Review

Machine Unlearning in Large Language Models: A Survey of Challenges and Methods

Xiaming Tu , Tianqing Zhu , Zhenni Liu , Ping Xiong * , Wanlei Zhou

Posted Date: 3 March 2026

doi: 10.20944/preprints202603.0114.v1

Keywords: machine unlearning; large language models; data privacy; LLM unlearning; knowledge erasure

 doi:10.20944/preprints202603.0114.v1

Review

Machine Unlearning in Large Language Models: A Survey of Challenges and Methods

Xiaming Tu [1], Tianqing Zhu [2], Zhenni Liu [1], Ping Xiong [1,*] and Wanlei Zhou [2]

[1] School of Information Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China
[2] Faculty of Data Science, City University of Macau, Macau, China
* Correspondence: pingxiong@zuel.edu.cn

Abstract

The rapid development of Large Language Models (LLMs) has made machine unlearning essential for privacy and compliance. This technology erases specific information without retraining the whole model. However, the inherent complexity of LLMs leads to fundamental differences in machine unlearning compared to traditional models. To analyze these distinctions, this survey conducts a detailed comparison of machine unlearning in traditional models and LLMs. This comparison reveals four major challenges: performance degradation, unlearning completeness, efficiency and cost, and black-box constraints. Instead of broadly categorizing algorithms, we structure our taxonomy around these core challenges, systematically evaluating how existing methodologies mitigate these specific risks, and finally discuss promising directions for future research.

Keywords: machine unlearning; large language models; data privacy; LLM unlearning; knowledge erasure

1. Introduction

Large Language Models (LLMs) possess an incredible ability to generate text that closely resembles human-authored content, with prominent examples including the GPT series [1], BERT [2], PaLM [3], and Deepseek [4]. Their capabilities in text generation, summarization, and reasoning have led to widespread application across various areas such as education, law, and communications. However, the reliance of LLMs on vast and varied datasets increases privacy and ethical concerns in machine learning [5,6]. To address these growing concerns, new legislation has emerged to protect individual privacy. For instance, regulations such as the European Union's General Data Protection Regulation (GDPR) [7], the California Consumer Privacy Act (CCPA) [8], Japan's Act on the Protection of Personal Information (APPI) [9], and Canada's Consumer Privacy Protection Act (CPPA) [10] now legally guarantee the "right to be forgotten". To comply with these regulations, a new technology called machine unlearning has been developed to address the issue [11]. Machine unlearning removes data from a model's training set and their residual effects on its internal parameters.

Machine unlearning is a technology designed to eliminate specific data samples and their influence from a trained model. To achieve this, algorithms apply various mechanisms to remove the effects of targeted data. The primary objective is to create an updated model that functions as if the unlearned data were never included in the original training set. However, applying machine unlearning to LLMs faces significant challenges. As shown in Figure 1, the LLMs workflow is more complex. Unlike traditional models trained on task-specific data, LLMs begin with pre-training on a massive corpus, followed by complex fine-tuning to align behavior. This extensive process embeds knowledge deep within the model's parameters, making specific information extremely difficult to isolate and erase.

 doi:10.20944/preprints202603.0114.v1

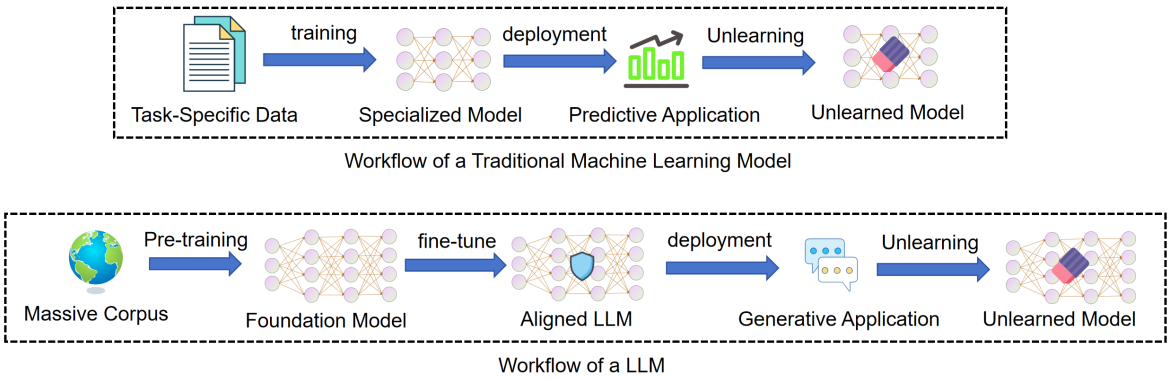

Figure 1. Comparison of Traditional Machine Learning and LLM Workflows

Despite these complexities, the need to align LLMs with privacy regulations and ethical standards makes machine unlearning a key research priority. This demand has led to a rapid increase in academic literature that introduces various unlearning algorithms and evaluation protocols. To organize this growing research, several survey papers categorize existing techniques and provide initial taxonomies for the field. However, current literature lacks a comprehensive review that specifically targets the unique technical challenges of deploying unlearning in LLMs. They often focus on broad security threats or methods rather than practical implementation issues. To address this gap, this survey systematically examines these unresolved issues. Table 1 details the distinctions between our work and existing surveys and highlights the necessity of an approach focused on these challenges.

Table 1. Comparison of Representative Surveys.

Reference	Year	Primary Taxonomy	Key Focus / Coverage
Liu et al. [12]	2025	Model and Input Optimization	**Focus:** Examines optimization algorithms and in-context learning techniques for LLMs. **Limitation:** Categorizes methods by algorithmic steps rather than the practical challenges of deployment such as cost or black-box access.
Xu et al. [13]	2024	Data Reorganization and Model Manipulation	**Focus:** Emphasizes verification mechanisms and data handling strategies in general machine learning. **Limitation:** Targets traditional models and lacks detail on LLM issues such as catastrophic forgetting in generative tasks.
Liu et al. [14]	2024	Data and Model Modification Techniques	**Focus:** Analyzes unlearning in the context of privacy risks such as inference attacks and defense mechanisms. **Limitation:** Views unlearning mainly as a security defense with less discussion on utility preservation or computational efficiency.
Wang et al. [15]	2024	LLM Lifecycle Stages	**Focus:** Reviews security threats across pre-training, fine-tuning, and deployment. **Limitation:** Discusses unlearning briefly as a remediation strategy and lacks detailed algorithmic comparison.
Ours	**2026**	**Practical Deployment Challenges**	**Focus:** Systematically organizes methods by four core *challenges*: performance degradation, completeness, efficiency, and black-box constraints. **Coverage:** Bridges the gap between theoretical algorithms and practical application requirements.

 doi:10.20944/preprints202603.0114.v1

To bridge these gaps, this survey organizes the field of machine unlearning for LLMs from a novel perspective, centering on the practical implementation hurdles rather than abstract taxonomies. Our core contributions are summarized as follows:

- We conduct a comprehensive comparison between machine unlearning in traditional models and LLMs across **data**, **model**, and **output** dimensions to highlight the unique challenges specific to LLMs.
- Based on this comparative analysis, we propose a taxonomy organized around four core implementation challenges: **performance degradation**, **unlearning completeness**, **efficiency and cost**, and **black-box constraints**.
- We review existing methodologies using this taxonomy and evaluate how current techniques address these constraints.
- Finally, we use this analysis to identify critical research gaps within each challenge and outline a clear direction for future investigation.

2. Machine Unlearning in LLMs and Traditional Models

2.1. Foundations of Machine Unlearning

2.1.1. Machine Unlearning

Machine unlearning aims to precisely remove the influence of a specific subset of data from a trained model, without affecting the model's performance on the remaining data. To formalize this, we first define the full training dataset as $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where each sample consists of an input x_i from an instance space $\mathcal{Z}$ and a corresponding label y_i from a label space $\mathcal{Y}$. In LLM scenarios, x_i typically represents the input text, while y_i denotes the subsequent token the model learns to predict. The full dataset splits into two disjoint subsets: the forgetting set ($D_f \subset D$), which contains the data points to be erased, and the retention set ($D_r = D \setminus D_f$), which contains the data that the model should keep remembering. The original model, denoted as M_0, is trained on the full dataset D, resulting in a set of parameters $\boldsymbol{\theta}$. The objective of the unlearning process is to produce an unlearned model M_u without retraining. We define the unlearning algorithm as U, which generates a new set of parameters $\boldsymbol{\theta}_u = U(D, D_f, \boldsymbol{\theta})$. The theoretical "gold standard" is the retrained model, M_r, which is trained from scratch using only the retention set D_r to derive parameters $\boldsymbol{\theta}_r$. Table 2 summarizes the symbols and notations.

Table 2. Symbols and Notations in Machine Unlearning

Notation	Description	Notation	Description
D	Full dataset	$\boldsymbol{\theta}$	Parameters of original model
D_f	Forgotten set	$\boldsymbol{\theta}_u$	Parameters of unlearned model
D_r	Retained set	$\boldsymbol{\theta}_r$	Parameters of retrained model
(x_i, y_i)	Sample pair	$F(\cdot;\boldsymbol{\theta})$	Output distribution of original model
$\mathcal{Z}$	Input space	$F(\cdot;\boldsymbol{\theta}_u)$	Output distribution of unlearned model
$\mathcal{Y}$	Label space	$F(\cdot;\boldsymbol{\theta}_r)$	Output distribution of retrained model
M_o	Original model	$A(\cdot)$	Learning process
M_u	Unlearned model	$U(\cdot)$	Unlearning process
M_r	Retrained model	$\mathrm{Dist}(\cdot,\cdot)$	Behavioral divergence measure

A robust machine unlearning process must balance two complementary objectives: effectiveness, which modifies behavior on the forgetting set, and locality, which safeguards performance on the retention set. We can express these dual goals jointly as:

$$\begin{cases} \max_{\boldsymbol{\theta}_u} \ \mathrm{Dist}\big(F(D_f;\boldsymbol{\theta}),\ F(D_f;\boldsymbol{\theta}_u)\big) & \text{(Effectiveness)} \\ \min_{\boldsymbol{\theta}_u} \ \mathrm{Dist}\big(F(D_r;\boldsymbol{\theta}),\ F(D_r;\boldsymbol{\theta}_u)\big) & \text{(Locality)} \end{cases}, \tag{1}$$

 doi:10.20944/preprints202603.0114.v1

where $F(\cdot;\boldsymbol{\theta}_u)$ and $F(\cdot;\boldsymbol{\theta})$ respectively denote the output distributions of unlearned model and original model. $\mathrm{Dist}(\cdot,\cdot)$ quantifies a divergence between these distributions, such as KL divergence.

2.1.2. Exact and Approximate Unlearning

While effectiveness and locality define the optimization goals in equation (1), we need formal guarantees to certify that data is truly gone. Machine unlearning is categorized into exact and approximate forms by measuring the statistical divergence between the unlearned and retrained models. Figure 2 illustrates this verification framework.

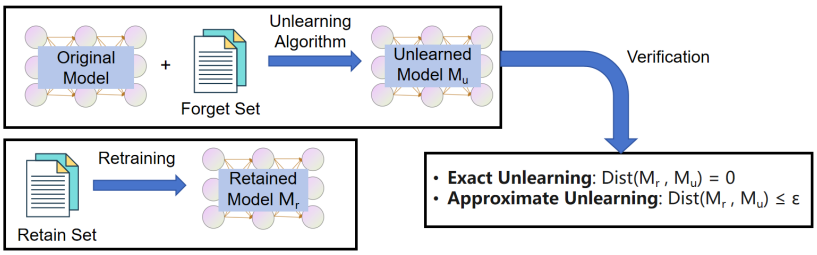

Figure 2. Exact and Approximate Unlearning

Exact Unlearning requires the unlearned model to be statistically indistinguishable from the retrained model. This is achieved when the divergence between their output distributions on the forgetting set is zero:

$$\mathrm{Dist}\big(F(D_f;\boldsymbol{\theta}_u), F(D_f;\boldsymbol{\theta}_r)\big) = 0. \tag{2}$$

Approximate Unlearning relaxes this strict constraint. Recognizing that perfect exact unlearning is often computationally intractable, it allows for a bounded discrepancy within a tolerance $\epsilon > 0$:

$$\mathrm{Dist}\big(F(D_f;\boldsymbol{\theta}_u), F(D_f;\boldsymbol{\theta}_r)\big) \leq \epsilon. \tag{3}$$

2.1.3. Machine Unlearning Methods for Traditional Models

In traditional machine learning, unlearning operates as a functional update mechanism. The process transforms the existing model parameters trained on the full dataset into a new state. The objective is to effectively remove the influence of specific samples so that the parameters function as if the forgotten data had never existed in the training set. We formalize this process as finding an unlearning function U that satisfies the following equivalence:

$$\boldsymbol{\theta}_u = U(A(D), D, D_f) \quad \text{s.t.} \quad \boldsymbol{\theta}_u \sim A(D_r), \tag{4}$$

where $A(D)$ denotes the initial model parameters generated by the learning algorithm A on the full dataset D. The unlearning algorithm U uses these pre-trained parameters and the relevant data subsets to produce the unlearned parameters $\boldsymbol{\theta}_u$. The constraint $\boldsymbol{\theta}_u \sim A(D_r)$ sets a strict distributional requirement where the unlearned model must be statistically indistinguishable from the parameters $A(D_r)$ generated by training exclusively on the retained set D_r.

2.1.4. Machine Unlearning Methods for LLMs

Due to the complexity and large parameter space of LLMs, unlearning methods range from direct parameter updates to inference-time interventions. Unlike traditional methods that enforce a hard distributional constraint (Eq. 4), LLM unlearning is formulated as a relaxed optimization problem. It includes adjusting activation or logit offsets and using few-shot in-context examples without modifying parameters. The process determines the optimal configuration $(\boldsymbol{\theta}_u, \delta_u)$ that minimizes a bi-objective loss:

 doi:10.20944/preprints202603.0114.v1

$$(\boldsymbol{\theta}_u, \delta_u) = \arg\min_{\boldsymbol{\theta},\delta} \left(\mathbb{E}_{x \in D_\mathrm{f}} [L_\mathrm{f}(f(x; \boldsymbol{\theta}, \delta), y_\mathrm{f})] + \lambda \mathbb{E}_{x \in D_\mathrm{r}} [L_\mathrm{r}(f(x; \boldsymbol{\theta}, \delta), y_\mathrm{r})] \right), \tag{5}$$

where L_f and L_r denote the task-specific loss functions for forgetting and retaining, respectively. These objectives are often asymmetric: L_r typically ensures the preservation of linguistic fluency and general knowledge, while L_f targets the suppression of specific information. The function $f(x; \boldsymbol{\theta}, \delta)$ generates the model output for input x, parameterized by weights $\boldsymbol{\theta}$ and modified by an auxiliary intervention variable δ. $(\boldsymbol{\theta}_u, \delta_u)$ defines the unlearned state, where $\boldsymbol{\theta}$ denotes internal model parameters and δ includes external or inference-time control components, such as activation or logit offsets and few-shot in-context examples.

2.2. Key Differences Between LLM Unlearning and Traditional Machine Unlearning

As summarized in Figure 3, we compare the unlearning in traditional models and LLMs across three dimensions: **data**, **model**, and **output**. These distinctions clarify why machine unlearning strategies designed for traditional models may not work for LLMs.

