Representation-Aware Unlearning via Activation Signatures: From Suppression to Knowledge-Signature Erasure

by

Md Rezaur Rahman Bhuiyan
22301294
Syed Naveed Mahmood
22301257
Jareen Tasneem Khondaker
22301308
Md Sameer Sakib
24241344
Tasfia Zaman
22301779

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
BRAC University
January 2026.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md Rezaur Rahman Bhuiyan

22301294

Syed Naveed Mahmood

22301257

Jareen Tasneem Khondaker

22301308

Md Sameer Sakib

24241344

Tasfia Zaman

22301779

Approval

The thesis titled "Representation-Aware Unlearning via Activation Signatures: From Suppression to Knowledge-Signature Erasure" submitted by

1. Md Rezaur Rahman Bhuiyan (22301294)
2. Syed Naveed Mahmood (22301257)
3. Jareen Tasneem Khondaker (22301308)
4. Md Sameer Sakib (24241344)
5. Tasfia Zaman (22301779)

of **Spring, 2025** has been accepted as satisfactory in partial fulfillment of the requirement for the Degree of B.Sc. in Computer Science in **Fall, 2025**.

Examining Committee:

Supervisor:
(Member)

Dr. Farig Yousuf Sadeque

Associate Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

K.M. Shadman Wadith

Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Associate Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Ethics Statement

Knowledge Immunization Framework (KIF) has significant implications on dual-use and data privacy, in line with the ACL Ethics Policy. On one hand, it enables compliance with regulations (e.g., GDPR) [1]–[3] and the removal of sensitive materials, and on the other hand, the erasure of sensitive content by KIF can be used as a tool of weaponised censorship, safety-guardrail removal, or political promotion. In addition, we note that "erasure" on high-dimensional neural networks is always relative; even with massive amounts of trace reduction, the existing paradigms do not mathematically guarantee it to be irrecoverable against extreme adversarial fine-tuning, or jailbreaks in the future [4]–[7]. Individuals using this framework should not have a false sense of security since erasing or exposure of data on non-deterministic systems do not carry the certainty as erasing data on traditional databases [4].

In addition to privacy, KIF implementation poses risks to the integrity of models and fairness in society. Latent pruning, as observed, is a possible result of targeted erasure, even with better calibration, that can produce so-called semantic holes whose erasure can lead to the unintentional destruction of more general contextual knowledge. These changes might bring in random performance declines or bias in demographics. But in terms of sustainability, KIF has a major moral benefit: with the help of Parameter-Efficient Fine-Tuning (PEFT) and the self-healing loop, it can be used as a computationally lightweight alternative to full-model retraining, which cuts carbon footprint and environmental cost by far and significantly reduces carbon footprint and environmental cost [8], [9] of maintaining LLMs, and thus, serves as a tool towards sustainable AI development worldwide.

Abstract

The rapid development of large language models (LLMs) has outpaced regulations (e.g. GDPR) and ethical frameworks, raising concerns about privacy compliance, bias, misuse, misinformation, and legal adaptability. This makes the ability to selectively erase knowledge from LLMs critical. Despite significant developments, current unlearning methods are not able to segregate behavioral suppression and true knowledge removal, allowing latent capabilities to persist beneath surface-level refusals. In this paper, we address this challenge by introducing **Knowledge Immunization Framework (KIF)**, a representation-aware architecture that differentiates between true erasure and obfuscation by operating on the internal activation signatures of the model, as opposed to surface-level outputs. KIF achieves near-oracle erasure (FQ $\approx$ 0.99 vs. 1.00) and utility preservation (MU = 0.62), effectively breaking the stability-erasure tradeoff that has constrained all prior work. Our observation shows that standard models exhibit scale-independent true erasure (<3% utility drift), while reasoning-prior models reveal fundamental architectural divergence. Our comprehensive dual-metric evaluation protocol, combining surface-level leakage with latent trace persistence, operationalizes the obfuscation - erasure distinction and enables the first systematic diagnosis of mechanism-level forgetting behavior across model families and scales.

