has been increasingly questioned. Several studies suggest that GA-based methods may only reflect superficial changes in model behaviors instead of truly removal [82], [83], raising an important question: *do these methods truly remove forgetting data?* Although GA-based methods are designed to achieve the true removal, emerging evidence indicates that they may, in practice, only perform suppression [62], [82].

In this section, we first discuss the evidences verifying the existence of this problem (Section 7.1). Then we show

a new understanding perspective to explain the mechanism of GA-based methods and why they do not truly remove (Section 7.2). Finally, we discuss the open problems: Can we truly remove? What is the challenge? Do we need true removal or is behavioral suppression sufficient? (Section 7.3)

7.1. Failures of removal-intended unlearning

Recent research examining the capabilities of GA-based unlearning reveals two key insights. First, unlearned models can still produce outputs related to the forgetting data under adversarial attacks [84], [85], [86], suggesting that the removal may be superficial and fails to eliminate deeper representations of the forgotten knowledge. Second, unlearned models remain highly vulnerable to subsequent fine-tuning: even without access to the forgetting content, these models can re-memorize it [87], [88]. This indicates that hidden patterns associated with the forgetting data may still persist—temporarily suppressed by unlearning but easily reactivated through additional training. In this subsection, we expand on these two observations.

7.1.1. Adversarial attacks. GA performs well on the widely used QA benchmark TOFU [11]. When prompted with a question x from the forgetting set $\mathcal{D}_f$, the unlearned LLM typically hallucinates the answer rather than recalling the correct response. However, Schwarzschild et al. [84] observe that if the model is instead prompted with a subsequence of x, the GA-unlearned model can still generate the correct answer y. This suggests that the model may memorize and suppress the exact full sequence of x, while partial cues still activate the forgotten knowledge. Similarly, Yuan et al. [85] apply the GCG attack [89] to optimize adversarial suffixes and find that 54% of the forgetting knowledge can be recovered—despite having no access to model parameters. To et al. [86] also use the GCG attack to assess whether the model has truly forgotten the data. They find that larger LLMs are more vulnerable to knowledge extraction attacks. The authors further evaluate Task Arithmetic, another removal-intended method, and find it poses even higher risk of forgetting data recovery than GA-based methods. This may be because model editing via task arithmetic is more aggressive, and the need to constrain the editing magnitude to preserve utility reduces its unlearning effectiveness.

To provide more evidence, Hong et al. [90] design a series of activation patching and parameter restoration experiments. They find that fine-tuning-based unlearning mainly manipulates behavior through output-layer coefficients, rather than actually removing stored knowledge.

In addition, Soft token attacks are proposed to bypass safety alignment and unlearning in open-source LLMs by directly perturbing the continuous input embeddings rather than discrete tokens [91], [92], [93]. For example, Schwinn et al. [91] add a adversarial embedding e_i^{adv} after a sequence x as its embeddings. It is optimized to minimize the cross-entropy loss between the model's output and x's groundtruth response:

$$\mathcal{L}_{\text{CE}}\left(f\left(x \| e_i^{adv}\right), y_i\right) \tag{12}$$

Experiments show this approach outperforms prior discrete attacks in success rate, efficiency, and its ability to recover supposedly deleted or pretraining data.

Interestingly, Chen et al. [94] find that soft token attacks cannot reliably audit unlearning. This attack is too powerful and can generate the knowledge even the model has never seen. Soft token attacks function more like an extreme case of prompt tuning. It optimizes the embedding space to enable the model to output nearly arbitrary sequences. This means the forgetting knowledge is injected by the adversarial embeddings and lead to false positive.

7.1.2. Fine-tuning after unlearning. It is found that the unlearned model might recover the forgetting knowledge after fine-tuning with other data [87], [88]. [87] reveals that models can easily recover supposedly unlearned information through small-scale finetuning on related but benign data—highlighting that unlearning is often superficial. [88] introduces a more rigorous evaluation framework, showing that state-of-the-art unlearning methods often fail to remove information from model weights, as the forgotten knowledge can be recovered through retraining on independent facts. These works demonstrate that existing unlearning approaches tend to hide knowledge rather than truly erase.

Takeaway. GA-based unlearning is often superficial and often fails to truly remove: forgetting knowledge can resurface under adversarial prompts or reappear after benign fine-tuning, revealing that current methods suppress rather than erase internal representations.

7.2. A new understanding: removal-intended methods are actually doing suppression

Based on the above observations, researchers have begun to realize that simply applying a reversed loss may not truly eliminate the training traces of the forgetting data. This has prompted a search for a deeper understanding of how GA-based methods actually operate. In this subsection, we present two key phenomena—*over-generalization* and *syntactic similarity*—that inspire a new understanding of GA-based unlearning. We analyze these phenomena and discuss how they contribute to a revised understanding of the underlying mechanisms of GA-based methods.

From over-generalization to unlearning signals. People find that the unlearning will **over-generalize** to other data that is related to forgetting knowledge [95]. When models are trained to forget certain knowledge (e.g., theft), they also implicitly forget similar contents (e.g., bomb-making). In the beginning, people connect this overgeneralization with the explanation of the reduction of utility (collateral damage) [11], [96], [97]. If GA-based methods truly negate

the training influence, over-generalization would suggest they also suppress related knowledge. However, since GA often only hides the forget data, over-generalization instead implies that these methods may unintentionally suppress any data similar to the forgetting data.

While the above seems just a problem of unclear decision boundary, Thaker et al. put forward a novel perspective in their position paper [82]. They find that when adding a question of forgetting data before/after a question of retaining data, the unlearned would be unable to answer the retaining data, too. This is a novel evaluation method because previous works only test the forgetting and retaining data separately, but they do it simultaneously.