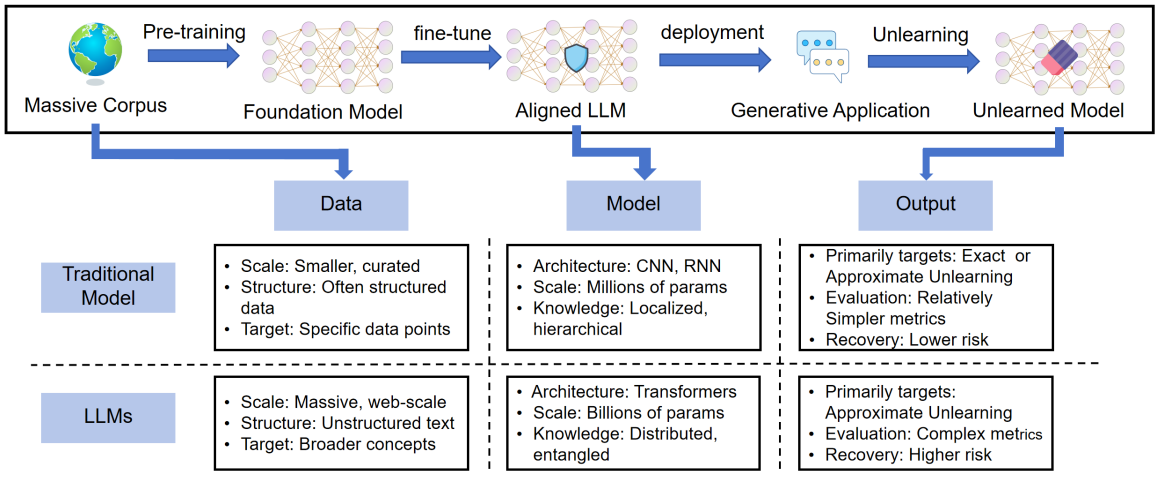

Figure 3. Comparing Machine Unlearning: Traditional Models vs. LLMs

2.2.1. Data

Pre-training Data Scale.

LLMs are pretrained on vast and diverse text corpora, which often contain billions to trillions of tokens. These datasets aggregate information from a variety of sources, including books, journals, and the Internet. Conversely, traditional models generally rely on much smaller, bounded datasets tailored to specific domains.

Data Dependencies.

LLMs run on unstructured text, which is inherently varied and fluid. To process such text effectively, the model must understand how words connect across long distances and how their meanings change depending on the context. This differs from traditional models, which typically rely on structured data with clearer, more direct relationships.

Unlearning Targets.

Machine unlearning in traditional models focuses on erasing certain data points or subsets to eliminate the impact of specific training data. In contrast, the unlearning targets in LLMs often cover concepts, subjects, or knowledge sets [16]. This difference arises because LLM unlearning addresses broad categories like copyrighted material, specific capabilities, or harmful content. These targets spread throughout the data rather than existing as isolated samples [17].

 doi:10.20944/preprints202603.0114.v1

2.2.2. Model

Model Architecture.

Traditional models, such as CNNs and RNNs, rely on architectures designed for local or sequential processing. In contrast, LLMs rely on the Transformer architecture [18,19]. This architecture leverages multi-head self-attention to establish dense, global connections between every token.

Model Scale.

LLMs operate on a massive scale of parameters, such as GPT-3 with its 175 billion parameters [20, 21]. In contrast, traditional models are much smaller, typically ranging from thousands to millions. For instance, the ResNet-50 only has about 25 million parameters [22].

Knowledge Distribution.

Traditional models like CNNs generally organize knowledge in a hierarchy, with each layer handling a specific task [23,24]. In contrast, LLMs integrate knowledge throughout the entire system. Instead of locking information into isolated components, the self-attention mechanism scatters it across all layers and heads [2,25].

2.2.3. Output

Unlearning Precision

For traditional models, achieving exact unlearning is feasible under certain conditions. Frameworks like SISA (Sharded, Isolated, Sliced, and Aggregated) enable this via partitioned retraining [26]. However, the sheer scale and complexity of LLMs render exact unlearning impractical. As a result, strategies for LLMs adopt approximate unlearning, focusing on removing data influence effectively without significantly affecting the model's general capabilities.

Evaluation Metrics

Evaluating traditional models is straightforward, leaning on standard metrics like accuracy and membership inference attacks [26–28]. In contrast, LLMs face a double challenge. Evaluating linguistic fluency is far more difficult than measuring raw accuracy. Moreover, verifying that abstract concepts have been truly erased is much harder than checking for isolated data points [29–31].

Knowledge Recovery

LLMs are more susceptible to knowledge recovery than traditional models [32,33]. Traditional models tend to localize knowledge in specific areas, making it easier to erase completely and harder to recover later. Conversely, the complex Transformer architecture in LLMs distributes knowledge across the network. This structural difference means traces of information often persist and makes it easier to recover forgotten knowledge [34,35].

2.3. Core Challenges in LLM Unlearning

Based on the differences discussed earlier, we outline four key challenges that LLM models face compared to traditional machine unlearning.

2.3.1. Performance Degradation

The most immediate challenge in LLM unlearning is preventing performance degradation. This refers to the unintended loss of the model's general capabilities the unlearning process. For instance, a model might lose linguistic fluency, leading to redundant or incoherent text generation. Performance degradation may also reduce the reasoning ability on tasks unrelated to the unlearning target. Any significant drop in quality weakens the model's value and lowers user confidence.

This challenge first arises primarily from the complex **data** dependencies inherent in unstructured text. LLMs process language by modeling word relationships across large contexts. Unlearning targets often span across these linguistic structures rather than existing as discrete units. Consequently,

 doi:10.20944/preprints202603.0114.v1

removing a specific concept creates gaps in the syntactic and semantic associations required for text generation. This disruption directly reduces the model's linguistic fluency.

The underlying **model** architecture further amplifies this challenge. Based on the Transformer framework, LLMs distribute knowledge across billions of parameters via multi-head self-attention [18, 19]. Consequently, a single concept is distributed throughout the network rather than localized to a specific region. Modifying parameters to excise a target inevitably alters weights responsible for broader functions. These unintended adjustments degrade general capabilities and weaken reasoning abilities on tasks unrelated to the unlearning target [36,37].

Finally, the specific constraints regarding model **output** make preserving utility more difficult. Unlike traditional approaches that often achieve exact unlearning, LLMs use approximate strategies to keep computational feasibility. This lack of precision requires a balance between effectively removing the target and maintaining overall performance. Filtering specific information without precise boundaries distorts the model's probability distribution. This distortion results in the hallucinations and reduced coherence often observed after the unlearning process.

2.3.2. Unlearning Completeness

In addition to maintaining performance, machine unlearning requires legal and ethical compliance, which presents a significant challenge: guaranteeing completeness. Completeness refers to fully erasing the target data's influence on the model. This objective extends beyond merely preventing the model from reproducing text. The process requires removing latent associations, including learned facts, biases, and writing styles. Failing to achieve this may trigger serious legal liabilities.

Defining precise unlearning targets within the training **data** is difficult because concepts are semantically entangled. Unlike traditional scenarios where targets are discrete samples, LLM unlearning addresses abstract concepts or broad knowledge sets [16]. These targets are not isolated units. Instead, they connect to other information throughout the dataset. Consequently, algorithms often struggle to separate the specific target concept from the general knowledge that must be retained.

The complex knowledge distribution within the **model** disperses specific information across the entire parameter space. Instead of being localized to a single location, a concept is distributed across the network through dense connections. Identifying and modifying every parameter related to a specific target is intractable. As a result, unlearning methods often fail to remove the information completely. Traces of the targeted data frequently persist within the parameters even if the explicit generation stops [36].

Relying solely on model **output** does not distinguish between erased information and suppressed generation. Current evaluation metrics cannot verify whether information is permanently eliminated or merely inaccessible during standard inference. This uncertainty indicates that information remains latent within the parameters. Adversaries can use techniques like fine-tuning to reactivate these hidden traces [32,38]. This potential for knowledge recovery makes it difficult to certify that data is irreversibly removed [39,40].

2.3.3. Efficiency and Cost

Even if a method ensures performance and completeness, it remains impractical if it is too slow or costly. This highlights the challenge of efficiency and cost. Although machine unlearning is far more efficient than retraining, it still requires significant computational resources, including processing power, time, and energy. If the process is too slow or expensive to handle unlearning requests on time, it can lead to serious consequences including legal risks.

The extensive volume of pre-training **data** imposes substantial computational demands. Processing trillions of tokens to locate specific concepts consumes significant time and energy. This scale means that a single concept is often distributed across countless examples. Consequently, algorithms must scan or index vast amounts of information to identify the relevant subsets for removal. This requirement significantly slows down the unlearning procedure prior to model updates.

 doi:10.20944/preprints202603.0114.v1

Updating the parameters of a large-scale **model** multiplies this cost by requiring substantial processing power. Since these systems contain billions of parameters, even a minimal update involves calculating gradients for huge matrices. This process demands high-performance GPUs and extensive memory. As model scale increases, the computational cost of modifying these weights increases rapidly. This makes frequent unlearning updates expensive for many organizations. The **output** verification process also introduces latency. Methods typically generate text sequences to validate that knowledge is successfully removed. While this step requires inference resources, the additional cost is generally less significant than the heavy demands of data processing and parameter updates.

2.3.4. Black-Box Constraints

The challenge of black-box constraints arises from commercial deployment restrictions. In most real-world scenarios, users only interact with a model through an interface that sends inputs and receives outputs. They have no direct access to the internal LLM components, such as parameters, gradients, and activation states. Consequently, this restriction presents a challenge for unlearning strategies.

Restricted access to internal **model** parameters prevents direct modification of neural weights. Since developers cannot calculate gradients or adjust specific layers, they are forced to use indirect methods. This restriction makes standard unlearning methods that rely on weight updates unavailable. Consequently, practitioners must interact with the system as a black box with inaccessible internal mechanisms.

Without access to the original training **data**, verifying which specific examples influenced the model becomes impossible. In black-box scenarios, practitioners often lack the dataset used for pre-training. This constraint forces reliance on limited user-provided examples or synthetic queries. Therefore, algorithms cannot accurately estimate the influence of the forget set without the full context of the original distribution.

Relying solely on external **output** signals constrains the verification of unlearning. Restricted access to internal patterns makes it harder to distinguish whether information is erased or merely suppressed. Adversaries may exploit this latent knowledge to recover information using specific prompts. As a result, black-box unlearning presents a higher risk of privacy leakage compared to white-box settings [17,37].

2.4. Summary

In this section, we outlined the basics of machine unlearning and explained why LLMs differ fundamentally from traditional models. These unique characteristics create four major challenges: performance degradation, unlearning completeness, efficiency and cost, and black-box constraints. Table 3 summarizes how these distinctions drive each challenge. We use a star (*) rating to show their impact: five stars indicate a primary cause, while fewer stars indicate a smaller influence. Figure 4 outlines the organizational structure of this survey. It correlates fundamental LLM characteristics (Key Distinctions) with specific associated difficulties (Challenges) and connects them to solution strategies discussed subsequently. In Sections 3–6, we explore each of these four challenges in detail. We review and categorize existing works based on how they address these fundamental problems.

 doi:10.20944/preprints202603.0114.v1

Table 3. Unlearning Challenges Driven by Key Model Distinctions.

Category	Distinctions	Performance	Completeness	Efficiency	Black-Box
Data	Pre-training Data Scale	****	****	*****	****
Data	Data Dependencies	****	****	**	***
Data	Unlearning Targets	****	*****	***	****
Model	Model Architecture	****	****	***	****
Model	Model Scale	****	*****	*****	**
Model	Knowledge Distribution	*****	*****	**	*****
Output	Unlearning Precision	***	***	**	*****
Output	Evaluation Metrics	***	*****	**	*****
Output	Knowledge Recovery	**	****	*	*****

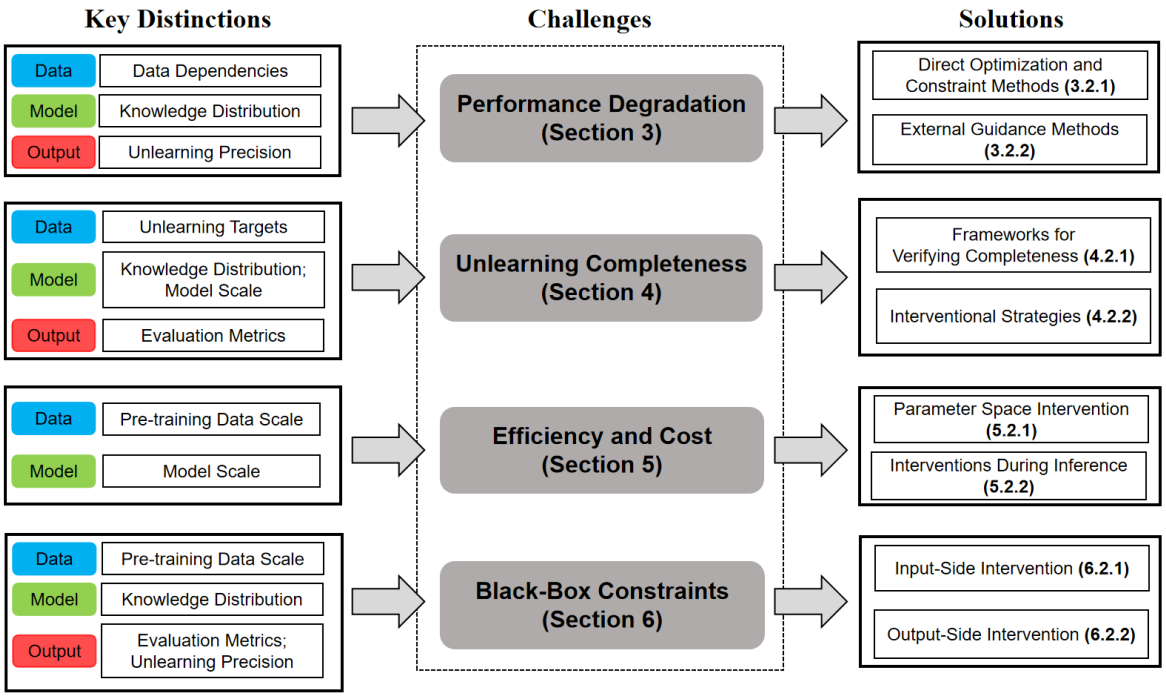

Figure 4. Overview of Challenges and Solutions in LLM Unlearning.

3. Challenge of Performance Degradation

As highlighted in Section 2.3.1, performance degradation is the most direct and serious side effect when removing data from LLMs. It impairs linguistic fluency and limits the ability of the model to handle ordinary tasks. This decline in quality makes the model less useful and reliable for most applications. Researchers have developed various methods to tackle this problem. In this section, we group them based on their working mechanisms. **Direct Optimization and Constraint Methods** update the model directly via specific datasets or parameter constraints. **External Guidance Methods** rely on auxiliary external modules to guide the unlearning process such as teacher models or reward models. The remainder of this section provides a detailed review of these strategies.

3.1. Methodologies for Mitigating Performance Degradation

3.1.1. Direct Optimization and Constraint Methods

This category focuses on strategies that directly modify the model's internal weights to achieve unlearning. These methods leverage specific constraints and training objectives to surgically remove unwanted data without compromising the model's broader knowledge. **Replacement Fine-tuning**

 doi:10.20944/preprints202603.0114.v1

We begin with the most straightforward method: Replacement Fine-tuning. This method replaces the sensitive or wrong information with secure alternatives. The model is then briefly fine-tuned on this modified data. Since the replacement words fit naturally into the original context, the modification preserves the model's linguistic fluency. Eldan and Russinovich [16] applied this concept to erase knowledge of the Harry Potter series. Their method identifies tokens that are strongly associated with the story and replaces them with generic terms. Then the model is trained to predict these neutral words instead of the original details, as illustrated in the framework in Figure 5. When the model receives a prompt about where Harry Potter studies, the original model answers 'Hogwarts,' whereas the modified model generates a generic response like 'Mystic Academy'. Based on this idea, Gu et al. [41] generate "inverted facts," to effectively overwrite the memory. They use alternative answers that directly contradict the original information instead of just being neutral.