Keywords: LLM Unlearning, Knowledge Entanglement, Exact Unlearning, Approximate Unlearning, Privacy, Computational Overhead, LoRA, KIF

Acknowledgement

We start by thanking **Allah S.W.T.**, the All-Knowing, for guiding us throughout this journey.

We would like to thank **Dr. Farig Yousuf Sadeque** for being our supervisor and steering this research in the right direction. We also appreciate **K. M. Shadman Wadith** for his support as our co-supervisor.

Lastly, we owe it all to our parents. Their belief in us pushed us to where we are today.

Table of Contents

List of Figures

List of Tables

Nomenclature

The following list describes several symbols & abbreviations that will be used within the body of the document.

Acronyms

DPO Direct Preference Optimization

EL10 Extraction Likelihood Ratio (Metric for latent traces)

KIF Knowledge Immunization Framework

LLM Large Language Model

LoRA Low-Rank Adaptation

MLP Multi-Layer Perceptron

PEFT Parameter-Efficient Fine-Tuning

SMR Subject Mention Rate (Metric for surface leakage)

TID Teacher Interaction Dataset

TOFU Task of Fictitious Unlearning (Benchmark)

Greek Symbols

α Suppression scalar controlling the strength of the capsule

λ Hyperparameter coefficient weighting loss components

Φ Parameters of the Global LoRA Adapter

$\phi(\cdot)$ Nonlinear activation function within the MLP (e.g., SiLU)

$\sigma(\cdot)$ Sigmoid activation function used in gating

τ Trigger threshold for the capsule gating mechanism

Latin Symbols

$\mathbf{v}$ Signature Vector: The direction representing subject-specific knowledge

$\mathcal{D}_{\text{pref}}$ Derived Preference Dataset used for LoRA training

$\mathcal{L}$ Composite optimization objective (Total Loss)

$\mathcal{L}_{\text{DPO}}$ Direct Preference Optimization loss term

$\mathcal{L}_{\text{NTUL}}$ Name-Token Unlikelihood loss term

$h^{(\ell)}$ Hidden state activation at layer ℓ

M The Large Language Model (stack of transformer blocks)

p Raw projection score of the hidden state onto the signature vector

$W_{\text{gate}}, W_{\text{up}}$ Projection weights within the MLP block

z Standardized Z-score of the projection

Chapter 1

Introduction

1.1 The Background

Large Language Models represent by far one of the most significant technological advancements of modern history [10]. Exhibiting advanced textual processing capabilities and coding skills, Large Language Models currently supported by **Retrieval-Augmented Generation** [11] allow for the retrieval of updated information from the web or user inputs. Coupled with **Continual Learning** and **Reinforcement Learning**, these architectures allow the LLM to train itself over time on updated data [10].

The foundation of Language Models, however, started when Noam Chomsky in 1956 [12], introduced the three hierarchical language models, corresponding to theories of Automata, to describe the structure of human language via specific rules. This bridged the world of linguistics and mathematics. The following decades—1960s and 1970s—began to observe the emergence of NLP-based applications, albeit predominantly relying on rule-based approaches like **ELIZA** (1966) [13] - a rule-based chatbot with preset questions, and **SHRDLU** (1970) [14] - which is able to respond to natural language in a controlled environment. However, the limitations of such rule-based approaches became apparent when faced with the ever-changing complexity, ambiguity, and variability of natural language. The sheer volume and nuance of linguistic rules are too vast to be manually encoded comprehensively.

In order to address the issue, language models started to take a data-driven approach via Statistical/Probabilistic Language Models [11]. The 1980s to early 2000s were dominated by the n-gram approach where the probability for the placement of a word depends on the preceding $n-1$ words [11]. This start of using mathematical approaches for Language Models laid the foundation for it to be integrated with Machine Learning Techniques [11]. Large text corpora [11] like the Penn Treebank, and the concurrent explosion of the internet [3] furnished an unprecedented volume of text, allowing for a broader application of Machine Learning techniques [3]. It prompted the development of **word embeddings** in the early 2010s, like **word2vec** by Thomas Mikolov [11], [15].