Based on this observation, the unlearning mechanism of GA is further explained as the unlearning signal by [62]. In [62], the authors observe some properties in the representation space $h(x)$. They provide three sub-sets: "forgetting" data, "retaining" data and "never-seen" data. They have similar representations because they are all synthetic data sampled from the same distribution of TOFU benchmark (details of TOFU will be discussed in Section 8.2). However, after unlearning, in the representation space, the clusters of "retaining" data and "never-seen" data are still close, while the cluster of "forgetting" data is far from them. This means unlearned LLM still recognize the forgetting data. The unlearned model does not forget it, but distinguish it from other data. The authors further find that better unlearning effectiveness is likely to be associated with better distinction. Lastly, they also add forgetting data questions into normal (retaining/never-seen) data questions and find that once mixed with target data, the representations of normal data is dominated by forgetting data (which is pulled toward the distinct cluster of forgetting data). Consequently, the model's ability to answer normal questions deteriorates. As the authors conclude, this implies that, instead of removing the forgetting data, GA-unlearned models treat it as a unlearning signal to suppress the generation. They behave like unlearning once there is a unlearning signal in the prompt. Worth to noted that this is actually the same as what suppression-intend unlearning is doing.

The previous discussion provides a refined understanding of the underlying behavior of GA-based unlearning methods. However, a critical question remains: why are these methods inherently limited in their effectiveness? The following paragraphs offer preliminary explanations based on syntactic similarity and related studies.

From syntactic similarity to true intention. The second phenomenon is the role of syntactic similarity, which is highly related to the selective forgetting in GA-based methods in Section 5.1.6. This phenomenon reflects the limited intention behind GA loss. Specifically, Chang et al [98] find that syntactic similarity plays a critical role in maintaining model utility. Syntactically similar means having the same sentence structure or grammatical pattern. For example, "*Who is the author of Watermelon on the Moon?*" and "*Who is the author of Attention is all you need?*" are syntactically similar (the same sentence structure and different literature titles). They find that more similar syntactic between forgetting data and retaining data would have a better model utility. When GA optimizes forgetting loss on the benign tokens (the tokens that are not directly related to the forgetting data such as "*Who is the author*"), these benign tokens are also calculated in retaining loss, which relieves the destroy on the model utility for benign tokens.

This indicates that GA does not really negate the knowledge. As shown by [99] if we look at the GA loss (Eq. 4), what the objective actually does is: **if x is in the prompt, increase its output error**. GA loss would not distinguish benign tokens or forgetting tokens (tokens that are directly related to forgetting data). The Gradient Ascent on benign tokens are meaningless and even harmful for the model. This explains the phenomenon of syntactic similarity. If there are corresponding syntactically similar tokens in retaining data, the destruction on benign tokens can be make up to some extent. This also explains the mechanist of unlearning signal: if forgetting data is in the prompt, increase its output error (i.e., behave like know nothing about it).

Although GA intends to remove the knowledge from LLMs, it actually does similar things as suppression-intended unlearning. However, we should not ignore the significant contribution of GA at this stage of LLM unlearning. Although people haven't fine a better way to truly remove the knowledge, we believe GA will be the basic for the future work. For example, as we mentioned in Section 5.1.6, selective forgetting has been proposed to explore a correct way to using GA.

Takeaway. GA-based unlearning does not truly remove knowledge; instead, it treats the forgetting data as a signal to trigger unlearning behavior. When this signal is detected, the model simulates ignorance. This aligns with the design of GA loss, which increases the model's output error when presented with the forgetting data.

7.3. Open problems

After reviewing and discussing the existing research in removal-intended unlearning, several important questions remain unresolved. In this subsection, we identify some key open problems.

Can we truly remove the influence of forgetting data? The fundamental goal of removal-intended unlearning is to erase the influence of specific data $\mathcal{D}_f$ from model parameters θ. However, no theoretical framework currently guarantees such removal in LLMs, nor are there conclusive empirical methods to verify it. The key challenge is the entangled, non-linear way in which knowledge is stored across millions or billions of parameters in transformer-based models. Simply reversing the loss signal, as done in GA, may not undo the distributed effects of training.

To achieve true removal, we may need new theoretical tools to (1) trace the specific parameter regions influenced by $\mathcal{D}_f$ and (2) isolate and invert those changes without degrading overall utility. This will likely require advances

in interpretability, causal attribution, or model editing beyond current fine-tuning paradigms. Moreover, new benchmarks and evaluation protocols are needed to distinguish true removal from behavioral suppression, especially under adversarial or compositional prompts.

Do we need true removal, or is behavioral suppression sufficient? From a practical standpoint—especially in commercial or production-level applications—strictly enforcing true removal may not always be necessary. In many deployment scenarios, behavioral suppression may be sufficient as long as the model no longer reproduces the target knowledge under expected usage conditions. Instead of pursuing theoretical perfection, practitioners should focus on whether the model meets practical requirements. If suppression-intended unlearning aligns with practical requirements, it can even be prioritized over true removal.

However, this should not discourage ongoing research into true removal. The pursuit of actual knowledge removal remains scientifically valuable, as it can inspire deeper understanding of fine-tuning, alignment, interpretability, and model editing. Developing mechanisms to trace, attribute, and eliminate specific knowledge in LLMs can open up broader research directions, including modular training, causal reasoning, and long-term memory control. Therefore, while suppression may be acceptable in practice, removal should remain a central research goal.

8. Evaluation metrics and benchmarks

Evaluation plays a critical role in the development of LLM unlearning. A reasonable and accurate evaluation can help researchers design more effective methods. At a high level, unlearning evaluation encompasses two key aspects: *unlearning effectiveness* and *model utility*. Numerous metrics have been proposed, and several benchmarks have been established to assess these dimensions. In this section, we review existing metrics and benchmarks, and discuss their limitations along with potential future directions.

8.1. A review of existing metrics

Unlearning effectiveness measures whether a model continue to output or leak information related to the forgetting data and knowledge, while model utility evaluates the extent to which the model retains its normal functionality on non-forgotten tasks. This subsection first introduces three general metrics applicable to both aspects, followed by metrics specifically tailored to unlearning effectiveness.