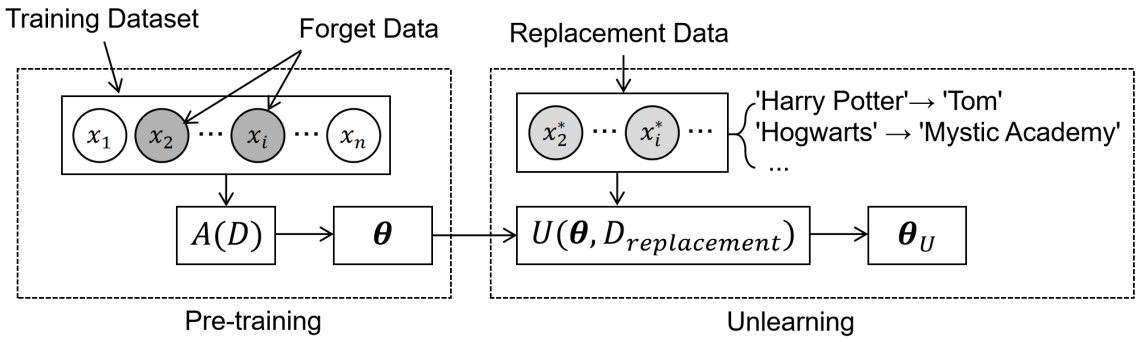

Figure 5. Workflow of Replacement Fine-tuning

Regularized Optimization

Regularized Optimization preserves model utility by deriving parameter importance from the retain set. Rather than treating all parameters equally, these strategies analyze the retain data to distinguish between critical and redundant weights. As illustrated in Figure 6, the process typically starts by computing a sensitivity matrix based on the retain set. This sensitivity map steers the unlearning update, strictly constraining changes to high-sensitivity parameters to protect general capabilities. This forces the necessary modifications into low-sensitivity regions.

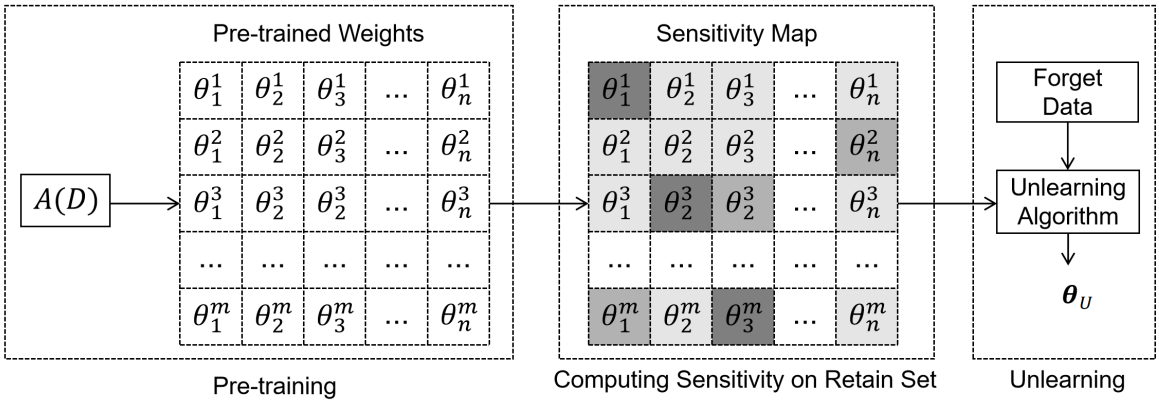

Figure 6. Schematic of Regularized Optimization

Cha et al. [42] introduced FILA (Fisher-weighted Initialization for LoRA Adapters). This strategy calculates an optimal initialization for adapter matrices to minimize the weighted difference from original weights. It uses the relative Fisher information to quantify the importance of each parameter. FILA restricts updates to less sensitive parameters and effectively preserves the model's general performance. Similarly, Ding et al. [43] propose LLMEraser, using influence functions to figure

 doi:10.20944/preprints202603.0114.v1

out precise parameter adjustments. The method structures the update as a quadratic optimization problem centered on the Hessian and the gradient. This optimization leverages model curvature to automatically curb modifications in highly curved directions, preserving general knowledge.

Representation Misalignment

Instead of constraining parameter updates directly, Representation Misalignment focuses on manipulating the model's internal activation space. RMU (Representation Misdirection for Unlearning) [44] imposes geometric constraints on the model's internal representations. It works by redirecting the internal representations of the model. To drive this process, the method minimizes a dual-objective loss function. The first part drives the dangerous activations to a random direction. The second part makes sure that the safe activations match the frozen reference model. To overcome convergence issues in deeper layers due to fixed scaling constants, Dang et al. [45] proposed Adaptive RMU, which replaces the static constant with a dynamic scaling factor derived from the representation norms of the frozen model.

Partial Parameter Modification

Partial Parameter Modification selectively updates the weights associated with the particular knowledge. As illustrated in Figure 7, the process first pinpoints the specific parameters responsible for the target knowledge. It then modifies only these selected parts while freezing the rest of the network to preserve the model's overall performance. We organize these methods by scale: ranging from single neurons to larger parameter blocks, and finally entire layers. Modifying only important weights keeps the network stable and protects general performance.

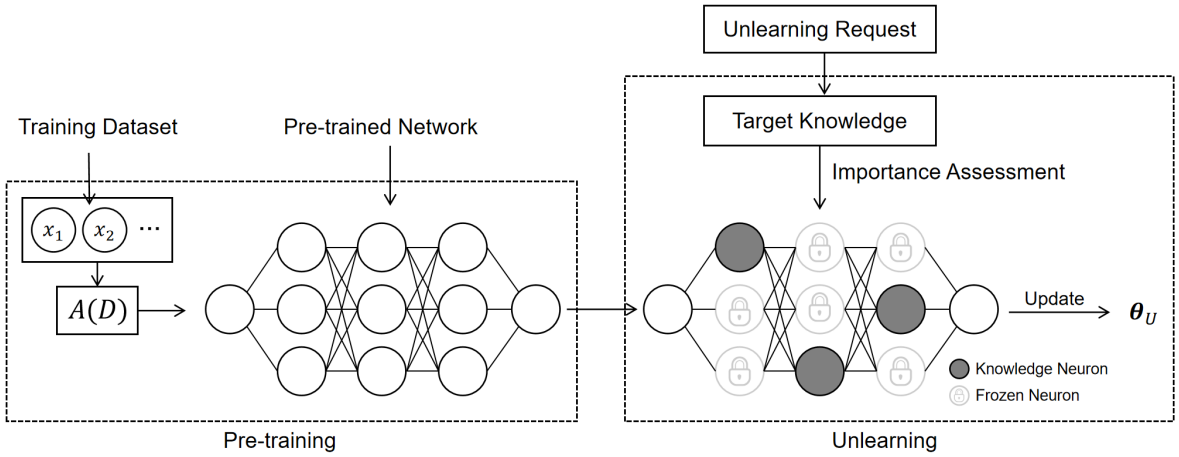

Figure 7. Schematic of Partial Parameter Modification.

At the neuron level, Dai et al. [46] introduced the concept of knowledge neurons in the feed-forward networks (FFNs) of Transformers that encode factual knowledge. They identify these neurons based on integrated gradients. The attribution score calculates each neuron's contribution to certain knowledge by measuring the integral of gradients with respect to the neuron's weight. This method pinpoints specific "knowledge neurons" by identifying neurons with high attribution scores, which can be updated with minimal impact on the remaining parameters. Similarly, KLUE (Knowledge-Localized UnlEarning) [47] uses attribution analysis with regularization to identify important parameters, ensuring exact updates without affecting the remainder of the model. WAGLE (Weight Attribution-Guided LLM Unlearning Framework) [48] calculates an importance score for each weight by solving a bi-level optimization problem. NeuMuter [49] uses a trainable mask to detect the neurons in feed-forward networks that memorize undesired data. The mask is applied to suppress the output of these identified neurons to achieve unlearning.

At the block level, PCGU (Partitioned Contrastive Gradient Unlearning) [50] targets parameter blocks to eliminate bias. The method first divides the model's parameters into blocks. The importance of each block is assessed using the cosine similarity between gradient directions. A lower cosine

 doi:10.20944/preprints202603.0114.v1

similarity indicates that the block is more relevant to the bias. Then the most important blocks are updated. PCGU freezes the rest of the model to achieve efficient machine unlearning while preserving overall language modeling abilities.

Finally, at the layer level, LUNAR (LLM Unlearning via Neural Activation Redirection) [51] identifies the optimal MLP layer that can trigger refusal responses effectively. FALCON (Fine-grained Activation Manipulation by Contrastive Orthogonal Unalignment) [52] utilizes Mutual Information (MI) to identify layers which have minimal entanglement between the forget sets and retain sets.

Preference Optimization

Preference Optimization enhances robustness through the refinement of Gradient Ascent (GA). As a basic machine unlearning strategy, GA updates the model to maximize the negative log-likelihood of the target sequences [53]. However, GA focuses on suppressing the target memory without using constraints to preserve the model's remaining knowledge, which may significantly reduce the model's overall utility [54,55]. On the TOFU benchmark, using GA to forget 10% of the data resulted in a severe loss of performance [29]. To address this issue, Rafailov et al. [56] propose Direct Preference Optimization (DPO), which reframes unlearning as a pairwise preference task. DPO trains the model to favor a benign reaction and suppress the target information. It also keeps the model anchored to a frozen reference policy to prevent it from drifting too far. Empirically, this approach prevents the severe performance collapse that often occurs with raw GA.

Based on this, Negative Preference Optimization (NPO) simplifies the approach [57]. Compared to DPO, it requires only examples of what should be forgotten. It penalizes a forget sample typically only when the current model gives it a higher score than the reference model does. This method includes an adaptive weight that scales the gradient. As the target is unlearned, this weight decreases to prevent unnecessary changes and preserve general model capabilities.

3.1.2. External Guidance Methods

These approaches rely on external systems to steer the unlearning process. They use auxiliary tools, such as teacher models or reward signals, to direct optimization. This helps the model remove specific information while preventing it from losing its original capabilities.

Reinforcement Learning.

Reinforcement Learning (RL) reframes unlearning as a dynamic process driven by rewards. It evaluates entire sequences and constrains the magnitude of model updates. This allows it to eliminate unwanted outputs while maintaining the overall utility of the model [58–60]. Lu et al. [61] introduce Quark (Quantized Reward Konditioning), an RL-based approach. The method employs a three-stage iterative process: exploration (evaluating model outputs), quantization (grouping samples by quality), and learning (fine-tuning). As illustrated in Figure 8, the training process involves updating the model using samples from each quantile conditioned on their specific reward tokens. To ensure stability, a KL-divergence penalty is used to keep the new model statistically near to the frozen original model. This penalty restricts the update, preventing the model from drifting too far from its original distribution. At inference time, by conditioning on the highest reward token, the model generates text that corresponds with high-reward behavior (e.g., reduced toxicity), while maintaining the fluency and diversity of the original pre-trained model.

 doi:10.20944/preprints202603.0114.v1

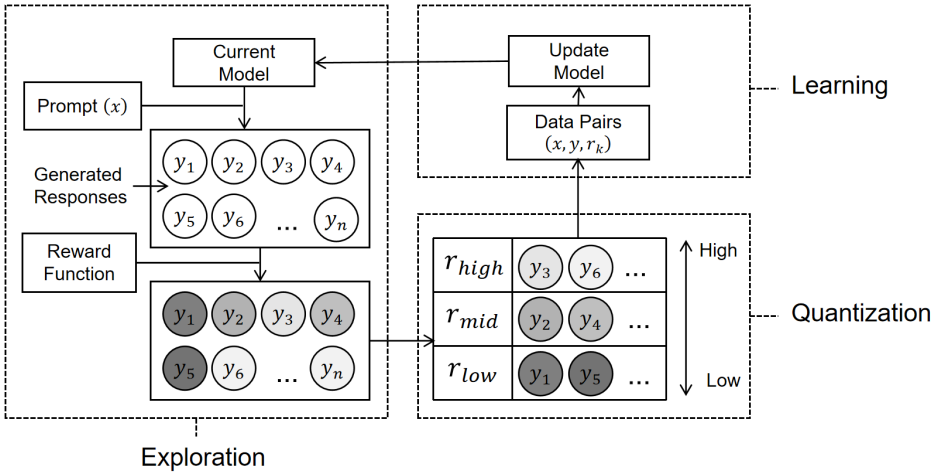

Figure 8. Schematic of the Quark framework.

Knowledge Distillation-Based Machine Unlearning

Knowledge distillation-based methods align the model's output distribution with a teacher model. By transferring the information contained in the teacher model, the student model can learn to remove specific information without affecting the broader knowledge structure [62]. Wang et al. [63] propose Knowledge Gap Alignment (KGA), designed for scalable machine unlearning in NLP. KGA compares the behavioral difference between model pairs trained on different datasets. As illustrated in Figure 9, the framework utilizes extra data to establish a "reference gap" representing the unseen state, and then minimizes the difference between this reference and the target gap on the forget data. This strategy compels the unlearned model A^* to process the forget set D_f as if it were never seen, while preserving its predictions on the remaining data D_r.

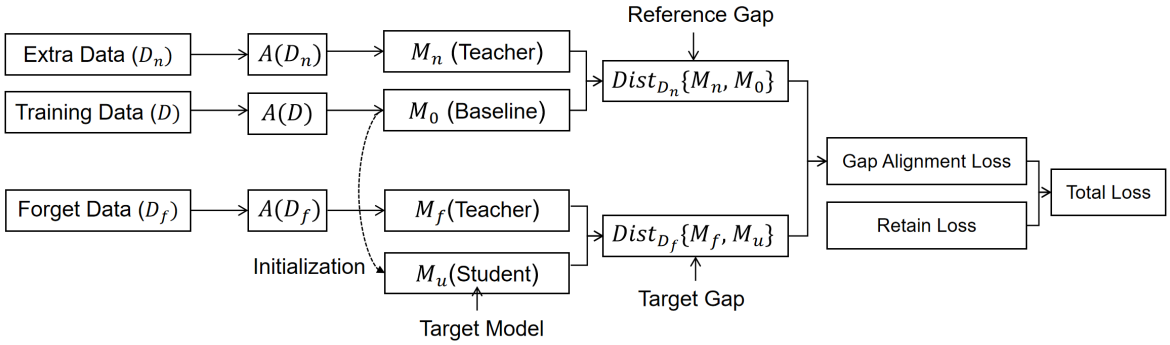

Figure 9. Schematic of the Knowledge Gap Alignment (KGA) framework.

3.2. Summary

We categorize strategies to preserve model utility generally into two primary categories: Direct Optimization and Constraint Methods and External Guidance Methods. These approaches have made significant progress in reducing performance loss. Many approaches manage to maintain fluency, coherence, and task accuracy after machine unlearning. Table 4 summarizes representative approaches discussed in this section. However, achieving perfect machine unlearning in LLMs without side effects remains highly challenging. Removing one specific piece of information usually accidentally affects related concepts even if we carefully select which parameters to change. Moreover, current evaluation metrics often lack the sensitivity to detect minor performance degradation.

 doi:10.20944/preprints202603.0114.v1

Table 4. Summary of Methodologies for Mitigating Performance Degradation.