The representation of words as dense vectors allowed for a rich mathematical representation of semantics, which, although struggled with ambiguity, was further

developed to be used in conjunction with RNNs like LSTM and GRU, before significant progress of Language Models slowed down [16]. The introduction of the Transformer model in 2017 acted as a significant catalyst for the development of Large Language Models [10]. Its capabilities to efficiently apply weights to important sections of a sequence 'self-attention' over long ranges, along with its scalability birthed Large Language Models like Bert, and GPT [11].

Using said models, natural language processing entered its new phase of Generative Artificial Intelligence—imitating the knowledge and reasonings of humans [10], with trained information readily available in its Conceptual spaces. ChatGPT, for example, the first exposure to LLMs for the masses across the globe single handedly changed the global perception of AI by becoming a software with the fastest usage growth in history [3], specifically GenAI, leaving people in awe.

1.2 Motivation

The basic right of privacy, a right recognised by the International law, including the Universal Declaration of Human Rights (Article 12) stating, "No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence", congruent with its digital counterpart "The Right to be forgotten", gaining prominence through the General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA), and more, empowers individuals to request the enforcement of their information to be removed from any publicly accessible platform [3].

The training of Large Language Models requires a vast amount of data for training. They have often been trained on publicly accessible data, which weren't always meant to be used without consent [3]. Companies, even if they want to comply with removal requests, face major complications due to a Large Language Model's data storage mechanism. An LLM's Conceptual Space (the information storage) is a set of vectors, weights and/or parameters, complicating the deletion of specific information. Furthermore, LLMs might inadvertently absorb biases and misinformation [3] present in their training data. With the ever evolving knowledge base of humans, the need for the ability to selectively 'forget' specific information is vital.
Crucially, a significant gap exists in how current methods address this need. Many existing techniques result in **obfuscation** rather than true **erasure** [5]. While the model may refuse to answer a query, the latent capability often remains intact within the parameters, leaving the model vulnerable to jailbreaking or adversarial probing. This distinction between behavioral suppression and genuine removal serves as the primary motivation for this work.

1.3 Problem Statement

Classical databases have a straightforward method of removing/deleting information. Large Language Models on the other hand are built as billions upon trillions of parameters [10]. These parameters are what stores the data and dictates the model's behavior. The problem lies in the fact that LLMs being black boxes, the

workings of the parameters are yet unfound. On top of which, the data is distributed and entangled [17] across the parameters making it very complicated to understand how they affect parameter behavior and subsequently even harder to make the model 'forget'.

Current methodologies, such as reverse-optimization (e.g., gradient ascent [18]) or preference-based objectives (e.g., NPO [19]), face a critical **stability-erasure tradeoff**. Strong forgetting pressure often leads to a degradation of the model's general utility, while gentler approaches fail to remove the underlying knowledge signature. Furthermore, there is no perfect standardized evaluation method to verify that the data has actually been forgotten at the representation level, rather than merely suppressed at the surface level [20]. All the stated factors combined leaves no full-proof option than to retrain the whole model from scratch in case there is a requirement to unlearn a specific knowledge.

1.4 Objective

The very definition of unlearning is an evolving construct. Initial ideas being limited to retraining the whole model from scratch, have seen significant progress in the hands of researchers. For Large Language Models, where retraining a model from scratch might prove to be extremely resource consuming, techniques like Gradient-Ascension [21], DeepCut [22], and representation steering (RMU [23]) are being consistently developed to allow LLMs to Unlearn only the part the user wants it to.

However, none of the techniques are without caveats; they often struggle to distinguish between erasure and obfuscation. This paper aims to bridge this gap by introducing the **Knowledge Immunization Framework (KIF)**. Unlike standard approaches, KIF targets the internal activation signatures of the model to distinguish genuine erasure from obfuscation. The objective is to produce a novel Targeted Unlearning technique that uses dynamic suppression and parameter-efficient updates (via LoRA [24]) to selectively 'forget' specific data while reaching as close as possible to the conditions of Exact Unlearning while minimizing drawbacks.