8.1.1. General metrics. Traditional metrics for generation quality and accuracy can be used to measure both unlearning and utility, we first present this kind of metrics below:

- **ROUGE Recall.** [11], [22], [100] measures the overlap between the model's generated answer and the ground truth at the word level. Specifically, it evaluates how much of the ground truth answer is recalled in the output. The precision of ROUGE is usually not used because the generation usually contains additional or auxiliary content beyond the ground truth answer, which should not influence the measurement.
- **Probability.** This metric [11], [62], [100] measures the likelihood that the model reproduces the ground truth answer, typically evaluated token by token.

$$Prob(y \mid x) = \frac{1}{T} \sum_{t=1}^{T} \mathrm{p}\left(y_t \mid x, y_{<t}\right)$$

  A high value indicates that the model still assigns high confidence to the answer.
- **Multi-choice accuracy.** [15], [54], [62] evaluates whether the model can correctly select the ground truth answer from a set of candidate options. It reflects the model's ability to distinguish the correct answer from distractors in a multiple-choice setting.

8.1.2. Unlearning effectiveness metrics. Some metrics have been proposed or adapted to evaluate unlearning effectiveness. We begin by introducing two categories of methods that uses existing tools:

- **Membership Inference Attacks (MIA)** are used by some benchmarks to evaluate the privacy leakage [22], [101], [102]. If the training data can be successfully inferred, it means the unlearning effectiveness may be not resistant. The common MIA includes LOSS [103], Zlib Entropy [104], Min-K% Prob [105] and so on.
- **Robustness** is an important evaluation aspect. As discussed in Section 7.1, it serves as a key tool for exposing superficial unlearning behaviors. A common approach involves using adversarial prompts to test the robustness of the model [85], [86], [101].

In addition to these two methods that leverage existing tools, several new tools have also been developed to measure unlearning effectiveness:

- **Watermark.** Lu et al. [106] propose to embed imperceptible, owner-specific watermarks into training data and verifies their presence in model outputs after unlearning.
- **Gradient Effect.** Want et al. [107] propose Gradient Effect that quantifies how an unlearning objective impacts model performance via gradient alignment. Specifically, it computes the dot product between the gradients of the unlearning objective $\mathcal{L}_{\mathrm{u}}$ and the risk function $\mathcal{R}$.
- **Unlearning Shapley**. Ma et al. [108] propose a novel data valuation framework that combines machine unlearning with Shapley value theory [109]. It measures the performance drop after unlearning target data from a pretrained model to estimate its value.
- **Representation-level.** Xu et al. [110] introduce a set of representation-level evaluation metrics for unlearning. These metrics go beyond token-level accuracy or perplexity to assess whether the model's internal feature representations are genuinely altered or merely superficially perturbed.

8.2. A review of existing benchmarks

Benchmarks are essential for evaluating unlearning methods, offering controlled and synthetic settings for fair comparison.A critical aspect is the construction of forgetting data, which directly affects evaluation validity. As unlearning methods continue to evolve, several benchmarks have been proposed to match this progress. In this section, we introduce a few representative and widely used benchmarks.

TOFU [11]. TOFU uses a synthetic dataset of fake books and authors. The data corpus is a set of QA pairs. Since no LLM has trained on the synthetic dataset, the tested model has to be fine-tuned to learn from the dataset first. Then the synthetic dataset is separated into forgetting data and retaining data. The utility is tested by three QA sets in TOFU: retaining data, and two real QA sets (real authors and world knowledge). The advantage of fine-tuning is easy to control, and the new knowledge has not much entanglement with real data. A few new benchmarks are also proposed based on it like PerMU focusing on robustness [111] and R-TOFU focusing reasoning [112].

MUSE [22]. MUSE has a similar pipeline as TOFU. It first fine-tunes LLMs to learn forgetting data. But it does not use synthetic datasets. It uses a dataset that is not in the training data of the tested LLMs. The advantage of this benchmark is that it provides a more comprehensive metrics, especially the test on scalability and sequential unlearning.

WMDP [15]. WMDP does not fine-tune to learn new knowledge. It tests the unlearning of harmful bio and chemical knowledge. It uses LLMU [113] and MT-Bench [114] to test the model utility across a broad spectrum of subjects. A multi-choice format is used to test the unlearning effectiveness and utility.

RWKU [101]. RWKU targets 200 real-world famous entities and rigorously evaluates unlearning effectiveness using fill-in-the-blank, question-answer, and nine types of adversarial probes. It also assesses side effects on neighboring knowledge and general utility.

In summary, the choice of benchmark significantly influences how unlearning methods are evaluated, but no single benchmark is inherently superior to others. Each focuses on different aspects—such as control, realism, scalability, or robustness—and over-reliance on a specific benchmark may lead to models being optimized for test performance rather than real-world forgetting objectives. Therefore, it is important to encourage the development of diverse and comprehensive benchmarks to ensure a more robust and generalizable assessment of unlearning methods.

8.3. Toward better evaluations: challenges and considerations

To foster more robust and meaningful evaluations, we outline several key considerations that should guide future efforts. These include both methodological principles and practical constraints that are often overlooked in current benchmarking practices.

Fixed and greedy sampling is too narrow. Scholten et al. [115] arguing that standard deterministic (greedy) evaluations fail to capture real-world risks such as information leakage in unlearning and alignment tasks. Although sometimes a model fails to generate a forgotten answer, the model still assigns a high probability to the correct answer, indicating that the knowledge is internally retained [116]. Thus, relying solely on fixed or greedy decoding may underestimate the model's actual knowledge retention; probabilistic evaluation across a distribution of outputs provides a more realistic and robust picture.

Unclear settings: what should the user submit? Current unlearning scenarios are all based on benchmark assumptions, where the data to be unlearned is fixed and clearly defined. However, in real-world applications, what exactly should users provide? Should they specify abstract concepts or concrete corpora? Abstract concepts are often too vague and underspecified, while user-provided corpora lack standardization and may vary in quality. We attempt to answer this question through real-world case studies, but unfortunately, although various unlearning methods have been proposed, there is no commercial product that explicitly offers unlearning capabilities or claims to have integrated unlearning techniques. As a result, there is still no practical and reliable answer to this question.