Sub-category	Basic Ideas	Advantages	Limitations
Direct Optimization and Constraint Methods			
Replacement Fine-tuning [16,41]	Replaces sensitive information with neutral or inverted alternatives and briefly fine-tunes the model	Simple to implement; effectively maintains the linguistic fluency of the model	Difficult to fully erase deep knowledge
Regularized Optimization [42,43]	Identifies important parameters and restricts their updates during unlearning	Theoretically guaranteed to preserve general capabilities; precise control over weight changes	High computational overhead (e.g., calculating second-order derivatives)
Representation Misalignment [44,45]	Optimizes model parameters to steer the latent representations of sensitive inputs towards random directions	Effectively reduces token confidence for forgotten concepts; high robustness against adversarial jailbreak attacks	Effectiveness declines in deeper layers; sensitive to the settings of steering layers and coefficients.
Partial Parameter Modification [46,50,51]	Identifies and updates only the specific neurons or blocks responsible for target knowledge	Minimizes impact on the rest of the model; low risk of overall performance decline	Difficult to locate distributed knowledge; Risk of residual knowledge
Preference Optimization [56,57]	Uses preference pairs (benign vs. forget) to guide the model, often anchored to a reference model	Training is relatively stable; effectively prevents model collapse	Relies heavily on high-quality preference data; requires maintaining a reference model
External Guidance Methods			
Reinforcement Learning [61]	Uses dynamic reward signals to penalize unwanted outputs and guide the unlearning process	Effectively maintains linguistic fluency and diversity; provides precise control over generation	Complex pipeline due to iterative generation and scoring; higher computational cost than simple fine-tuning
KD-Based Unlearning [62,63]	Leverages teacher model outputs to constrain the unlearning process and preserve general capabilities	Can be applied to diverse NLP tasks; effectively maintains performance on retained data	Requires storage and maintenance of auxiliary models

4. Challenge of Unlearning Completeness

Completeness refers to the total removal of specific knowledge from the model's internal memory, rather than merely suppressing generation. If machine unlearning is incomplete, residual traces of the data remain in the model. This poses significant legal and security risks because attackers can find ways to recover the sensitive information. To ensure thorough unlearning, we divide our discussion into two parts. **Frameworks for Verifying Completeness** focus on evaluation tools necessary to verify that data has been effectively erased. **Interventional Strategies** discuss specific techniques designed to eliminate deep knowledge. The remainder of this section details these approaches.

 doi:10.20944/preprints202603.0114.v1

4.1. Methodologies for Verifying and Enhancing Completeness

4.1.1. Frameworks for Verifying the Completeness of Unlearning

Simply deleting data is not enough if we cannot prove it is gone. These frameworks act as strict inspectors, looking beyond simple surface-level answers to find any hidden traces of the original information. They ensure that the unlearning process meets high standards of safety and privacy.

Standard Benchmarks

Effective benchmarks must employ rigorous metrics to certify data removal, rather than merely checking if the model stops generating the verbatim text. Maini et al. [29] propose TOFU, a benchmark utilizing a dataset of fictitious authors to ensure the model has not seen them during training. It assesses unlearning completeness by testing whether the model can still tell the real answer apart from similar but wrong ones. If the model cannot distinguish the truth from fabricated options, the data is considered unlearned.

Unlike benchmarks that use synthetic data, Shi et al. [30] introduces MUSE (Machine Unlearning Six-Way Evaluation), a benchmark designed to evaluate machine unlearning algorithms in language models in six key aspects: no verbatim memorization, no knowledge memorization, no privacy leakage, utility preservation, scalability, and sustainability. For strictly hazardous domains, Li et al. [44] propose the Weapons of Mass Destruction Proxy (WMDP) benchmark, focusing on biosecurity, cybersecurity, and chemical security.

Advanced Probing and Deep Verification

Standard benchmarks primarily assess model outputs, but often fail to capture hidden traces of forgotten information. Consequently, researchers have developed advanced verification methodologies from diverse perspectives. Adversarial strategies are employed to induce the model to generate suppressed content. Chen et al. [64] use hard token attacks to force the model to recall sensitive texts under strict conditions. Jeung et al. [65] propose an evaluation framework that mixes forgotten and retained information in a single prompt and checks if the model can separate them.

Some assessments check the structural integrity of knowledge rather than single pieces of information. Qiu et al. [66] use knowledge graphs to verify if the logical links between facts are truly broken. Hsu et al. [67] measure remaining knowledge by checking if the model still recognizes the forgotten text when it is slightly altered. Lang et al. [68] focus on detecting tiny traces of harmful content, while Wichert and Sikdar [69] improves evaluation sensitivity by testing with the most influential data points.

Research has also focused on the model's internal geometry to identify deep traces of data. Cohen et al. [70] introduce REMIND (Residual Memorization In Neighborhood Dynamics) to examine the shape of the model's loss landscape. They show that truly forgotten data looks flat, but retained information looks sharp. To inspect the model more rigorously, Che et al. [71] propose model manipulation attacks. They note that relying solely on prompts often misses harmful capabilities latent in the model. Their experiments show that these concealed capabilities are easy to restore, as a brief phase of LoRA fine-tuning (fewer than 50 steps) can regenerate information that was supposedly removed. To quantify these latent traces before they are exploited, Jeon et al. [72] introduce the Information Difference Index (IDI), a white-box metric designed to quantify residual knowledge within a model's intermediate layers. As illustrated in Figure 10, the metric assesses the cumulative mutual information between features at each layer and the forget label. It calculates the ratio of the information difference in the unlearned model to that of the original model, using a retained model as the baseline. A value close to zero means that the data has been successfully forgotten, while a large value means that there are still residual information within the network.

 doi:10.20944/preprints202603.0114.v1

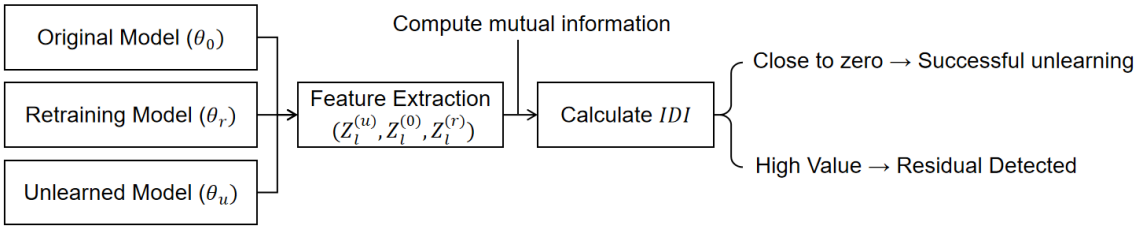

Figure 10. Schematic of the Information Difference Index (IDI) calculation.

4.1.2. Interventional Strategies

While verification validates the immediate elimination of data, interventional strategies prevent its subsequent recovery. These methods prevent the regeneration of previous data and restrict information exposure during external tool usage.

Mitigation of Relearning Risks

Even with strict evaluation, machine unlearning is not always permanent. After a seemingly successful process, LLMs may accidentally recover forgotten knowledge triggered by certain signals such as related questions [73]. Therefore, we need proactive strategies to ensure the unlearning process is robust and the data remains permanently deleted [74,75]. As illustrated in Figure 11, these risks generally stem from two distinct mechanisms: parameter-level vulnerabilities exposed by structural modifications, and inference-time cognitive activation triggered by prompt engineering.

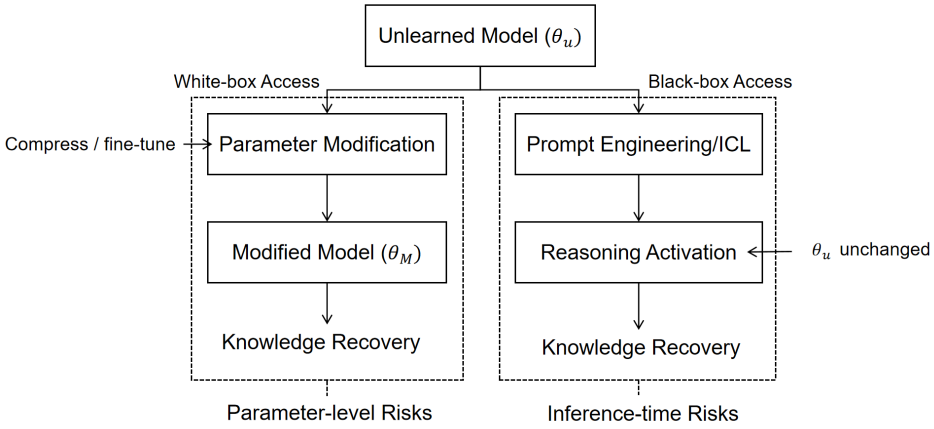

Figure 11. Two Pathways of Relearning Risks.

Specific vulnerabilities exist within each path shown in the figure. To defend against Parameter-level Risks (left), Fan et al. [76] propose SimNPO, which removes the dependency on a reference model to prevent optimization bias, thereby improving robustness against varying unlearning difficulties. Meanwhile, Fan et al. [77] apply sharpness-aware minimization to smooth the model's loss landscape, making the model significantly less likely to be activated by either weight modifications or adversarial inputs. Zhang et al. [78] expose a critical vulnerability: post-hoc quantization of LLMs can restore up to 83% of previously forgotten knowledge. This happens because the minor adjustments made by common unlearning methods (like GA [53] or NPO [57]) are often nullified during compression. To address this, the authors propose SURE (Saliency-Based Unlearning with a Large Learning Rate). This method selectively applies a large learning rate to substantially update specific parts of the model, ensuring the data stays deleted even after compression.

Meanwhile, Inference-time Risks (right) stem from the model's own reasoning capacity. Shumailov et al. [74] identify "UnUnlearning," a phenomenon where specific prompts can induce the unlearned model to reconstruct forbidden information via in-context learning. To mitigate these specific risks, Yuan et al. [39] introduce Latent Adversarial Unlearning (LAU), a strategy that proactively trains

 doi:10.20944/preprints202603.0114.v1

models to maintain robustness against extreme input perturbations. These strategies represent a shift from basic deletion to preventing data reconstruction.

Machine Unlearning in External Tools

LLMs now commonly use external tools to expand their capabilities and solve complex problems. These systems allow models to retrieve information and perform actions using external services like web search or code interpreters, which introduces a new risk of data leakage. Therefore, the challenge of removing knowledge extends to external components.

Cheng and Amiri [79] suggest improving unlearning in these systems by managing both the internal knowledge of the model and the results from external resources. Their work shows that modifying model parameters alone is not enough because external modules can find the unlearned data again. They recommend a combined approach that updates internal parameters and filters data produced by external tools. In the field of code generation, Geng et al. [80] propose a strategy with two parts. It combines basic fine-tuning with a filter that inspects prompts to block sensitive code patterns. Jiang et al. [81] propose PROD (Probabilistic Redistribution for Output Distribution). This is a method specific to code that changes the output probability of the model. It sets the chance of choosing forbidden words to zero and shifts that likelihood to valid alternatives. This preserves code generation capabilities while ensuring specific data is eliminated.

4.2. Summary

We categorize strategies for ensuring completeness into two main types: Frameworks for Verifying Completeness and Interventional Strategies. Beyond standard benchmarks like TOFU and MUSE, researchers now use advanced probing techniques including adversarial attacks and internal geometry checks. Furthermore, new strategies focus on preventing relearning and mitigating data leakage in LLMs that interact with external tools. Table 5 summarizes these key approaches. However, guaranteeing perfect unlearning remains difficult because knowledge is scattered throughout the model. Consequently, it remains hard to prove that data is truly deleted rather than just hidden.

Table 5. Summary of Methodologies for Verifying and Enhancing Completeness.

Sub-category	Basic Ideas	Advantages	Limitations
Frameworks for Verifying Completeness			
Standard Benchmarks [29,30,44]	Uses standardized synthetic datasets and multi-dimensional metrics to rigorously certify data removal	Offers reproducible benchmarks for different models; covers diverse evaluation dimensions instead of simple accuracy	Synthetic data may lack real-world complexity; insufficient to detect deep or latent knowledge traces
Advanced Probing and Deep Verification [64,71,72]	Employs aggressive adversarial attacks or internal state analysis to detect hidden residual knowledge under extreme conditions	Highly sensitive to latent residuals; finds hidden risks missed by standard evaluation	White-box metrics require full parameter access; high computational costs for extensive adversarial probing
Interventional Strategies			
Mitigation of Relearning Risks [39,76,78]	Proactively modifies model parameters to prevent the recovery of forgotten information	Greatly improves the permanence of unlearning; prevents knowledge recovery triggered by prompts or model modifications	Aggressive optimization may degrade general model utility; increases training complexity
Machine Unlearning in External Tools [79–81]	Filters outputs from external tools and RAG systems to prevent displaying sensitive information retrieved from outside sources	Addresses privacy risks across the entire system, not just within the model's parameters; prevents data leakage that occurs when the model accesses external databases	May increase response time due to extra processing steps; keeping the model aligned with external data sources can be challenging

5. Challenge of Efficiency and Cost

As highlighted in Section 2.3.3, computational and financial costs are major barriers for machine unlearning in LLMs. While machine unlearning is generally faster than fully retraining, many un-

 doi:10.20944/preprints202603.0114.v1

learning methods still require complex calculations and heavy hardware resources. If the unlearning process is too slow or expensive, organizations may fail to meet strict legal deadlines. To address this problem, researchers have developed lightweight methods that are much faster than standard approaches. We group these into two categories. **Parameter Space Intervention** applies algebraic operations to directly update the model's weights, avoiding slow training processes. **Interventions During Inference** modify activation patterns or output distributions rather than altering parameters.

5.1. Methodologies for Improving Efficiency and Cost

5.1.1. Parameter Space Intervention

Training deep learning models is usually slow and expensive. These methods take a shortcut by using simple math to edit the weights directly. This facilitates rapid unlearning without iterative optimization.

Task Vector Arithmetic

Task vector methods were originally proposed by Ilharco et al. [82], where a task vector encodes the direction in parameter space corresponding to a specific task. Given a pre-trained model and a fine-tuned model (as shown in the left panel of Figure 12), the task vector is defined as the difference between the fine-tuned weights and the pre-trained weights. As shown in Figure 12, this method enables flexible model editing. The center panel demonstrates *Learning via Addition*, where we sum vectors from different tasks to create a single model capable of handling multiple tasks. Conversely, the right panel depicts *Forgetting via Negation*, where we negate a vector representing unwanted behavior like toxicity to shift the parameters away from the harmful behavior.

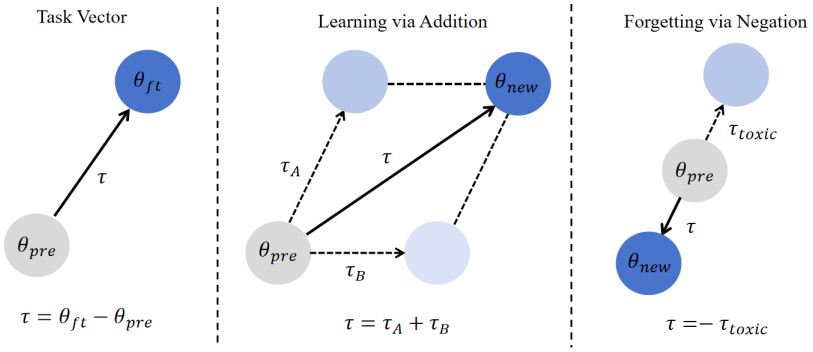

Figure 12. Schematic of Task Vector Arithmetic.

Recent studies have improved the flexibility and precision of task vector operations. Wang et al. [83] extend the vector's reach to push the model further away from harmful tasks, mitigating the impact on unrelated skills during the removal process. Liu et al. [84] refine this process by training specific vectors to represent unwanted content, enhancing the precision of the unlearning process. Additionally, Ni et al. [85] and Jung et al. [86] introduce sequential approaches, where harmful information is first isolated and then removed via subtraction.