1.5 Methodology

The process begins with a systematic evaluation of every unlearning framework to implement a quantitative and experimental approach to understand existing technologies. We propose a lightweight framework that leverages internal representation signals, extracted using a custom real-world entity-based prompt dataset.
This paper aims to contribute in mitigating the drawbacks of LLM Unlearning by pioneering a new architecture, **KIF**, which leans towards leveraging the Knowledge Editing framework. This architecture continually distills suppression into parameter weights using LoRA to ensure stability-aware updates. This study shall map the following for every unlearning framework:

- **Unlearning efficacy** - Does the model really unlearn well (True Erasure vs. Obfuscation)?

- **Model utility** - How much will the model degrade post unlearning?
- **Vulnerabilities** - What might render the unlearning moot per framework? What are its solutions?
- **Computational overhead** - How resource intensive are the techniques per framework?

The subsequent research will focus on innovating upon this foundation, specifically evaluating the framework across open-source chat and reasoning model families (e.g., Llama, Mistral, Qwen, DeepSeek) to identify regimes where unlearning remains stable versus regimes that revert to obfuscation-like behavior.

1.6 Scopes and Challenges

Challenges in this research can be categorised into two parts. Firstly: *Hardware*. Limitations in access to hardware may cause limitations in assessing certain techniques and may coc nsume a significant amount of time. Secondly, a conceptual issue where there is no evaluation technique that can objectively determine if an unlearning is absolutely successful. Furthermore, being a relatively new domain—LLMU, most of the literature the authors of this paper came across are in a pre-print format. Not formally published.

About the scopes however, this paper aims to follow through with three objectives:

1. To evaluate and examine advanced LLMU techniques on specific Large Language Models and compare their efficacy (how well they forget), utility (performance degradation of LLM post Unlearning) and the associated computational costs.
2. Compare evaluation techniques and frameworks in an attempt to standardise evaluation protocol for LLM Unlearning that separates behavioral suppression from durable erasure by jointly measuring leakage and internal signature persistence.
3. Development of a novel technique, the **Knowledge Immunization Framework (KIF)**, that can make LLMs selectively unlearn specific data. The technique aims to minimize the gap between exact unlearning and approximate unlearning, effectively breaking the stability-erasure tradeoff.