Fine-tuning knowledge vs. pre-training knowledge. For TOFU and MUSE, the model first fine-tunes to learn new knowledge. The new knowledge is separate and easiser to control than pre-training knowledge. However, fine-tuning would also bring two disadvantages. The first is that fine-tuning suffers from catastrophic forgetting [117]. When unlearning on the fine-tuned models, the reduce of the ability in answering forgetting data might be somehow attributed to catastrophic forgetting, making it hard to isolate the true effect of unlearning. The second is that fine-tuning on a small synthetic dataset would lead to overfitting and will cause the reduction of utility on two real QA sets [11]. Thus, the reduction of utility bring by unlearning is hard to quantify.

Core-set vs. complete set. Patil et al. [96] propose that unlearning the whole forgetting set might be one of reason of the reduction in model utility. Thus, they identify and prune 10% to 30% outliers from the forget set. Pal at al. [118] further study this core-set effect. The authors find that even a randomly selected 5% subset can result in comparable unlearning performance to using the full dataset. These findings indicate that existing evaluations may overestimate the necessity of forgetting the entire dataset. Thus, future unlearning benchmarks should consider core-set-aware evaluation protocols to more realistically assess unlearning effectiveness and efficiency.

What is a reasonable retaining set? Syntactic similarity has already been shown to affect utility degradation—when the retaining set is more similar to the forgetting set, the utility drop tends to be smaller [98]. Some benchmarks use idealized retaining sets; for example, TOFU selects retaining data that is highly similar to the forgetting data. However, such assumptions are unrealistic in real-

world scenarios, where we often do not know exactly what data the user wants to unlearn in advance, making it difficult to provide such well-matched retaining sets. Therefore, benchmark design should take this limitation into account when selecting the retaining set.

Overlooked utility evaluation. Utility evaluations based solely on unlearning benchmarks may fail to reflect the actual performance of the LLM. In practice, every LLM undergoes internal performance testing before being released or deployed, and utility should be assessed using these evaluation suites. For example, ChatGPT has its own benchmark suite [119], and unlearning should be considered acceptable only if the model's performance on these internal tasks remains stable. However, current unlearning benchmarks often use their own utility criteria, which may not align with real-world expectations.

9. Other gaps in existing unlearning

To enable unlearning techniques to be truly adopted in real-world applications, this section discusses the key gaps that still hinder practical deployment. Some of these challenges remain open problems and have yet to receive sufficient attention from the community. We highlight these issues in the hope that future research will address them, ultimately bridging the gap between unlearning research and its real-world implementation.

Model utility. The harm to model utility is always the most critical obstacle to deployment. Let's think from the perspective of a model builder: if unlearning did not bring any negative impact on model performance, who would mind adding such a beneficial feature to their LLM? Therefore, regardless of the unlearning effectiveness, model utility must be preserved to give model builders confidence in using it. Thus, both maintaining and verifying utility are of paramount importance.

Sequential unlearning. While preserving utility is a well-recognized goal in unlearning, one particularly challenging setting is sequential unlearning, where utility degradation becomes more severe and difficult to mitigate. In real-world applications, users are likely to submit new unlearning requests continuously, and most existing methods suffer from accumulated reductions in model utility as a result.

There are currently two main ideas for addressing this challenge. One approach is to avoid sequential unlearning altogether and instead perform unlearning on the entire set of forgetting data each time. The coreset effect suggests that scalability is promising—meaning we may not need to process the full dataset to achieve good forgetting performance. However, if we persist with sequential unlearning and unlearn one subset at a time, the utility degradation quickly becomes significant. For example, even if each step results in just a 3% drop in utility, after 20 iterations, the model could lose up to 45% of its utility. In practice, users are likely to submit far more than 20 unlearning requests, and in fact, the utility loss per step is often much greater than 3% [11], [22], especially for methods that directly modify model parameters or internal representations.

An alternative and promising direction is to base unlearning on RAG as we discussed in Section 6.2.2. New requests only need to update the retrieval system.

Scalability of model size. Speaking of scalability, another important aspect to consider is the scalability with respect to model size. Most current studies conduct experiments on relatively small models, such as 7B or 8B parameters [11], [15], [101]. However, for real-world deployment, it is essential to evaluate unlearning methods on much larger models. This shift introduces entirely new challenges, both in terms of computational resources and in the design of scalable and efficient unlearning methods that can handle the complexity of large-scale LLMs.

Poisoning risks in unlearning submissions. If the unlearning service accepts user-provided corpora, as discussed in Section 8.3, the quality of such data is often difficult to guarantee. Taking this a step further, what if someone intentionally injects poisoned data into the unlearning request? How can we ensure that the data is clean and trustworthy? In fact, recent work has demonstrated that it is possible to poison the unlearning data to deliberately degrade the model's utility [99]. Therefore, ensuring that user-provided corpora are both safe and standards-compliant is of critical importance.

Interference Between Different Training and Inference Stages. Can unlearning interfere with other training processes such as RLHF? It is currently assumed that unlearning is performed only after the model has been fully trained, but does this risk undermining the alignment achieved through RLHF? Many studies suggest that alignment is rather superficial [120], [121]. If further RLHF tuning is required after unlearning, it may compromise the intended purpose of unlearning. Similarly, if downstream tasks are introduced after unlearning, would that diminish the effectiveness of unlearning? Moreover, if quantization is applied during deployment, the effectiveness may also be reduced [122]—this is another factor that needs to be considered.

10. Conclusion

Machine unlearning has emerged as a critical technique to address growing concerns around privacy, copyright, and regulatory compliance in LLMs. This paper presents a structured overview of unlearning in LLMs. While prior work focuses on technical methods, we propose a new taxonomy based on intentions: removal vs. suppression. Beyond taxonomy, we make three core contributions: (1) we examine whether popular removal-intended methods like GA can truly erase knowledge, revealing both theoretical and empirical limitations; (2) we survey evaluation strategies, identifying key shortcomings in current metrics and benchmarks; and (3) we highlight open challenges for real-world deployment, including usability, side-effect mitigation, and verification. This study aims to provide a principled foundation for future research and practical deployment of unlearning in LLMs.