5.1.2. Interventions During Inference

Unlike methods that permanently alter the model's static weights, these strategies operate dynamically during the inference process. They adjust internal activations or output probabilities. This approach effectively suppresses specific knowledge during generation, avoiding the high computational costs associated with updating the model structure.

Representation Space-based Methods

Representation Space-based Machine Unlearning focuses on modifying the model's internal activity during the forward pass rather than changing its permanent weights. Figure 13 depicts this geometric process. The original activation is projected onto a subspace orthogonal to the sensitive direction, effectively isolating and removing the sensitive component while retaining utility.

 doi:10.20944/preprints202603.0114.v1

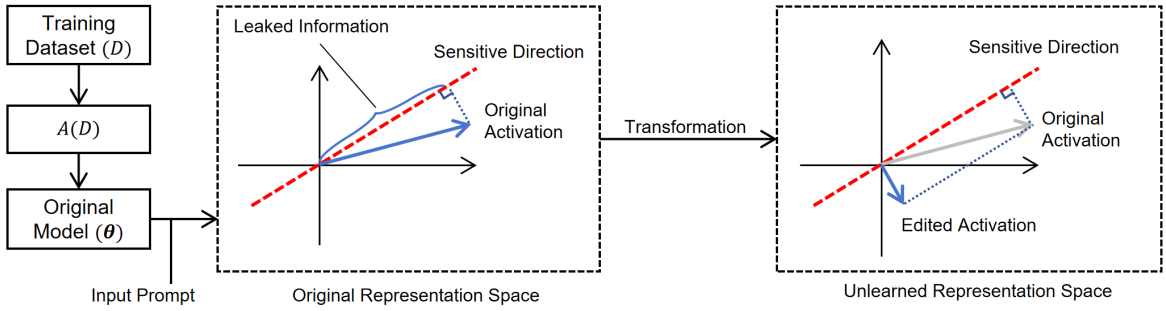

Figure 13. Mechanism of Representation Space-based Unlearning

Belrose et al. [87] propose LEACE (LEAst-squares Concept Erasure) to erase specific concepts directly from the representation space. This method calculates a transformation that removes the target concept while preserving the original data structure. Specifically, it solves a constrained optimization problem to minimize the deviation between the transformed and original representations. The key constraint is that the covariance between the output and the target concept must be zero. This guarantee ensures that linear classifiers cannot recover the erased concept.

Recent studies have developed more dynamic strategies to improve precision. Ren et al. [88] introduce GRUN (Gated Representation UNlearning), which uses a gating mechanism to restrict modifications to specific tokens, preserving general performance. Similarly, He et al. [89] propose DeepCUT, which actively reshapes the model's internal geometry. It pushes the representations of unwanted data away from their original meaning and breaks the model's ability to recall specific concepts.

Neuron-level Activation Editing

Neuron-level Activation Editing operates at a fine scale by modifying individual neurons. This strategy is highly efficient as it requires minimal additional computation. Figure 14 illustrates the general workflow. The process consists of two main steps: first, identifying sensitive neurons responsible for specific knowledge; second, applying a masking mechanism during inference to suppress their output. The model weights remain frozen throughout this process.

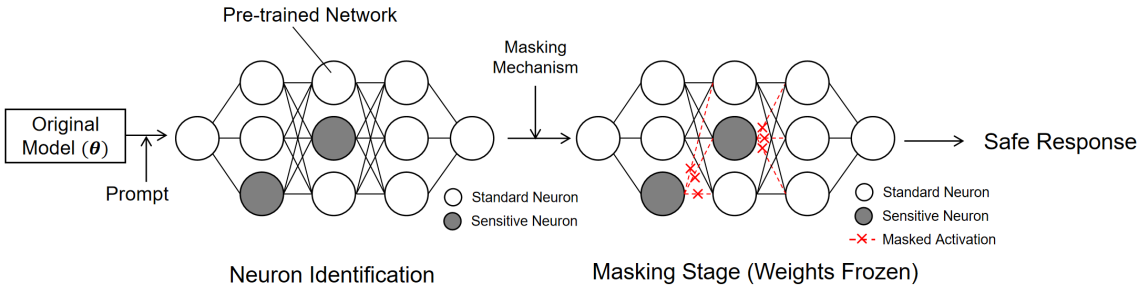

Figure 14. Schematic of Neuron-level Activation Editing.

Wu et al. [90] propose DEPN (Detecting and Editing Privacy Neurons), a post-hoc unlearning method designed to locate and suppress specific neurons that store private information. The method first determines the importance of each neuron by computing a privacy attribution score based on integrated gradients. The score evaluates how much a neuron affects the probability of generating the private sequence. Specifically, the attribution sums activation gradients to measure the neuron's contribution to generating the text. To eliminate private information, the method deactivates neurons during inference that are identified according to these scores.

Pochinkov and Schoots [91] introduce selective a pruning method which targets neurons using activation statistics derived from both forget and retain datasets. It then sets the weights of the units with the lowest scores to zero. Liu et al. [92] further validate that such sensitivity scores can guide

 doi:10.20944/preprints202603.0114.v1

selective removal in broader architectures like Multimodal Large Language Models (MLLM). Moving beyond binary suppression, Li et al. [93] propose a finer intervention called Neuron Adjust. This approach identifies units linked to specific tasks and adjusts their output values instead of disabling them completely. This strategy shows that adjusting values offers a gentler alternative to hard pruning.

Sparse Autoencoder Feature Editing

Sparse autoencoder (SAE) feature editing translates complex model signals into clear and separate features [94]. As illustrated in Figure 15, the method eliminates specific concepts by forcing their activation values to a negative number. Specifically, if a target feature is active, its value is clamped to a fixed negative constant, effectively steering the model away from the sensitive concept. Then, the decoder turns these features back into dense activations and returns them to the model. Khoriaty et al. [95] introduce a method that conditionally modifies activations based on semantic criteria, eliminating unwanted information more effectively. Karvonen et al. [96] validate these methods using a new testing method called SAEBench. Their experiments demonstrate that optimized SAE architectures are particularly effective.

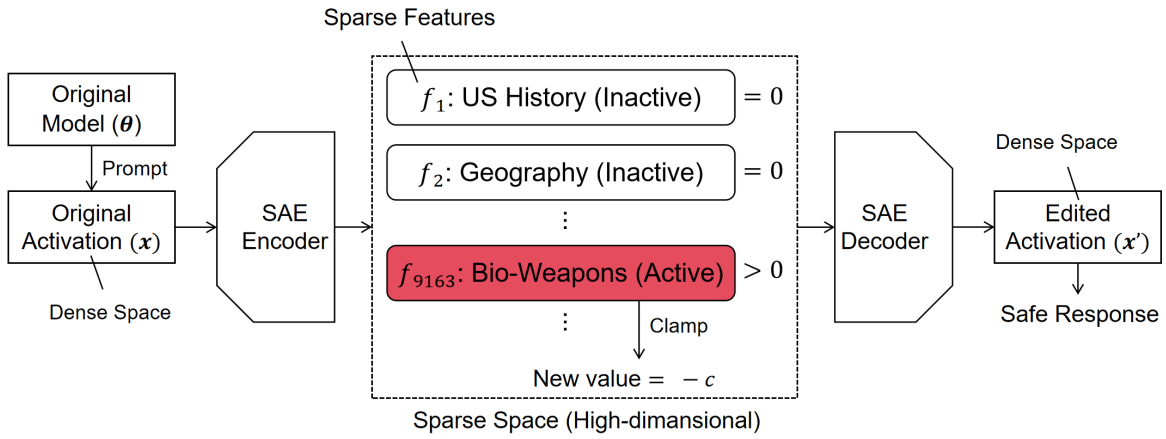

Figure 15. Schematic of Sparse Autoencoder Feature Editing.

Logit-Level Strategies

Logit-level strategies focus on the output layer and use the prediction probabilities to guide machine unlearning during inference. Ji et al. [36] propose Unlearning from Logit Difference (ULD). As shown in Figure 16, this method employs a small assistant model trained to memorize unwanted data. During inference, the framework subtracts the assistant model's output from the original model. Specifically, it calculates the final logits by taking the original model's logits and subtracting the scaled assistant model's logits. This process effectively suppresses harmful tokens while preserving safe ones. Suriyakumar et al. [97] introduce Unlearning via Contrastive Decoding (UCD) using two assistant models: a corrupted model trained on forget data and a clean model trained on retain data. It first calculates a contrastive vector by computing the difference between the log-probabilities of the corrupted and clean models. This difference captures the specific knowledge found only in the corrupted model. This value is then applied to adjust the target model's output by subtracting this contrastive difference from the corrupted model's logits. Similarly, Wu et al. [98] and Huang et al. [99] use a small external model to calculate a correction vector. By adding this vector to the main model's prediction scores, these approaches prevent the generation of sensitive information.

 doi:10.20944/preprints202603.0114.v1

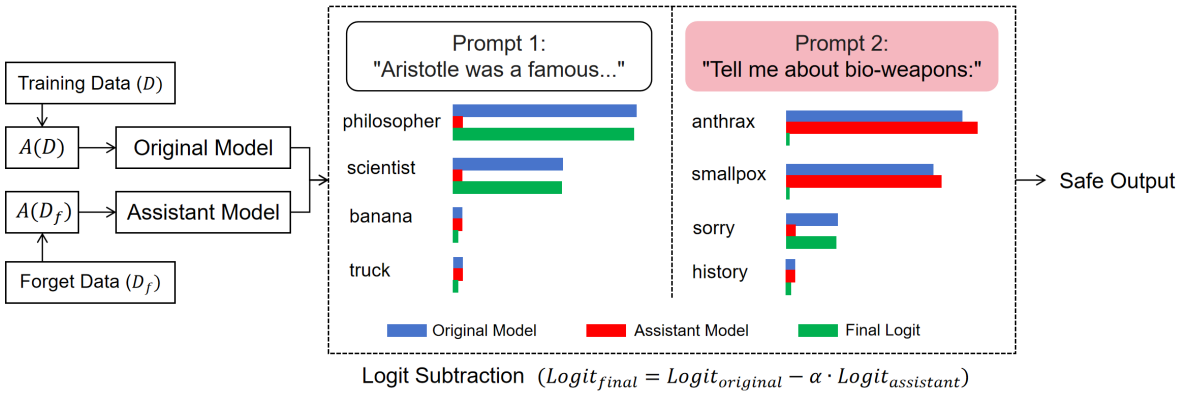

Figure 16. Schematic of Unlearning from Logit Difference (ULD).

ULD and UCD apply a constant adjustment at every step, which may cause unnecessary cost and risks if the assistant makes mistakes on safe text. Deng et al. [100] propose Generation-time Unlearning via Adaptive Restriction and Detection (GUARD), which identifies forbidden tokens and applies penalties only when these tokens appear. The penalty is defined such that infinite penalties are assigned to forbidden tokens when risk is high, while no penalty is applied otherwise. The method calculates a generation cost by adding this penalty to the model's negative log-probability. This adaptive strategy ensures that forbidden tokens are effectively blocked only when necessary, which preserves general capabilities.

5.2. Summary

We categorize strategies for improving machine unlearning efficiency into two primary categories: **Parameter Space Intervention** and **Inference-Time Intervention**. Parameter Space Interventions, like Task Vectors, achieve unlearning by arithmetic operations on the model's static weights. Inference-Time Interventions modify the forward pass or the final output. Techniques like representation editing, neuron suppression, sparse autoencoder feature clamping, and logit-level strategies provide efficient ways to remove specific knowledge from a model. These strategies show that effective unlearning is achievable without the high resource demands of optimization based on gradient descent. Table 6 summarizes representative machine unlearning methods discussed in this section. However, it is still very challenging to balance efficiency with unlearning quality. Since these methods typically do not fine-tune modify parameters, they often struggle to completely remove deeply embedded knowledge.

Table 6. Summary of Methodologies for Improving Efficiency and Cost.

Sub-category	Basic Ideas	Advantages	Limitations
Parameter Space Intervention			
Task Vector Arithmetic [82–84]	Encodes task knowledge into vectors and performs arithmetic operations directly on model weights to remove information	Updates models instantly via simple subtraction; Allows users to combine multiple unlearning tasks easily	Requires a fine-tuned reference model to calculate the vector; risk of degrading unrelated skills when knowledge is intertwined

 doi:10.20944/preprints202603.0114.v1

Table 6. *Cont.*

Sub-category	Basic Ideas	Advantages	Limitations
Interventions During Inference			
Representation Space-based Unlearning [87–89]	Projecting activations onto a subspace orthogonal to the target concept	Preserves the original model weights; Theoretical guarantee for linear erasure	Effectiveness decreases when data associations are non-linear
Neuron-level Activation Editing [90,91,93]	Identifies specific neurons responsible for sensitive knowledge and suppresses these neurons based on importance scores	Minimal computational cost during inference; Enables precise control at the individual neuron level	Risk of damaging polysemantic neurons; Incomplete removal occurs if knowledge is distributed across many neurons
Sparse Autoencoder Feature Editing [94–96]	Decomposes dense activations into interpretable sparse features; Blocks features related to sensitive concepts	High interpretability of internal behaviors; Effectively steers generation away from specific concepts	Additional inference cost due to encoder-decoder steps; relies on the quality of feature decomposition
Logit-Level Strategies [36,97,100]	Modifies final output probabilities; Subtracts logits from an assistant model or applies specific penalties	Applicable to diverse models; Effectively blocks the verbatim generation of sensitive tokens	Running assistant models causes significant inference latency; Performance depends on the alignment of the assistant model

6. Challenge of Black-Box Constraints

LLMs operate as black-boxes where users interact solely through restricted interfaces like APIs. This creates a major barrier for machine unlearning. Since developers lack access to internal parameters or gradients, effective techniques like fine-tuning or activation editing are infeasible. Intervention methods operate as external control layers, manipulating the data flow to and from the model. These are typically classified by their point of application: **Input-Side Interventions**, which preemptively modify prompts to prevent knowledge retrieval, and **Output-Side Interventions**, which filter responses post-generation.

6.1. Methodologies for Black-Box Intervention and Auditing

6.1.1. Input-Side Intervention

Input-side interventions operate by strategically modifying the query or context before it is processed by the model. By embedding specific instructions or examples into the prompt, these methods steer the generation process away from sensitive information. This ensures that the model is guided to produce safe responses without requiring access to its internal parameters. A representative approach in this category is In-Context Unlearning (ICUL) introduced by Pawelczyk et al. [101], which leverages the concept of In-Context Learning (ICL) [21]. As illustrated in Figure 17, this method constructs in-context prompts to suppress the influence of the original memory. It first performs label flipping on the target forget data. Then, it incorporates correctly labeled retain data to maintain the model's general capabilities. For example, to make the model "forget" a specific input in a classification task, the system would present that input with a flipped label (e.g., changing "Positive" to "Neutral") followed by normal examples. This constructed context effectively tricks the model into suppressing the sensitive information.

 doi:10.20944/preprints202603.0114.v1

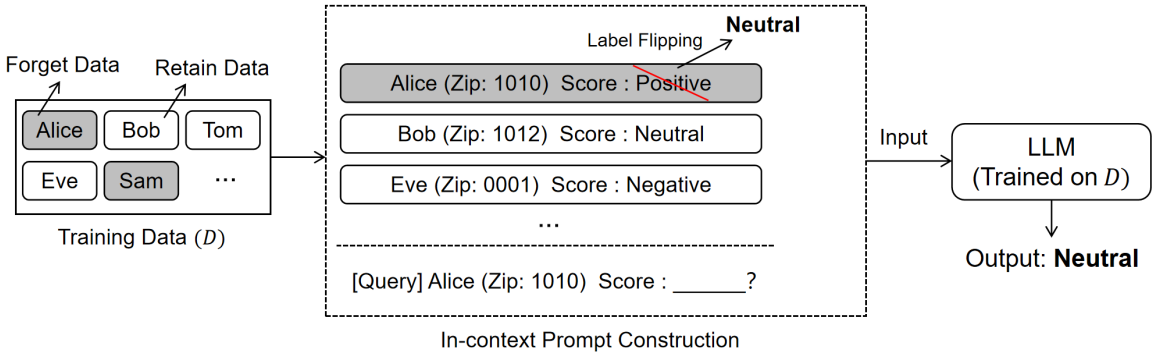

Figure 17. Schematic of In-Context Unlearning (ICUL).