1.7 Workflow

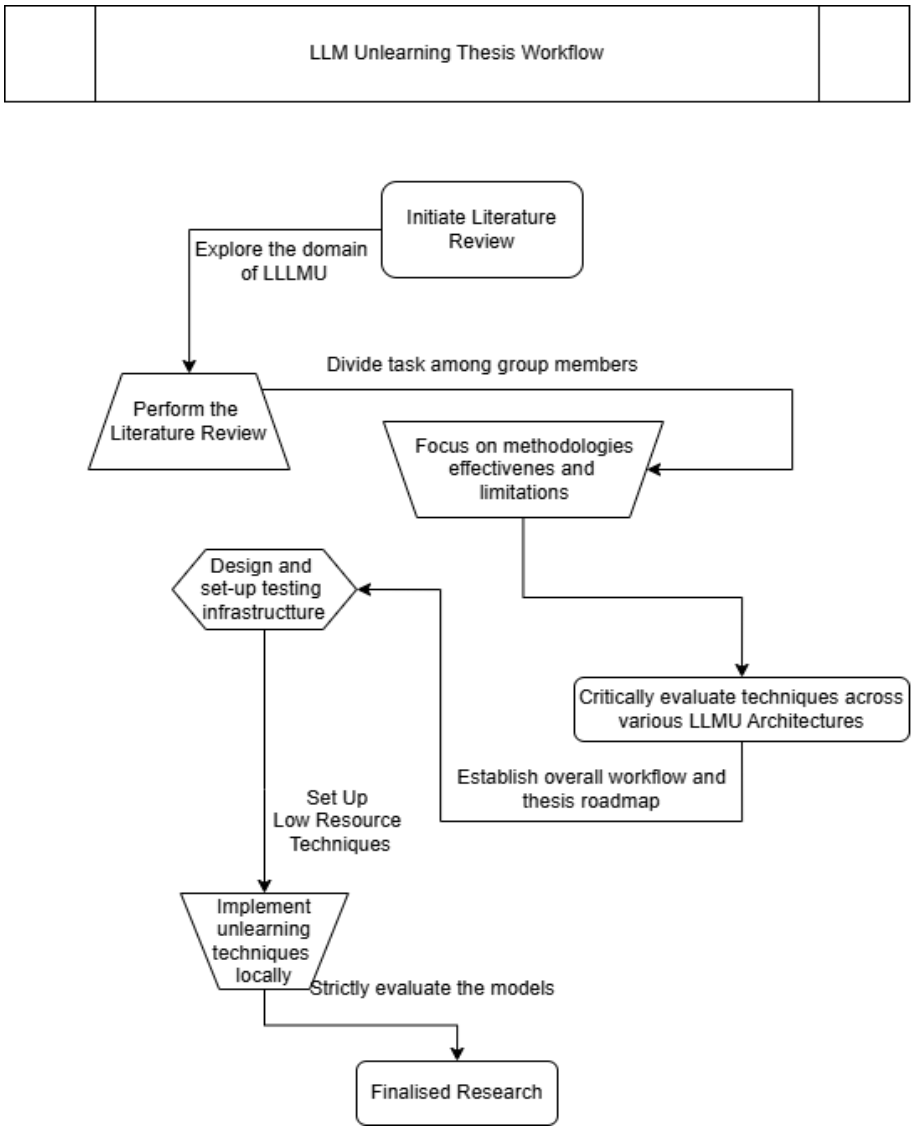

Figure 1.1: Thesis Workflow

Chapter 2

Literature Review

2.1 Reasons for LLM Unlearning

The drive to develop effective LLM unlearning techniques stems from a confluence of pressing legal, ethical, security, environmental, and operational demands. Each of these factors underscores the necessity for mechanisms that allow for the controlled and efficient modification of knowledge within already trained LLMs.

2.1.1 Legal Compliance

Prominent data protection regulations such as the European Union's General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA), have established the "right to be forgotten" (RTBF) or "right to erasure." These legal frameworks grant individuals the right to request the deletion of their personal data from systems that process it, including AI models [25]. For LLMs, this implies not only removing the raw data if it was directly ingested but also mitigating any residual influence that data may have had on the model's learned parameters, knowledge, and subsequent behavior.

Moreover, the conventional approach to ensure such removal is retraining the entire model from scratch on a dataset excluding the specified user's data. But this is often impractical for both small and large scale LLMs due to the immense computational costs, extended time requirements, and associated service disruptions as well as repeated retraining when an individual person asks for data removal [25]. So, for these reasons LLM unlearning techniques emerge as a potentially viable solution to meet these legal obligations more efficiently.

But, achieving genuine data removal in large language models that truly aligns with both the intent and requirements of the Right to Be Forgotten is still a major hurdle. Some analyses suggest that the only method to guarantee complete removal of an individual's data influence is to retrain the model entirely from scratch, a solution deemed infeasible for many practical purposes [25]. The design of generative AI models makes the right to be forgotten practically unworkable, since such models may retain generalized patterns from deleted data, preventing full erasure [26].

2.1.2 Ethical Imperative

Large language models (LLMs) trained on extensive internet text datasets may unintentionally learn and reinforce societal biases concerning characteristics like gender, race, religion, and socioeconomic background. These biases can manifest in the models' outputs which lead to unfair, discriminatory, or offensive content and thereby raising significant ethical concerns.

In a systematic analysis of GPT-3, it was found that the model associated the term "Muslim" with "terrorist" in 23% of test cases, revealing a consistent and significant anti-Muslim bias. Conversely, the term "Jewish" was linked to "money" in only 5% of instances [27]. Even after introducing strongly positive adjectives, violent completions involving Muslims remained at 20%, significantly higher than the baseline for other religious groups. Additionally, significant ethical issues arise when LLMs are trained on personal information without explicit consent or retain learned patterns that result in harmful, toxic, hateful, or misleading content, highlighting the critical need for post-hoc unlearning methods [28].