References

[1] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee *et al.*, "Natural questions: a benchmark for question answering research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453–466, 2019.

[2] F. Stahlberg, "Neural machine translation: A review," *Journal of Artificial Intelligence Research*, vol. 69, pp. 343–418, 2020.

[3] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert systems with applications*, vol. 165, p. 113679, 2021.

[4] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky, "Deep reinforcement learning for dialogue generation," *arXiv preprint arXiv:1606.01541*, 2016.

[5] P. Hacker, A. Engel, and M. Mauer, "Regulating chatgpt and other large generative ai models," in *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, 2023, pp. 1112–1123.

[6] N. Lucchi, "Chatgpt: a case study on copyright challenges for generative artificial intelligence systems," *European Journal of Risk Regulation*, vol. 15, no. 3, pp. 602–624, 2024.

[7] M. M. Grynbaum and R. Mac. (2023) The times sues openai and microsoft over a.i. use of copyrighted work. [Online]. Available: https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html

[8] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, no. 3152676, pp. 10–5555, 2017.

[9] J. Rosen, "The right to be forgotten," *Stan. L. Rev. Online*, vol. 64, p. 88, 2011.

[10] S. Liu, Y. Yao, J. Jia, S. Casper, N. Baracaldo, P. Hase, Y. Yao, C. Y. Liu, X. Xu, H. Li *et al.*, "Rethinking machine unlearning for large language models," *arXiv preprint arXiv:2402.08787*, 2024.

[11] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, and J. Z. Kolter, "TOFU: A task of fictitious unlearning for LLMs," in *First Conference on Language Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=B41hNBoWLo

[12] J. Ren, H. Xu, P. He, Y. Cui, S. Zeng, J. Zhang, H. Wen, J. Ding, P. Huang, L. Lyu *et al.*, "Copyright protection in generative ai: A technical perspective," *arXiv preprint arXiv:2402.02333*, 2024.

[13] Y. Yao, X. Xu, and Y. Liu, "Large language model unlearning," *arXiv preprint arXiv:2310.10683*, 2023.

[14] R. Zhang, L. Lin, Y. Bai, and S. Mei, "Negative preference optimization: From catastrophic collapse to effective unlearning," *arXiv preprint arXiv:2404.05868*, 2024.

[15] N. Li, A. Pan, A. Gopal, S. Yue, D. Berrios, A. Gatti, J. D. Li, A.-K. Dombrowski, S. Goel, G. Mukobi *et al.*, "The wmdp benchmark: Measuring and reducing malicious use with unlearning," in *Forty-first International Conference on Machine Learning*, 2024.

[16] J. Y. Huang, W. Zhou, F. Wang, F. Morstatter, S. Zhang, H. Poon, and M. Chen, "Offset unlearning for large language models," *arXiv preprint arXiv:2404.11045*, 2024.

[17] M. Pawelczyk, S. Neel, and H. Lakkaraju, "In-context unlearning: Language models as few shot unlearners," *arXiv preprint arXiv:2310.07579*, 2023.

[18] J. R. B. Jr., "Executive order 14110: Safe, secure, and trustworthy development and use of artificial intelligence," https://www.federalregister.gov/documents/2023/11/01/2023-24283/safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence, October 2023, 88 Fed. Reg. 75191.

[19] "Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence (artificial intelligence act)," https://eur-lex.europa.eu/eli/reg/2024/1689/oj, 2024, article 53.

[20] "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (general data protection regulation)," https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2016, article 17.

[21] J. Jang, D. Yoon, S. Yang, S. Cha, M. Lee, L. Logeswaran, and M. Seo, "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)*, 2023, pp. 14 389–14 408.

[22] W. Shi, J. Lee, Y. Huang, S. Malladi, J. Zhao, A. Holtzman, D. Liu, L. Zettlemoyer, N. A. Smith, and C. Zhang, "Muse: Machine unlearning six-way evaluation for language models," *arXiv preprint arXiv:2407.06460*, 2024.

[23] J. Yao, E. Chien, M. Du, X. Niu, T. Wang, Z. Cheng, and X. Yue, "Machine unlearning of pre-trained large language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 8403–8419.

[24] A. K. Veldanda, S.-X. Zhang, A. Das, S. Chakraborty, S. Rawls, S. Sahu, and M. Naphade, "Llm surgery: Efficient knowledge unlearning and editing in large language models," *arXiv preprint arXiv:2409.13054*, 2024.

[25] C. Fan, J. Liu, L. Lin, J. Jia, R. Zhang, S. Mei, and S. Liu, "Simplicity prevails: Rethinking negative preference optimization for llm unlearning," *arXiv preprint arXiv:2410.07163*, 2024.

[26] Z. Bu, X. Jin, B. Vinzamuri, A. Ramakrishna, K.-W. Chang, V. Cevher, and M. Hong, "Unlearning as multi-task optimization: A normalized gradient difference approach with an adaptive learning rate," *arXiv preprint arXiv:2410.22086*, 2024.

[27] K.-W. Kim, J.-H. Park, J.-M. Han, and S.-W. Lee, "Grail: Gradient-based adaptive unlearning for privacy and copyright in llms," *arXiv preprint arXiv:2504.12681*, 2025.

[28] X. Zhong, H. Luo, and C. Liu, "Dualoptim: Enhancing efficacy and stability in machine unlearning with dual optimizers," *arXiv preprint arXiv:2504.15827*, 2025.

[29] J. Jia, Y. Zhang, Y. Zhang, J. Liu, B. Runwal, J. Diffenderfer, B. Kailkhura, and S. Liu, "Soul: Unlocking the power of second-order optimization for llm unlearning," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 4276–4292.

[30] Z. Huang, X. Cheng, J. Zhang, J. Zheng, H. Wang, Z. He, T. Li, and X. Huang, "A unified gradient-based framework for task-agnostic continual learning-unlearning," *arXiv preprint arXiv:2505.15178*, 2025.

[31] H. Liu, Z. Li, D. L. W. Hall, P. Liang, and T. Ma, "Sophia: A scalable stochastic second-order optimizer for language model pre-training," in *The Twelfth International Conference on Learning Representations*.