Beyond simple label flipping, other studies employ advanced prompting strategies. Gallegos et al. [102] propose zero-shot self-debiasing. This method instructs the model to evaluate its own initial response for biases. The model then repairs the output itself through reprompting. This achieves unlearning without extra training. Similarly, Sanyal and Mandal [103] introduce a multi-agent framework. In this system, separate agents collaborate to critique and rewrite prompts. They identify and remove sensitive information before the model processes the text. For more formal safety guarantees, Muresanu et al. [104] propose ERASE (Efficient Removal And Selection of Examples). This method selects context examples using quantized k-means clustering.

In addition to prompt manipulation, some methods introduce external guardrails or architectural adjustments at the input level. Thaker et al. [105] establish strict guardrails by combining specific prompt templates with explicit filtering methods. This system automatically rejects any answer that fails to meet privacy requirements. SERAC (Semi-Parametric Editing with a Retrieval-Augmented Counterfactual Model) [106] takes a memory-based approach. It routes inputs to a counterfactual model. This model replaces the original output with a corrected response from a cache of user edits. Similarly, Wang et al. [107] propose a lightweight method specifically for Retrieval-Augmented Generation (RAG) systems. This technique updates the external knowledge base with confidentiality rules. It then retrieves constraints that direct the model to suppress target knowledge without internal access. Finally, operating at the embedding level, Liu et al. [37] explore embedding corruption. This technique adds noise to specific token embeddings. This disruption prevents the model from retrieving specific memories while maintaining fluent output.

6.1.2. Output-Side Intervention

Output-side interventions function as a verification layer that processes the text generated by the model. These strategies analyze the output stream to detect and intercept sensitive content. By filtering or regenerating harmful segments, they ensure compliance with safety standards before the response is presented to the user. A representative approach in this category is C-SafeGen introduced by Kang et al. [108]. This method uses a specific decoding algorithm called Claim-based Stream Decoding (CSD). As illustrated in Figure 18, this approach monitors the generation process in real time. It divides the output text into meaningful units called claims. If the system identifies a risky claim, it backtracks to remove the unsafe content. It then generates a new and safe path. This ensures safety without simply refusing to answer the user.

Other studies use different strategies to achieve similar goals. For example, Wu et al. [109] introduce PSG-Agent. This framework incorporates a guard agent within the pipeline. It applies safety rules based on user profile to ensure the output is appropriate for the individual user.

 doi:10.20944/preprints202603.0114.v1

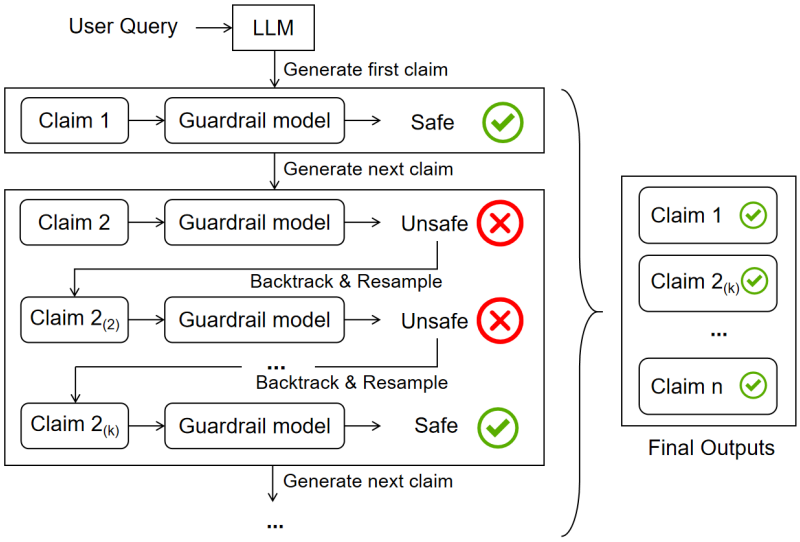

Figure 18. Schematic of C-SafeGen's Claim-based Stream Decoding.

6.2. Summary

We categorized solutions for black-box machine unlearning in LLMs into two main groups: Input-Side Intervention and Output-Side Intervention. Input-Side Interventions use specific context or instructions to stop the model from generating unwanted information. Output-Side Interventions focus on the generated results, applying filters or verifications to ensure that sensitive content is removed before being shown to the user. Table 7 summarizes representative approaches discussed in this section. However, these approaches have a major limitation. Since we cannot actually change the model itself, the sensitive information is only hidden rather than truly deleted. As a result, the data still exists within the network, making it difficult to guarantee that it cannot be recovered.

Table 7. Summary of Methodologies for Black-Box Intervention and Auditing.

Sub-category	Basic Ideas	Advantages	Limitations
Input-Side Intervention [37,101,105,106]	Sanitizes inputs using defensive prompts, external rules, or corrective agents	Deploys rapidly without architectural changes; Minimizes impact on response style	Remains vulnerable to prompt injection; Consumes context window capacity; Incurs extra costs for processing defensive prompts
Output-Side Intervention [108,109]	Monitors decoding streams to backtrack and regenerate unsafe claims. Verifies final outputs using auxiliary classifiers or guard agents to block harmful content	Operates independently from the main model; achieves high precision on specific safety categories; Acts as a backup safety net for bypassed inputs	Causes false alarms due to missing context; wastes computation on blocked outputs; Adds post-processing latency

7. Future Directions

The methods discussed in the previous sections provide partial solutions to four key challenges in LLM unlearning: performance degradation, unlearning completeness, computational efficiency, and black-box constraints. However, these methods don't fully solve the core tension between LLMs' plasticity and its stability. To reach the next level of reliability, we need to look beyond traditional boundaries. To build frameworks that are mathematically verifiable and inherently robust, future research should look to physics, cognitive science, and game theory for answers.

 doi:10.20944/preprints202603.0114.v1

7.1. Performance Degradation

Current unlearning methods primarily rely on adversarial optimization within the parameter space. Recent theoretical research indicates that LLMs exhibit an intrinsic elasticity, causing them to revert to pre-training distributions and resist post-training interventions [110]. Consequently, the repeated use of these adversarial techniques degrades model plasticity, leading to irreversible rigidity in continuous unlearning settings [111,112].

To address this, future research should advance toward an inverse alignment paradigm. Rather than simply modifying weights, it leverages the model's intrinsic mechanics to reverse the learning process. This requires algorithms capable of calculating the precise trajectory of model deformation. The application of a precise opposing force restores the model to an unlearned state along the most efficient path. Alternatively, unlearning can be viewed as the reverse of data compression, which may effectively remove traces of knowledge [110]. This perspective transforms unlearning from blind parameter tuning into the precise control of model dynamics.

Beyond mere parameter updates, a broader research horizon emerges in cognitive and contextual alignment. Future unlearning strategies should apply cognitive science principles to simulate human-like forgetting. The objective is contextual suppression, which blocks sensitive information within specific contexts while preserving general understanding. These strategies aim to establish the psychological plausibility of unlearning behavior. [47,113,114]. Ultimately, this strategy prioritizes psychological realism over simple statistics.

7.2. Unlearning Completeness

While benchmarks like TOFU, MUSE, and WMDP provide a starting point, they often fail to catch latent degradation [29,30,44]. More fundamentally, verifying exact unlearning in LLMs is computationally infeasible due to the immense complexity. Thus, establishing a universal, perfect verifier within reasonable time limits is impossible [115]. Experiments also show that current methods are fragile under relearning attacks, which indicates a high risk of latent knowledge retention [72,116].

To overcome these challenges, verification standards must focus on formal proofs and information-theoretic bounds. For instance, building frameworks that minimize mutual information is a promising direction. The aim is to reduce the mutual information between unlearned data and model weights to a theoretical minimum. Additionally, phase transition theory can predict unlearning difficulty based on model scale [117]. Designs like recursive sketching and freezing protocols provide provable unlearning with differential privacy guarantees while meeting complexity constraints [115].

At the same time, game theory represents a promising frontier for future research. The unlearning process can be modeled as a Stackelberg game [38]. Algorithms must prove they maintain a Nash equilibrium against the "strongest auditors." These auditors use advanced methods like membership inference or reverse engineering [118]. This adversarial paradigm compels unlearning technologies to possess active defense capabilities. This establishes a completeness standard that is more rigorous than simple behavioral testing.

7.3. Efficiency and Cost

LLM unlearning still faces severe cost and efficiency challenges. From a physics perspective, unlearning reverses the entropy reduction achieved during training. This results in a sharp increase in system entropy. Simultaneously, calculating precise parameter updates creates unsustainable computational complexity, as it requires balancing the conflicting goals of forgetting and retaining information [119]. These distinct thermodynamic and computational costs cause heavy resource burdens, making traditional optimization methods impractical for large-scale LLMs.

To address this, we can adopt a thermodynamic approach to entropy management. Unlearning can be regarded as maximizing entropy within a specific set of constraints [119]. This allows for effective forgetting without high costs of exact gradient calculations. However, this may inherently challenge the linguistic order. Therefore, future research should not just focus on data erasure. Instead,

 doi:10.20944/preprints202603.0114.v1

it needs to balance between the entropy of forgetting and the structure of retained knowledge [120]. As models move toward Federated Learning and decentralized Model-as-a-Service (MaaS), unlearning becomes an economic challenge involving multiple participants [121]. Future work must build an ecosystem that motivates users and servers to handle costly unlearning tasks. This perspective treats unlearning not just as an algorithmic task, but as a resource allocation game within a larger AI supply chain [122,123].

7.4. Black-Box Constraints

In black-box settings, unlearning is particularly vulnerable. Standard methods like output filtering are easily bypassed by indirect attacks such as synonym substitution, translation exploits, or malicious code completion [17,124,125]. In distributed systems, the lack of transparent logs allows providers to tamper with unlearning records. Furthermore, the verification process itself poses a risk of privacy leakage [126,127].

To address these trust gaps, researchers should integrate modern cryptographic technologies, such as Zero-Knowledge Proofs (ZKPs) [128,129]. Future work should focus on developing verifiable unlearning frameworks. By transforming unlearning algorithms into arithmetic circuits, providers can generate concise cryptographic proofs. This allows users to mathematically verify the compliance of the unlearning process without accessing sensitive model parameters [130]. Additionally, decentralized verification mechanisms that integrate blockchain and secure multi-party computation are becoming essential. Implementing encrypted markers and immutable audit logs ensures full traceability and tamper resistance in distributed settings like federated learning [131]. It offers robust technical guarantees for data sovereignty in black-box environments [132].

8. Conclusions

The rapid development of LLMs has highlighted critical data privacy and ethical risks. Machine unlearning has emerged as an important solution for ensuring legal compliance and responsible deployment. This survey systematically reviews the current landscape of machine unlearning in LLMs around four core challenges: performance degradation, unlearning completeness, efficiency and cost, and black-box constraints.

Significant progress has been made in addressing these hurdles. Researchers have proposed diverse methodologies, ranging from white-box strategies that precisely update model parameters to black-box interventions that use external filters. Crucially, the field has also established rigorous verification frameworks. These frameworks not only evaluate whether data is truly removed but also assess the preservation of general model utility. These advancements demonstrate that it is feasible to unlearn specific information while keeping the impact on the model's capabilities within an acceptable range. Moving forward, there is a critical need for more reliable and effective solutions. The focus is expanding to include proactive, verifiable, and robust systems. Ultimately, machine unlearning is evolving into a critical safeguard for the safety of future AI development.

Author Contributions: Conceptualization, X.T., T.Z. and P.X.; investigation, X.T. and Z.L.; writing—original draft preparation, X.T.; writing—review and editing, T.Z., P.X. and W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

 doi:10.20944/preprints202603.0114.v1

References

1. OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* **2023**, [2303.08774].
2. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2019), 2019, pp. 4171–4186. https://doi.org/10.18653/V1/N19-1423.
3. Anil, R.; et al. PaLM 2 Technical Report. *arXiv preprint arXiv:2305.10403* **2023**, [2305.10403].
4. DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948* **2025**, [2501.12948].
5. Feldman, V. Does Learning Require Memorization? A Short Tale About a Long Tail. In Proceedings of the ACM SIGACT Symposium on Theory of Computing (2020), 2020, pp. 954–959. https://doi.org/10.1145/3357713.3384290.
6. Carlini, N.; Liu, C.; Erlingsson, Ú.; Kos, J.; Song, D. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In Proceedings of the USENIX Security Symposium (2019), 2019, pp. 267–284.
7. General Data Protection Regulation (GDPR), 2018. Online.
8. California Consumer Privacy Act (CCPA), 2018. Online.
9. Japan - Data Protection Overview(JDPO), 2019. Online.
10. Consumer Privacy Protection Act (CPPA), 2022. Online.
11. Cao, Y.; Yang, J. Towards Making Systems Forget with Machine Unlearning. In Proceedings of the IEEE Symposium on Security and Privacy (2015), 2015, pp. 463–480. https://doi.org/10.1109/SP.2015.35.
12. Liu, S.; Yao, Y.; Jia, J.; Casper, S.; Baracaldo, N.; Hase, P.; Yao, Y.; Liu, C.Y.; Xu, X.; Li, H.; et al. Rethinking Machine Unlearning for Large Language Models. *Nature Machine Intelligence* **2025**, *7*, 181–194. https://doi.org/10.1038/S42256-025-00985-0.
13. Xu, H.; Zhu, T.; Zhang, L.; Zhou, W.; Yu, P.S. Machine Unlearning: A Survey. *ACM Computing Surveys* **2024**, *56*, 9:1–9:36. https://doi.org/10.1145/3603620.
14. Liu, H.; Xiong, P.; Zhu, T.; Yu, P.S. A Survey on Machine Unlearning: Techniques and New Emerged Privacy Risks. *Journal of Information Security and Applications* **2025**, *90*, 104010. https://doi.org/10.1016/J.JISA.2025.104010.
15. Wang, S.; Zhu, T.; Liu, B.; Ding, M.; Ye, D.; Zhou, W.; Yu, P.S. Unique Security and Privacy Threats of Large Language Models: A Comprehensive Survey. *ACM Computing Surveys* **2026**, *58*, 83:1–83:36. https://doi.org/10.1145/3764113.
16. Eldan, R.; Russinovich, M. Who's Harry Potter? Approximate Unlearning in LLMs. *arXiv preprint arXiv:2310.02238* **2023**, [2310.02238].
17. Doshi, J.; Stickland, A.C. Does Unlearning Truly Unlearn? A Black Box Evaluation of LLM Unlearning Methods. *arXiv preprint arXiv:2411.12103* **2024**, [2411.12103].
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems (2017), 2017, pp. 5998–6008.
19. Tian, C.; Qin, X.; Tam, K.; Li, L.; Wang, Z.; Zhao, Y.; Zhang, M.; Xu, C. CLONE: Customizing LLMs for Efficient Latency-Aware Inference at the Edge. In Proceedings of the USENIX Annual Technical Conference (2025), 2025, pp. 563–585.
20. Bai, Y.; Mei, J.; Yuille, A.L.; Xie, C. Are Transformers More Robust Than CNNs? In Proceedings of the Advances in Neural Information Processing Systems 34 (2021), 2021, pp. 26831–26843.
21. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. In Proceedings of the Advances in Neural Information Processing Systems (2020), 2020.
22. Gundla, K.; Martha, S.; Panigrahy, A.K. Toward Explainable AI in Satellite Imagery: A ResNet-50-Based Study on EuroSAT Classification. *IEEE Access* **2025**, *13*, 165900–165908. https://doi.org/10.1109/ACCESS.2025.3612014.
23. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In Proceedings of the European Conference on Computer Vision (2014), 2014, pp. 818–833. https://doi.org/10.1007/978-3-319-10590-1_53.