As for ethics in LLM Unlearning, it requires value judgments and risks creating narrow perspectives. Overly sanitized models may struggle with complex topics. Transparency and public discourse are crucial for responsible unlearning that considers diverse viewpoints [28].

2.1.3 Security

LLMs face security threats like data poisoning and backdoor attacks. An attacker can intentionally inject "poisoned" data into the training set to manipulate the model's behavior. This can lead to incorrect outputs, biased responses, or system failures. For example, a model could be poisoned to consistently provide positive sentiment for a specific product or to generate harmful content when given certain prompts. Unlearning can be used to remove this poisoned data and its effects from the model [29].

Identifying and removing poisoned data in Large Language Models is difficult because of their generalization capabilities and the shared representation space with benign data. Simple data removal can cause collateral damage to model performance, and some backdoors require more complex defense mechanisms [29]. For this reason effective security-oriented unlearning needs both accurate detection methods and robust mitigation protocols to remove data influence without harming overall model integrity. Future solutions may involve continual learning, differential privacy, or certifiable robustness to enhance security.

2.1.4 Environmental Cost

Retraining a large language model from scratch to remove even a small amount of data carries a substantial environmental burden. For instance, retraining a 13B parameter model requires 230 MWh of energy and 892,000 liters of water, emitting

over 100 metric tons of CO_2 in the process. LLM unlearning offers a far more sustainable alternative by avoiding these high costs [29].

While approximate unlearning methods have their own computational footprint, it is orders of magnitude smaller than the full retraining costs. Faced with these substantial environmental costs, the development and adoption of LLM unlearning techniques offer a far more sustainable pathway. Instead of the resource-intensive process of retraining it works by targeting specific problematic information within the model's parameters, these methods significantly reduce computational resources, energy consumption, water usage, and carbon emissions compared to full retraining. This leads to a more ecologically responsible approach to AI development and deployment.

2.1.5 Operational Scalability and Costs

In practice, retraining Large Language Models (LLMs) assumes recommending more time and money, and is economically unsustainable even at smaller model sizes. The traditional retraining of LLMs requires huge amounts of computing power, time-consuming training, and considerable downtime, and is impractical when it comes to systems that need frequent updates, mistakes corrected, or data deleted due to compliance.

According to recent reports, training a model such as GPT-3 ($175 billion parameters) [30], required about $4.6 million dollars in compute resources alone, while Google's PaLM (540 billion parameters) [31] is estimated to have cost $9–12 million to train. These estimates do not cover operating costs such as infrastructure maintenance, data curation and fine tuning work flows. It is not economically justified or scalable to have to retrain such models each time there is a slight update of the content or regulatory requirements. Moreover the delays that can be caused by this kind of model re-initialization, training and redeployment may turn into a bottleneck in the production set up based on real-time decision-making.

A much more viable substitute is unlearning. Instead of re-training the complete model, unlearning methods aimed at specific data deletion or forgetting the concepts without interference with the rest of the knowledge. As demonstrated in recent work, such as Chen and Yang's SISA-inspired approach, effective unlearning can reduce computational cost by up to 80% compared to full retraining [32], making unlearning a viable pathway for cost-effective compliance with GDPR, CCPA, or internal auditing requirements in dynamic applications.

2.2 Foundational Concepts in LLM Unlearning

Since large language models (LLMs) are increasingly being applied across various applications, it has become of great significance to be able to modify or delete certain knowledge in these models. In the framework of LLMs, unlearning refers to the procedure of re-training a trained model to remove the effect of a particular set of data, called the "forget set". The goal is to adjust the settings of a large language

model to delete or interfere with a specific knowledge or behavior of interest [33]. The targeted goal is that the updated model should no longer be represented or maintain any information associated with a specific forget set (D_f), and at the same time retain the knowledge maintained in the retain set (D_r).