[32] G.-O. Bărbulescu and P. Triantafillou, "To each (textual sequence) its own: Improving memorized-data unlearning in large language models," in *International Conference on Machine Learning*. PMLR, 2024, pp. 3003–3023.

[33] X. Feng, C. Chen, Y. Li, and Z. Lin, "Fine-grained pluggable gradient ascent for knowledge unlearning in language models," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 10 141–10 155.

[34] L. Wang, X. Zeng, J. Guo, K.-F. Wong, and G. Gottlob, "Selective forgetting: Advancing machine unlearning techniques and evaluation in language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 1, 2025, pp. 843–851.

[35] G. Ilharco, M. T. Ribeiro, M. Wortsman, S. Gururangan, L. Schmidt, H. Hajishirzi, and A. Farhadi, "Editing models with task arithmetic," *arXiv preprint arXiv:2212.04089*, 2022.

[36] H. Kim, D. Han, and J. Choe, "Negmerge: Consensual weight negation for strong machine unlearning," *arXiv preprint arXiv:2410.05583*, 2024.

[37] H. Li, Y. Zhang, S. Zhang, P.-Y. Chen, S. Liu, and M. Wang, "When is task vector provably effective for model editing? a generalization analysis of nonlinear transformers," in *The Thirteenth International Conference on Learning Representations*.

[38] K. Kuo, A. Setlur, K. Srinivas, A. Raghunathan, and V. Smith, "Exact unlearning of finetuning data via model merging at scale," *arXiv preprint arXiv:2504.04626*, 2025.

[39] S. Cha, S. Cho, D. Hwang, and M. Lee, "Towards robust and parameter-efficient knowledge unlearning for llms," in *The Thirteenth International Conference on Learning Representations*.

[40] M. Russinovich and A. Salem, "Obliviate: Efficient unmemorization for protecting intellectual property in large language models," *arXiv preprint arXiv:2502.15010*, 2025.

[41] Z. Liu, G. Dou, Z. Tan, Y. Tian, and M. Jiang, "Towards safer large language models through machine unlearning," *arXiv preprint arXiv:2402.10058*, 2024.

[42] Y. Ishibashi and H. Shimodaira, "Knowledge sanitization of large language models," *arXiv preprint arXiv:2309.11852*, 2023.

[43] T. Xu, X. Liu, F. Wu, X. Wang, and J. Gao, "Suv: Scalable large language model copyright compliance with regularized selective unlearning," *arXiv preprint arXiv:2503.22948*, 2025.

[44] R. Eldan and M. Russinovich, "Who's harry potter? approximate unlearning in llms," *arXiv preprint arXiv:2310.02238*, 2023.

[45] Y. Liu, Y. Zhang, T. Jaakkola, and S. Chang, "Revisiting who's harry potter: Towards targeted unlearning from a causal intervention perspective," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 8708–8731.

[46] H. Xu, N. Zhao, L. Yang, S. Zhao, S. Deng, M. Wang, B. Hooi, N. Oo, H. Chen, and N. Zhang, "Relearn: Unlearning via learning for large language models," *arXiv preprint arXiv:2502.11190*, 2025.

[47] S. Takashiro, T. Kojima, A. Gambardella, Q. Cao, Y. Iwasawa, and Y. Matsuo, "Answer when needed, forget when not: Language models pretend to forget via in-context knowledge unlearning," *arXiv preprint arXiv:2410.00382*, 2024.

[48] S. Wang, T. Zhu, D. Ye, and W. Zhou, "When machine unlearning meets retrieval-augmented generation (rag): Keep secret or forget knowledge?" *arXiv preprint arXiv:2410.15267*, 2024.

[49] S. Vilella and G. Ruffo, "(de)-indexing and the right to be forgotten," *arXiv preprint arXiv:2501.03989*, 2025.

[50] D. Sanyal and M. Mandal, "Alu: Agentic llm unlearning," *arXiv preprint arXiv:2502.00406*, 2025.

[51] C. Y. Liu, Y. Wang, J. Flanigan, and Y. Liu, "Large language model unlearning via embedding-corrupted prompts," *arXiv preprint arXiv:2406.07933*, 2024.

[52] D. Huu-Tien, T.-T. Pham, H. Thanh-Tung, and N. Inoue, "On effects of steering latent representation for large language model unlearning," *arXiv preprint arXiv:2408.06223*, 2024.

[53] W. F. Shen, X. Qiu, M. Kurmanji, A. Iacob, L. Sani, Y. Chen, N. Cancedda, and N. D. Lane, "Lunar: Llm unlearning via neural activation redirection," *arXiv preprint arXiv:2502.07218*, 2025.

[54] D. Huu-Tien, H. Thanh-Tung, L.-M. Nguyen, and N. Inoue, "Improving the robustness of representation misdirection for large language model unlearning," *arXiv preprint arXiv:2501.19202*, 2025.

[55] J. Hu, Z. Huang, X. Yin, W. Ruan, G. Cheng, Y. Dong, and X. Huang, "Falcon: Fine-grained activation manipulation by contrastive orthogonal unalignment for large language model," *arXiv preprint arXiv:2502.01472*, 2025.

[56] Y. Wang, R. Wu, Z. He, X. Chen, and J. McAuley, "Large scale knowledge washing," *arXiv preprint arXiv:2405.16720*, 2024.

[57] E. Farrell, Y.-T. Lau, and A. Conmy, "Applying sparse autoencoders to unlearn knowledge in language models," in *Neurips Safe Generative AI Workshop 2024*.

[58] M. Khoriaty, A. Shportko, G. Mercier, and Z. Wood-Doughty, "Don't forget it! conditional sparse autoencoder clamping works for unlearning," *arXiv preprint arXiv:2503.11127*, 2025.

[59] Q. Li, J. Geng, D. Zhu, F. Cai, C. Lyu, and F. Karray, "Sauce: Selective concept unlearning in vision-language models with sparse autoencoders," *arXiv preprint arXiv:2503.14530*, 2025.