 doi:10.20944/preprints202603.0114.v1

24. Liu, M.; Shi, J.; Li, Z.; Li, C.; Zhu, J.; Liu, S. Towards Better Analysis of Deep Convolutional Neural Networks. *IEEE Transactions on Visualization and Computer Graphics* **2017**, *23*, 91–100. https://doi.org/10.1109/TVCG.2016.2598831.
25. Clark, K.; Khandelwal, U.; Levy, O.; Manning, C.D. What Does BERT Look at? An Analysis of BERT's Attention. In Proceedings of the ACL Workshop BlackboxNLP(2019), 2019, pp. 276–286. https://doi.org/10.18653/V1/W19-4828.
26. Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C.A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; Papernot, N. Machine Unlearning. In Proceedings of the IEEE Symposium on Security and Privacy (2021), 2021, pp. 141–159. https://doi.org/10.1109/SP40001.2021.00019.
27. Chen, K.; Huang, Y.; Wang, Y. Machine Unlearning via GAN. *arXiv preprint arXiv:2111.11869* **2021**.
28. Warnecke, A.; Pirch, L.; Wressnegger, C.; Rieck, K. Machine Unlearning of Features and Labels. In Proceedings of the Network and Distributed System Security Symposium (2023), 2023. https://doi.org/10.14722/ndss.2023.23087.
29. Maini, P.; Feng, Z.; Schwarzschild, A.; Lipton, Z.C.; Kolter, J.Z. TOFU: A Task of Fictitious Unlearning for LLMs. In Proceedings of the Advances in Neural Information Processing Systems (2024), 2024.
30. Shi, W.; Lee, J.; Huang, Y.; Malladi, S.; Zhao, J.; Holtzman, A.; Liu, D.; Zettlemoyer, L.; Smith, N.A.; Zhang, C. MUSE: Machine Unlearning Six-Way Evaluation for Language Models. In Proceedings of the International Conference on Learning Representations (2025), 2025.
31. Blanco-Justicia, A.; Jebreel, N.; Manzanares-Salor, B.; Sánchez, D.; Domingo-Ferrer, J.; Collell, G.; Tan, K.E. Digital Forgetting in Large Language Models: A Survey of Unlearning Methods. *Artificial Intelligence Review* **2025**, *58*, 90. https://doi.org/10.1007/S10462-024-11078-6.
32. Łucki, J.; Wei, B.; Huang, Y.; Henderson, P.; Tramèr, F.; Rando, J. An Adversarial Perspective on Machine Unlearning for AI Safety. In Proceedings of the NeurIPS 2024 Workshop on Red Teaming GenAI: What Can We Learn from Adversaries?, 2024.
33. Zhang, J.; Sun, J.; Yeats, E.; Ouyang, Y.; Kuo, M.; Zhang, J.; Yang, H.F.; Li, H. Min-K%++: Improved Baseline for Pre-Training Data Detection from Large Language Models. In Proceedings of the The Thirteenth International Conference on Learning Representations (2025), 2025.
34. Rezaei, K.; Chandu, K.R.; Feizi, S.; Choi, Y.; Brahman, F.; Ravichander, A. RESTOR: Knowledge Recovery in Machine Unlearning. *Transactions on Machine Learning Research* **2025**.
35. Cooper, A.F.; Choquette-Choo, C.A.; Bogen, M.; Jagielski, M.; Filippova, K.; Liu, K.Z.; Chouldechova, A.; Hayes, J.; Huang, Y.; Mireshghallah, N.; et al. Machine Unlearning Doesn't Do What You Think: Lessons for Generative AI Policy, Research, and Practice. In Proceedings of the Advances in Neural Information Processing Systems (2025), 2025.
36. Ji, J.; Liu, Y.; Zhang, Y.; Liu, G.; Kompella, R.; Liu, S.; Chang, S. Reversing the Forget-Retain Objectives: An Efficient LLM Unlearning Framework from Logit Difference. In Proceedings of the Advances in Neural Information Processing Systems (2024), 2024.
37. Liu, C.; Wang, Y.; Flanigan, J.; Liu, Y. Large Language Model Unlearning via Embedding-Corrupted Prompts. In Proceedings of the Advances in Neural Information Processing Systems (2024), 2024.
38. Liu, H.; Zhu, T.; Zhang, L.; Xiong, P. Game-Theoretic Machine Unlearning: Mitigating Extra Privacy Leakage. *IEEE Transactions on Information Forensics and Security* **2025**, *20*, 11591–11606. https://doi.org/10.1109/TIFS.2025.3623364.
39. Yuan, H.; Jin, Z.; Cao, P.; Chen, Y.; Liu, K.; Zhao, J. Towards Robust Knowledge Unlearning: An Adversarial Framework for Assessing and Improving Unlearning Robustness in Large Language Models. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence (2025), 2025, pp. 25769–25777. https://doi.org/10.1609/AAAI.V39I24.34769.
40. Ren, J.; Xing, Y.; Cui, Y.; Aggarwal, C.C.; Liu, H. SoK: Machine Unlearning for Large Language Models. *arXiv preprint arXiv:2506.09227* **2025**, [2506.09227].
41. Gu, T.; Huang, K.; Luo, R.; Yao, Y.; Yang, Y.; Teng, Y.; Wang, Y. MEOW: MEMOry Supervised LLM Unlearning Via Inverted Facts. *arXiv preprint arXiv:2409.11844* **2024**, [2409.11844].
42. Cha, S.; Cho, S.; Hwang, D.; Lee, M. Towards Robust and Parameter-Efficient Knowledge Unlearning for LLMs. In Proceedings of the The Thirteenth International Conference on Learning Representations (ICLR 2025), 2025.
43. Ding, C.; Wu, J.; Yuan, Y.; Lu, J.; Zhang, K.; Su, A.; Wang, X.; He, X. Unified Parameter-Efficient Unlearning for LLMs. In Proceedings of the International Conference on Learning Representations (2025), 2025.

 doi:10.20944/preprints202603.0114.v1

44. Li, N.; Pan, A.; Gopal, A.; Yue, S.; Berrios, D.; Gatti, A.; Li, J.D.; Dombrowski, A.; Goel, S.; Mukobi, G.; et al. The WMDP Benchmark: Measuring and Reducing Malicious Use with Unlearning. In Proceedings of the International Conference on Machine Learning (2024), 2024.
45. Dang, H.; Pham, T.; Thanh-Tung, H.; Inoue, N. On Effects of Steering Latent Representation for Large Language Model Unlearning. In Proceedings of the AAAI Conference on Artificial Intelligence (2025), 2025, pp. 23733–23742. https://doi.org/10.1609/AAAI.V39I22.34544.
46. Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; Wei, F. Knowledge Neurons in Pretrained Transformers. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (2022), 2022, pp. 8493–8502. https://doi.org/10.18653/V1/2022.ACL-LONG.581.
47. Yang, N.; Kim, M.; Yoon, S.; Shin, J.; Jung, K. FaithUn: Toward Faithful Forgetting in Language Models by Investigating the Interconnectedness of Knowledge. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, 2025, pp. 12999–13014. https://doi.org/10.18653/v1/2025.emnlp-main.657.
48. Jia, J.; Liu, J.; Zhang, Y.; Ram, P.; Baracaldo, N.; Liu, S. WAGLE: Strategic Weight Attribution for Effective and Modular Unlearning in Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems (2024), 2024.
49. Hou, L.; Wang, Z.; Liu, G.; Wang, C.; Liu, W.; Peng, K. Decoupling Memories, Muting Neurons: Towards Practical Machine Unlearning for Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics (2025), 2025, pp. 13978–13999. https://doi.org/10.18653/v1/2025.findings-acl.719.
50. Yu, C.; Jeoung, S.; Kasi, A.; Yu, P.; Ji, H. Unlearning Bias in Language Models by Partitioning Gradients. In Proceedings of the Findings of the Association for Computational Linguistics: ACL (2023), 2023, pp. 6032–6048. https://doi.org/10.18653/v1/2023.findings-acl.375.
51. Shen, W.F.; Qiu, X.; Kurmanji, M.; Iacob, A.; Sani, L.; Chen, Y.; Cancedda, N.; Lane, N.D. LLM Unlearning via Neural Activation Redirection. In Proceedings of the Advances in Neural Information Processing Systems (2025), 2025.
52. Hu, J.; Huang, Z.; Yin, X.; Ruan, W.; Cheng, G.; Dong, Y.; Huang, X. FALCON: Fine-grained Activation Manipulation by Contrastive Orthogonal Unalignment for Large Language Model. In Proceedings of the Advances in Neural Information Processing Systems (2025), 2025.
53. Jang, J.; Yoon, D.; Yang, S.; Cha, S.; Lee, M.; Logeswaran, L.; Seo, M. Knowledge Unlearning for Mitigating Privacy Risks in Language Models. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (2023), 2023, pp. 14389–14408. https://doi.org/10.18653/V1/2023.ACL-LONG.805.
54. Fan, C.; Liu, J.; Zhang, Y.; Wong, E.; Wei, D.; Liu, S. SalUn: Empowering Machine Unlearning via Gradient-Based Weight Saliency in Both Image Classification and Generation. In Proceedings of the International Conference on Learning Representations (2024), 2024.
55. Jia, J.; Liu, J.; Ram, P.; Yao, Y.; Liu, G.; Liu, Y.; Sharma, P.; Liu, S. Model Sparsity Can Simplify Machine Unlearning. In Proceedings of the Advances in Neural Information Processing Systems (2023), 2023.
56. Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C.D.; Ermon, S.; Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In Proceedings of the Advances in Neural Information Processing Systems (2023), 2023.
57. Zhang, R.; Lin, L.; Bai, Y.; Mei, S. Negative Preference Optimization: From Catastrophic Collapse to Effective Unlearning. In Proceedings of the First Conference on Language Modeling, 2024.
58. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training Language Models to Follow Instructions with Human Feedback. In Proceedings of the Advances in Neural Information Processing Systems (2022), 2022.
59. Lee, H.; Phatale, S.; Mansoor, H.; Mesnard, T.; Ferret, J.; Lu, K.; Bishop, C.; Hall, E.; Carbune, V.; Rastogi, A.; et al. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. In Proceedings of the International Conference on Machine Learning (2024), 2024.
60. Yuan, H.; Yuan, Z.; Tan, C.; Wang, W.; Huang, S.; Huang, F. RRHF: Rank Responses to Align Language Models with Human Feedback. In Proceedings of the Advances in Neural Information Processing Systems (2023), 2023.
61. Lu, X.; Welleck, S.; Hessel, J.; Jiang, L.; Qin, L.; West, P.; Ammanabrolu, P.; Choi, Y. QUARK: Controllable Text Generation with Reinforced Unlearning. In Proceedings of the Advances in Neural Information Processing Systems (2022), 2022.

 doi:10.20944/preprints202603.0114.v1

62. Dong, Y.R.; Lin, H.; Belkin, M.; Huerta, R.; Vulic, I. UNDIAL: Self-Distillation with Adjusted Logits for Robust Unlearning in Large Language Models. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2025), 2025, pp. 8827–8840. https://doi.org/10.18653/V1/2025.NAACL-LONG.444.
63. Wang, L.; Chen, T.; Yuan, W.; Zeng, X.; Wong, K.; Yin, H. KGA: A General Machine Unlearning Framework Based on Knowledge Gap Alignment. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (2023), 2023, pp. 13264–13276. https://doi.org/10.18653/V1/2023.ACL-LONG.740.
64. Chen, H.; Szyller, S.; Xu, W.; Himayat, N. Soft Token Attacks Cannot Reliably Audit Unlearning in Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP (2025), 2025, pp. 2183–2192. https://doi.org/10.18653/v1/2025.findings-emnlp.117.
65. Jeung, W.; Yoon, S.; No, A. SEPS: A Separability Measure for Robust Unlearning in LLMs. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (2025), 2025, pp. 5556–5587. https://doi.org/10.18653/v1/2025.emnlp-main.283.
66. Qiu, X.; Shen, W.F.; Chen, Y.; Cancedda, N.; Stenetorp, P.; Lane, N.D. PISTOL: Dataset Compilation Pipeline for Structural Unlearning of LLMs. *arXiv preprint arXiv:2406.16810* **2024**, [2406.16810].
67. Hsu, H.; Niroula, P.; He, Z.; Brugere, I.; Lecue, F.; Chen, C.F. The Unseen Threat: Residual Knowledge in Machine Unlearning under Perturbed Samples. In Proceedings of the Advances in Neural Information Processing Systems (2025), 2025.
68. Lang, Y.; Guo, K.; Huang, Y.; Zhou, Y.; Zhuang, H.; Yang, T.; Su, Y.; Zhang, X. Beyond Single-Value Metrics: Evaluating and Enhancing LLM Unlearning with Cognitive Diagnosis. In Proceedings of the Findings of the Association for Computational Linguistics (2025), 2025, pp. 21397–21420. https://doi.org/10.18653/v1/2025.findings-acl.1102.
69. Wichert, L.; Sikdar, S. Rethinking Evaluation Methods for Machine Unlearning. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP (2024), 2024, pp. 4727–4739. https://doi.org/10.18653/v1/2024.findings-emnlp.271.
70. Cohen, L.; Nemcovesky, Y.; Mendelson, A. REMIND: Input Loss Landscapes Reveal Residual Memorization in Post-Unlearning LLMs. *arXiv preprint arXiv:2511.04228* **2025**, [2511.04228].
71. Che, Z.; Casper, S.; Satheesh, A.; Gandikota, R.; Rosati, D.; Slocum, S.; McKinney, L.E.; Wu, Z.; Cai, Z.; Chughtai, B.; et al. Model Manipulation Attacks Enable More Rigorous Evaluations of LLM Unlearning. In Proceedings of the Neurips Safe Generative AI Workshop 2024, 2024.
72. Jeon, D.; Jeung, W.; Kim, T.; No, A.; Choi, J. An Information Theoretic Metric for Evaluating Unlearning Models. In Proceedings of the AAAI Conference on Artificial Intelligence (2026), 2026.
73. Hu, S.; Fu, Y.; Wu, S.; Smith, V. Jogging the Memory of Unlearned Models Through Targeted Relearning Attacks. In Proceedings of the International Conference on Machine Learning (ICML) Workshop on Foundation Models in the Wild (2024), 2024.
74. Shumailov, I.; Hayes, J.; Triantafillou, E.; Ortiz-Jiménez, G.; Papernot, N.; Jagielski, M.; Yona, I.; Howard, H.; Bagdasaryan, E. UnUnlearning: Unlearning is not sufficient for content regulation in advanced generative AI. *arXiv preprint arXiv:2407.00106* **2024**, [2407.00106].
75. Lynch, A.; Guo, P.; Ewart, A.; Casper, S.; Hadfield-Menell, D. Eight Methods to Evaluate Robust Unlearning in LLMs. *arXiv preprint arXiv:2402.16835* **2024**, [2402.16835].
76. Fan, C.; Liu, J.; Lin, L.; Jia, J.; Zhang, R.; Mei, S.; Liu, S. Simplicity Prevails: Rethinking Negative Preference Optimization for LLM Unlearning. In Proceedings of the Advances in Neural Information Processing Systems (2025), 2025.
77. Fan, C.; Jia, J.; Zhang, Y.; Ramakrishna, A.; Hong, M.; Liu, S. Towards LLM Unlearning Resilient to Relearning Attacks: A Sharpness-Aware Minimization Perspective and Beyond. In Proceedings of the International Conference on Machine Learning (2025), 2025.
78. Zhang, Z.; Wang, F.; Li, X.; Wu, Z.; Tang, X.; Liu, H.; He, Q.; Yin, W.; Wang, S. Catastrophic Failure of LLM Unlearning via Quantization. In Proceedings of the International Conference on Learning Representations (2025), 2025.
79. Cheng, J.; Amiri, H. Tool Unlearning for Tool-Augmented LLMs. In Proceedings of the International Conference on Machine Learning (2025), 2025.
80. Geng, R.; Geng, M.; Wang, S.; Wang, H.; Lin, Z.; Dong, D. Mitigating Sensitive Information Leakage in LLMs4Code through Machine Unlearning. *arXiv preprint arXiv:2502.05739* **2025**, [2502.05739].