[60] A. Muhamed, J. Bonato, M. Diab, and V. Smith, "Saes can improve unlearning: Dynamic sparse autoencoder guardrails for precision unlearning in llms," *arXiv preprint arXiv:2504.08192*, 2025.

[61] J. Chen and D. Yang, "Unlearn what you want to forget: Efficient unlearning for llms," in *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[62] J. Ren, Z. Dai, X. Tang, H. Liu, J. Zeng, Z. Li, R. Goutam, S. Wang, Y. Xing, Q. He, and H. Liu, "A general framework to enhance fine-tuning-based llm unlearning," *arXiv preprint arXiv:2502.17823*, 2025.

[63] C. Gao, L. Wang, K. Ding, C. Weng, X. Wang, and Q. Zhu, "On large language model continual unlearning," in *The Thirteenth International Conference on Learning Representations*, 2025.

[64] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[65] N. Pochinkov and N. Schoots, "Dissecting language models: Machine unlearning via selective pruning," *arXiv preprint arXiv:2403.01267*, 2024.

[66] Z. Liu, G. Dou, X. Yuan, C. Zhang, Z. Tan, and M. Jiang, "Modality-aware neuron pruning for unlearning in multimodal large language models," *arXiv preprint arXiv:2502.15910*, 2025.

[67] J. Ji, Y. Liu, Y. Zhang, G. Liu, R. R. Kompella, S. Liu, and S. Chang, "Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference," *arXiv preprint arXiv:2406.08607*, 2024.

[68] Z. Deng, C. Y. Liu, Z. Pang, X. He, L. Feng, Q. Xuan, Z. Zhu, and J. Wei, "Guard: Generation-time llm unlearning via adaptive restriction and detection," *arXiv preprint arXiv:2505.13312*, 2025.

[69] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 141–159.

[70] B. Liu, Q. Liu, and P. Stone, "Continual learning and private unlearning," in *Conference on Lifelong Learning Agents*. PMLR, 2022, pp. 243–254.

[71] L. Graves, V. Nagisetty, and V. Ganesh, "Amnesiac machine learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, 2021, pp. 11 516–11 524.

[72] A. Thudi, G. Deza, V. Chandrasekaran, and N. Papernot, "Unrolling sgd: Understanding factors influencing machine unlearning," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2022, pp. 303–319.

[73] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[74] M. Pawelczyk, S. Neel, and H. Lakkaraju, "In-context unlearning: Language models as few-shot unlearners," in *Forty-first International Conference on Machine Learning*.

[75] Y. Li, C.-E. Sun, and T.-W. Weng, "Effective skill unlearning through intervention and abstention," *arXiv preprint arXiv:2503.21730*, 2025.

[76] H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey, "Sparse autoencoders find highly interpretable features in language models," *arXiv preprint arXiv:2309.08600*, 2023.

[77] D. Shu, X. Wu, H. Zhao, D. Rai, Z. Yao, N. Liu, and M. Du, "A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models," *arXiv preprint arXiv:2503.05613*, 2025.

[78] A. Ng *et al.*, "Sparse autoencoder."

[79] P. Guo, A. Syed, A. Sheshadri, A. Ewart, and G. K. Dziugaite, "Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization," *arXiv preprint arXiv:2410.12949*, 2024.

[80] N. Yang, M. Kim, S. Yoon, J. Shin, and K. Jung, "Faithun: Toward faithful forgetting in language models by investigating the interconnectedness of knowledge," *arXiv preprint arXiv:2502.19207*, 2025.

[81] H. Lee, U. Hwang, H. Lim, and T. Kim, "Does localization inform unlearning? a rigorous examination of local parameter attribution for knowledge unlearning in language models," *arXiv preprint arXiv:2505.16252*, 2025.

[82] P. Thaker, S. Hu, N. Kale, Y. Maurya, Z. S. Wu, and V. Smith, "Position: Llm unlearning benchmarks are weak measures of progress," *arXiv preprint arXiv:2410.02879*, 2024.

[83] A. Lynch, P. Guo, A. Ewart, S. Casper, and D. Hadfield-Menell, "Eight methods to evaluate robust unlearning in llms," *arXiv preprint arXiv:2402.16835*, 2024.

[84] A. Schwarzschild, Z. Feng, P. Maini, Z. Lipton, and J. Z. Kolter, "Rethinking llm memorization through the lens of adversarial compression," *Advances in Neural Information Processing Systems*, vol. 37, pp. 56 244–56 267, 2024.

[85] H. Yuan, Z. Jin, P. Cao, Y. Chen, K. Liu, and J. Zhao, "Towards robust knowledge unlearning: An adversarial framework for assessing and improving unlearning robustness in large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 24, 2025, pp. 25 769–25 777.

[86] B. T. T. To and T. Le, "Harry potter is still here! probing knowledge leakage in targeted unlearned large language models via automated adversarial prompting," *arXiv preprint arXiv:2505.17160*, 2025.

[87] S. Hu, Y. Fu, S. Wu, and V. Smith, "Jogging the memory of unlearned models through targeted relearning attacks," in *ICML 2024 Workshop on Foundation Models in the Wild*, 2024.

[88] A. Deeb and F. Roger, "Do unlearning methods remove information from language model weights?" *arXiv preprint arXiv:2410.08827*, 2024.

[89] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *arXiv preprint arXiv:2307.15043*, 2023.

[90] Y. Hong, Y. Zou, L. Hu, Z. Zeng, D. Wang, and H. Yang, "Dissecting fine-tuning unlearning in large language models," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 3933–3941.

[91] L. Schwinn, D. Dobre, S. Xhonneux, G. Gidel, and S. Günnemann, "Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space," *Advances in Neural Information Processing Systems*, vol. 37, pp. 9086–9116, 2024.

[92] J. Du, Z. Wang, J. Zhang, X. Pang, J. Hu, and K. Ren, "Textual unlearning gives a false sense of unlearning," *arXiv preprint arXiv:2406.13348*, 2024.

[93] A. Zou, L. Phan, J. Wang, D. Duenas, M. Lin, M. Andriushchenko, J. Z. Kolter, M. Fredrikson, and D. Hendrycks, "Improving alignment and robustness with circuit breakers," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[94] H. Chen, S. Szyller, W. Xu, and N. Himayat, "Soft token attacks cannot reliably audit unlearning in large language models," *arXiv preprint arXiv:2502.15836*, 2025.

[95] Z. Zhang, J. Yang, P. Ke, S. Cui, C. Zheng, H. Wang, and M. Huang, "Safe unlearning: A surprisingly effective and generalizable solution to defend against jailbreak attacks," *arXiv preprint arXiv:2407.02855*, vol. 1, no. 2, p. 3, 2024.

[96] V. Patil, E. Stengel-Eskin, and M. Bansal, "Upcore: Utility-preserving coreset selection for balanced unlearning," *arXiv preprint arXiv:2502.15082*, 2025.

[97] W. Jeung, S. Yoon, H. Hong, S. Kim, S. Han, Y. Yu, and A. No, "Dusk: Do not unlearn shared knowledge," *arXiv preprint arXiv:2505.15209*, 2025.

[98] H. Chang and H. Lee, "Which retain set matters for llm unlearning? a case study on entity unlearning," *arXiv preprint arXiv:2502.11441*, 2025.

[99] J. Ren, Z. Dai, X. Tang, Y. Xing, S. Zeng, H. Liu, J. Zeng, Q. Peng, S. Varshney, S. Wang, Q. He, C. C. Aggarwal, and H. Liu, "Keeping an eye on llm unlearning: The hidden risk and remedy," 2025. [Online]. Available: https://arxiv.org/abs/2506.00359

[100] X. Yuan, T. Pang, C. Du, K. Chen, W. Zhang, and M. Lin, "A closer look at machine unlearning for large language models," *arXiv preprint arXiv:2410.08109*, 2024.

[101] Z. Jin, P. Cao, C. Wang, Z. He, H. Yuan, J. Li, Y. Chen, K. Liu, and J. Zhao, "Rwku: Benchmarking real-world knowledge unlearning for large language models," *arXiv preprint arXiv:2406.10890*, 2024.

[102] A. Ramakrishna, Y. Wan, X. Jin, K.-W. Chang, Z. Bu, B. Vinzamuri, V. Cevher, M. Hong, and R. Gupta, "Lume: Llm unlearning with multitask evaluations," *arXiv preprint arXiv:2502.15097*, 2025.

[103] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE, 2018, pp. 268–282.

[104] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson *et al.*, "Extracting training data from large language models," in *30th USENIX security symposium (USENIX Security 21)*, 2021, pp. 2633–2650.

[105] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer, "Detecting pretraining data from large language models," *arXiv preprint arXiv:2310.16789*, 2023.

[106] X. Lu, X. Niu, G. K. R. Lau, B. T. C. Nhung, R. H. L. Sim, F. Wen, C.-S. Foo, S.-K. Ng, and B. K. H. Low, "Waterdrum: Watermarking for data-centric unlearning metric," *arXiv preprint arXiv:2505.05064*, 2025.

[107] Q. Wang, J. P. Zhou, Z. Zhou, S. Shin, B. Han, and K. Q. Weinberger, "Rethinking llm unlearning objectives: A gradient perspective and go beyond," *arXiv preprint arXiv:2502.19301*, 2025.

[108] L. Ma, S. Yang, Z. Wang, Y. Wang, L. Wang, T. Wei, and K. Zhang, "Losing is for cherishing: Data valuation based on machine unlearning and shapley value," *arXiv preprint arXiv:2505.16147*, 2025.

[109] L. S. Shapley *et al.*, "A value for n-person games," 1953.

[110] X. Xu, X. Yue, Y. Liu, Q. Ye, H. Hu, and M. Du, "Unlearning isn't deletion: Investigating reversibility of machine unlearning in llms," *arXiv preprint arXiv:2505.16831*, 2025.

[111] H. Wang, Y. Jing, H. Sun, Y. Wang, J. Wang, J. Liao, and D. Tao, "Erasing without remembering: Implicit knowledge forgetting in large language models," *arXiv preprint arXiv:2502.19982*, 2025.

[112] S. Yoon, W. Jeung, and A. No, "R-tofu: Unlearning in large reasoning models," *arXiv preprint arXiv:2505.15214*, 2025.

[113] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.03300*, 2020.

[114] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 595–46 623, 2023.

[115] Y. Scholten, S. Günnemann, and L. Schwinn, "A probabilistic perspective on unlearning and alignment for large language models," *arXiv preprint arXiv:2410.03523*, 2024.

[116] A. Krishnan, S. Reddy, and M. Mosbach, "Not all data are unlearned equally," *arXiv preprint arXiv:2504.05058*, 2025.

[117] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang, "An empirical study of catastrophic forgetting in large language models during continual fine-tuning," *arXiv preprint arXiv:2308.08747*, 2023.

[118] S. Pal, C. Wang, J. Diffenderfer, B. Kailkhura, and S. Liu, "Llm unlearning reveals a stronger-than-expected coreset effect in current benchmarks," *arXiv preprint arXiv:2504.10185*, 2025.

[119] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[120] B. Wei, K. Huang, Y. Huang, T. Xie, X. Qi, M. Xia, P. Mittal, M. Wang, and P. Henderson, "Assessing the brittleness of safety alignment via pruning and low-rank modifications," *arXiv preprint arXiv:2402.05162*, 2024.

[121] B. Y. Lin, A. Ravichander, X. Lu, N. Dziri, M. Sclar, K. Chandu, C. Bhagavatula, and Y. Choi, "The unlocking spell on base llms: Rethinking alignment via in-context learning," *arXiv preprint arXiv:2312.01552*, 2023.

[122] Z. Zhang, F. Wang, X. Li, Z. Wu, X. Tang, H. Liu, Q. He, W. Yin, and S. Wang, "Catastrophic failure of llm unlearning via quantization," *arXiv preprint arXiv:2410.16454*, 2024.