 doi:10.20944/preprints202603.0114.v1

81. Jiang, X.; Dong, Y.; Fang, Z.; Ma, Y.; Wang, T.; Cao, R.; Li, B.; Jin, Z.; Jiao, W.; Li, Y.; et al. Large Language Model Unlearning for Source Code. In Proceedings of the AAAI Conference on Artificial Intelligence (2026), 2026.
82. Ilharco, G.; Ribeiro, M.T.; Wortsman, M.; Schmidt, L.; Hajishirzi, H.; Farhadi, A. Editing Models with Task Arithmetic. In Proceedings of the International Conference on Learning Representations (2023), 2023.
83. Wang, W.; Zhang, M.; Ye, X.; Ren, Z.; Ren, P.; Chen, Z. UIPE: Enhancing LLM Unlearning by Removing Knowledge Related to Forgetting Targets. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP (2025), 2025, pp. 25212–25227. https://doi.org/10.18653/v1/2025.findings-emnlp.1374.
84. Liu, Z.; Dou, G.; Tan, Z.; Tian, Y.; Jiang, M. Towards Safer Large Language Models Through Machine Unlearning. In Proceedings of the Findings of the Association for Computational Linguistics (2024), 2024, pp. 1817–1829. https://doi.org/10.18653/V1/2024.FINDINGS-ACL.107.
85. Ni, S.; Chen, D.; Li, C.; Hu, X.; Xu, R.; Yang, M. Forgetting Before Learning: Utilizing Parametric Arithmetic for Knowledge Updating in Large Language Models. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (2024), 2024, pp. 5716–5731. https://doi.org/10.18653/V1/2024.ACL-LONG.310.
86. Jung, D.; Seo, J.; Lee, J.; Park, C.; Lim, H. CoME: An Unlearning-Based Approach to Conflict-Free Model Editing. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistic (2025), 2025, pp. 6410–6422. https://doi.org/10.18653/v1/2025.naacl-long.325.
87. Belrose, N.; Schneider-Joseph, D.; Ravfogel, S.; Cotterell, R.; Raff, E.; Biderman, S. LEACE: Perfect linear concept erasure in closed form. In Proceedings of the Advances in Neural Information Processing Systems (2023), 2023.
88. Ren, J.; Dai, Z.; Tang, X.; Liu, H.; Zeng, J.; Li, Z.; Goutam, R.; Wang, S.; Xing, Y.; He, Q. A General Framework to Enhance Fine-Tuning-Based LLM Unlearning. In Proceedings of the Findings of the Association for Computational Linguistics (2025), 2025, pp. 18464–18476. https://doi.org/10.18653/v1/2025.findings-acl.949.
89. He, E.; Sarwar, T.; Khalil, I.; Yi, X.; Wang, K. Deep Contrastive Unlearning for Language Models. *arXiv preprint arXiv:2503.14900* **2025**, [2503.14900].
90. Wu, X.; Li, J.; Xu, M.; Dong, W.; Wu, S.; Bian, C.; Xiong, D. DEPN: Detecting and Editing Privacy Neurons in Pretrained Language Models. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (2023), 2023, pp. 2875–2886. https://doi.org/10.18653/V1/2023.EMNLP-MAIN.174.
91. Pochinkov, N.; Schoots, N. Dissecting Language Models: Machine Unlearning via Selective Pruning. *arXiv preprint arXiv:2403.01267* **2024**, [2403.01267].
92. Liu, Z.; Dou, G.; Yuan, X.; Zhang, C.; Tan, Z.; Jiang, M. Modality-Aware Neuron Pruning for Unlearning in Multimodal Large Language Models. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (2025), 2025, pp. 5913–5933. https://doi.org/10.18653/v1/2025.acl-long.295.
93. Li, Y.; Sun, C.; Weng, T. Effective Skill Unlearning Through Intervention and Abstention. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (2025), 2025, pp. 6358–6371. https://doi.org/10.18653/v1/2025.naacl-long.322.
94. Farrell, E.; Lau, Y.T.; Conmy, A. Applying Sparse Autoencoders to Unlearn Knowledge in Language Models. *arXiv preprint arXiv:2410.19278* **2024**, [2410.19278].
95. Khoriaty, M.; Shportko, A.; Mercier, G.; Wood-Doughty, Z. Don't Forget It! Conditional Sparse Autoencoder Clamping Works for Unlearning. *arXiv preprint arXiv:2503.11127* **2025**, [2503.11127].
96. Karvonen, A.; Rager, C.; Lin, J.; Tigges, C.; Bloom, J.; Chanin, D.; Lau, Y.T.; Farrell, E.; McDougall, C.; Ayonrinde, K.; et al. SAEBench: A Comprehensive Benchmark for Sparse Autoencoders in Language Model Interpretability. In Proceedings of the International Conference on Machine Learning (2025), 2025.
97. Suriyakumar, V.M.; Sekhari, A.; Wilson, A. UCD: Unlearning in LLMs via Contrastive Decoding. *arXiv preprint arXiv:2506.12097* **2025**, [2506.12097].
98. Wu, J.; Sun, H.; Cai, H.; Su, L.; Wang, S.; Yin, D.; Li, X.; Gao, M. Cross-Model Control: Improving Multiple Large Language Models in One-Time Training. In Proceedings of the Advances in Neural Information Processing Systems (2024), 2024.
99. Huang, J.Y.; Zhou, W.; Wang, F.; Morstatter, F.; Zhang, S.; Poon, H.; Chen, M. Offset Unlearning for Large Language Models. *Transactions on Machine Learning Research* **2025**, *2025*.

 doi:10.20944/preprints202603.0114.v1

100. Deng, Z.; Liu, C.Y.; Pang, Z.; He, X.; Feng, L.; Xuan, Q.; Zhu, Z.; Wei, J. GUARD: Generation-time LLM Unlearning via Adaptive Restriction and Detection. In Proceedings of the International Conference on Machine Learning (2025), 2025.
101. Pawelczyk, M.; Neel, S.; Lakkaraju, H. In-Context Unlearning: Language Models as Few-Shot Unlearners. In Proceedings of the International Conference on Machine Learning (2024), 2024.
102. Gallegos, I.O.; Aponte, R.; Rossi, R.A.; Barrow, J.; Tanjim, M.M.; Yu, T.; Deilamsalehy, H.; Zhang, R.; Kim, S.; Dernoncourt, F.; et al. Self-Debiasing Large Language Models: Zero-Shot Recognition and Reduction of Stereotypes. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (2025), 2025, pp. 873–888. https://doi.org/10.18653/V1/2025.NAACL-SHORT.74.
103. Sanyal, D.; Mandal, M. Agents are all you need for LLM unlearning. In Proceedings of the Second Conference on Language Modeling (2025), 2025.
104. Muresanu, A.I.; Thudi, A.; Zhang, M.R.; Papernot, N. Fast Exact Unlearning for In-Context Learning Data for LLMs. In Proceedings of the International Conference on Machine Learning (2025), 2025.
105. Thaker, P.; Maurya, Y.; Hu, S.; Wu, Z.S.; Smith, V. Guardrail Baselines for Unlearning in LLMs. In Proceedings of the ICLR 2024 Workshop on Secure and Trustworthy Large Language Models (2024), 2024.
106. Mitchell, E.; Lin, C.; Bosselut, A.; Manning, C.D.; Finn, C. Memory-Based Model Editing at Scale. In Proceedings of the International Conference on Machine Learning (2022), 2022, pp. 15817–15831.
107. Wang, S.; Zhu, T.; Ye, D.; Zhou, W. When Machine Unlearning Meets Retrieval-Augmented Generation (RAG): Keep Secret or Forget Knowledge? *IEEE Transactions on Dependable and Secure Computing* **2025**, pp. 1–16. https://doi.org/10.1109/TDSC.2025.3620832.
108. Kang, M.; Chen, Z.; Li, B. C-SafeGen: Certified Safe LLM Generation with Claim-Based Streaming Guardrails. In Proceedings of the Advances in Neural Information Processing Systems (2025), 2025.
109. Wu, Y.; Guo, J.; Li, D.; Zou, H.P.; Huang, W.C.; Chen, Y.; Wang, Z.; Zhang, W.; Li, Y.; Zhang, M.; et al. PSG-Agent: Personality-Aware Safety Guardrail for LLM-based Agents. *arXiv preprint arXiv:2509.23614* **2025**, [2509.23614].
110. Ji, J.; Wang, K.; Qiu, T.A.; Chen, B.; Zhou, J.; Li, C.; Lou, H.; Dai, J.; Liu, Y.; Yang, Y. Language Models Resist Alignment: Evidence From Data Compression. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (2025), 2025, pp. 23411–23432. https://doi.org/10.18653/v1/2025.acl-long.1141.
111. Hu, J.; Lian, Z.; Wen, Z.; Li, C.; Chen, G.; Wen, X.; Xiao, B.; Tan, M. Continual Knowledge Adaptation for Reinforcement Learning. In Proceedings of the Advances in Neural Information Processing Systems (2025), 2025.
112. Chen, Y.; Zhang, S. Intrinsic Preservation of Plasticity in Continual Quantum Learning. *arXiv preprint arXiv:2511.17228* **2025**, [2511.17228].
113. Xu, S.; Strohmer, T. Machine Unlearning via Information Theoretic Regularization. *arXiv preprint arXiv:2502.05684* **2025**, [2502.05684].
114. Gur-Arieh, Y.; Suslik, C.; Hong, Y.; Barez, F.; Geva, M. Precise In-Parameter Concept Erasure in Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP (2025), 2025.
115. Srivastava, A. The Fundamental Limits of LLM Unlearning: Complexity-Theoretic Barriers and Provably Optimal Protocols. In Proceedings of the ICLR Workshop on Building Trust in Language Models and Applications (2025), 2025.
116. Zou, A.; Phan, L.; Wang, J.; Duenas, D.; Lin, M.; Andriushchenko, M.; Kolter, J.Z.; Fredrikson, M.; Hendrycks, D. Improving Alignment and Robustness with Circuit Breakers. In Proceedings of the Advances in Neural Information Processing Systems (2024), 2024.
117. Foster, J.; Fogarty, K.; Schoepf, S.; Dugue, Z.; Öztireli, C.; Brintrup, A. An Information Theoretic Approach to Machine Unlearning. *Transactions on Machine Learning Research* **2025**, *2025*.
118. Di, Z.; Yu, S.; Vorobeychik, Y.; Liu, Y. Adversarial Machine Unlearning. In Proceedings of the International Conference on Learning Representations (2025), 2025.
119. Entesari, T.; Hatami, A.; Khaziev, R.; Ramakrishna, A.; Fazlyab, M. Constrained Entropic Unlearning: A Primal-Dual Framework for Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems 39 (2025), 2025.
120. Zhai, N.; Shao, P.; Zheng, B.; Yang, Y.; Shen, F.; Bai, L.; Yang, X. Maximizing Local Entropy Where It Matters: Prefix-Aware Localized LLM Unlearning. *arXiv preprint arXiv:2601.03190* **2026**, [2601.03190].

 doi:10.20944/preprints202603.0114.v1

121. Cui, Y.; Cheung, M.H. The Price of Forgetting: Incentive Mechanism Design for Machine Unlearning. *IEEE Transactions on Mobile Computing* **2025**, *24*, 11852–11864. https://doi.org/10.1109/TMC.2025.3582904.
122. Wang, Q.; Xu, R.; He, S.; Berry, R.; Zhang, M. Unlearning Incentivizes Learning Under Privacy Risk. In Proceedings of the Proceedings of the ACM Web Conference (2025), 2025, pp. 1456–1467. https://doi.org/10.1145/3696410.3714740.
123. Ding, N.; Sun, Z.; Wei, E.; Berry, R. Incentivized Federated Learning and Unlearning. *IEEE Transactions on Mobile Computing* **2025**, *24*, 8794–8810. https://doi.org/10.1109/TMC.2025.3557857.
124. Hsu, H.; Niroula, P.; He, Z.; Chen, C.F. Are We Really Unlearning? The Presence of Residual Knowledge in Machine Unlearning. In Proceedings of the International Conference on Learning Representations (ICLR) Workshop on I Can't Believe It's Not Better: Challenges in Applied Deep Learning (2025), 2025.
125. Jenko, S.; Mündler, N.; He, J.; Vero, M.; Vechev, M.T. Black-Box Adversarial Attacks on LLM-Based Code Completion. In Proceedings of the International Conference on Machine Learning (2025), 2025.
126. Zhang, B.; Chen, Z.; Shen, C.; Li, J. Verification of Machine Unlearning is Fragile. In Proceedings of the International Conference on Machine Learning (2024), 2024.
127. Xu, H.; Zhu, T.; Zhang, L.; Zhou, W. Really Unlearned? Verifying Machine Unlearning via Influential Sample Pairs. *IEEE Transactions on Dependable and Secure Computing* **2026**, *23*, 1671–1686. https://doi.org/10.1109/TDSC.2025.3620308.
128. Wang, N.; Wu, N.; Hui, X.; Wang, J.; Yuan, X. zkUnlearner: A Zero-Knowledge Framework for Verifiable Unlearning with Multi-Granularity and Forgery-Resistance. *arXiv preprint arXiv:2509.07290* **2025**, [2509.07290].
129. Maheri, M.M.; Cotterill, S.; Davidson, A.; Haddadi, H. ZK-APEX: Zero-Knowledge Approximate Personalized Unlearning with Executable Proofs. *arXiv preprint arXiv:2512.09953* **2025**, [2512.09953].
130. Eisenhofer, T.; Riepel, D.; Chandrasekaran, V.; Ghosh, E.; Ohrimenko, O.; Papernot, N. Verifiable and Provably Secure Machine Unlearning. In Proceedings of the IEEE Conference on Secure and Trustworthy Machine Learning (2025), 2025, pp. 479–496. https://doi.org/10.1109/SaTML64287.2025.00033.
131. Gao, X.; Ma, X.; Wang, J.; Sun, Y.; Li, B.; Ji, S.; Cheng, P.; Chen, J. VeriFi: Towards Verifiable Federated Unlearning. *IEEE Transactions on Dependable and Secure Computing* **2024**, *21*, 5720–5736. https://doi.org/10.1109/TDSC.2024.3382321.
132. Nguyen, T.L.; de Oliveira, M.T.; Braeken, A.; Ding, A.Y.; Pham, Q.V. Towards Verifiable Federated Unlearning: Framework, Challenges, and The Road Ahead. *IEEE Internet Computing* **2026**. https://doi.org/10.1109/MIC.2026.3656638.