Formally, given an original model (M_{orig}) trained on a dataset (D_{train}), and a subset ($D_{\text{forget}} \subset D_{\text{train}}$) that necessitates removal (e.g., due to privacy concerns, bias, or inaccuracies), the aim is to produce a model ($M_{\text{unlearned}}$) such that:

- $M_{\text{unlearned}}$ exhibits behavior consistent with a model trained from scratch exclusively on the retained set ($D_{\text{retain}} = D_{\text{train}} \setminus D_{\text{forget}}$), and
- the influence of D_{forget} on $M_{\text{unlearned}}$ is effectively nullified.

This process involves changing the parameters of M_{orig} in such a way that the knowledge that is being encoded by D_{forget} is completely removed but the performance of the model is not affected on the rest of the data. Achieving this efficiently for large-scale models is a core challenge in the field.

2.2.1 Unlearning vs. Retraining

The idea of unlearning is based on the removal of a certain information of an already-trained model without necessarily beginning all over and re-training the model right back to the beginning. This makes it very efficient both in terms of time and computing power. It is particularly helpful with large models such as LLMs, in which a full retraining is frequently just impossible due to volume of data and size of models.

In contrast, retraining implies discarding the afflicted model and thereafter educating a new one (or refining the current one) on an updated dataset that either overlooks or substitutes the data you need to eliminate. Although retraining is certainly the surest method of removing the information, it demands access to all the original information and is computationally expensive.

Unlearning methods attempt to approach the impact of retraining by searching and removing the impact of specific data points or knowledge. It may include such approaches as SISA training [34], influence functions, or gradient-based knowledge erasure [25]. Recent research has expanded these ideas to LLMs, where precisely editing or unlearning facts or behaviors is crucial without messing up the model's general language abilities. Therefore, retraining has greater theoretical assurances, but unlearning is a more feasible choice in those case where the requirement is immediate, localized additions to existing models.

2.2.2 Exact vs. Approximate Unlearning

Exact Unlearning

Exact unlearning is the ideal scenario for forgetting. The goal here is to create a model that's absolutely identical to one that was never trained on the information you want it to forget [35]. It's like turning back time so the model truly never encountered that "forget set." This completely wipes out the influence of that data.

One of the most well-known ways to achieve this is through a framework called SISA (Sharded, Isolated, Sliced, and Aggregated). Imagine dividing your entire training dataset into smaller chunks, or “shards,” and then training separate mini-models for each shard, which are later combined to make the final unlearned model [4]. Each subset trains a sub-model, and these sub-models are later combined to form the final model [25].

Although this method is correct, and can be significantly more computationally efficient than the regular ML models, it is not always feasible with large LLMs since even re-training a single component can demand a lot of computational power [4]. You also require having all those parted datasets and possibly in-between training processes at hand and easily accessed, that is, large storage needs. Cao and Yang [25] suggested transforming traditional machine learning algorithms into a "summation form," allowing for quick updates by just tweaking a few sums instead of retraining the whole model.In other methods, the model is divided into a core component and a removable component related to the forget set whereby targeted unlearning can be performed. Nevertheless, they are even optimized and still need much storage of intermediate states.

Furthermore, the precise unlearning is most effective when you just have to forget few definite pieces of information. In case you are forced to discard a large or heterogeneous part of the training data, particularly when this is dispersed across a large number of very different sub-models, then you may find yourself obliged to re-train a large number of the sub-models, or even all the sub-models. When this happens, the computational advantage disappears and you would still require complete access to all of the data and internal parameters and it is not as scalable to large-scale unlearning problems.

Approximate Unlearning

Approximate unlearning is a more practical and widely explored alternative to exact unlearning. It aims at producing a model which will behave nearly identically to one which has been retrained, but with only a small fraction of the computing expense of a complete retraining. This is the direction in which the bulk of the studies in the field of LLM unlearning is moving.

Approximate unlearning directly acts on the parameters (the internal settings) of a previously-trained model without necessarily retraining the entire model (or even some part of it). It modifies these weights based only on the data you want the model to forget [35]. The aim is to generate a new model, which we can refer to as M_{approx}, that behaves in a similar manner as a model trained using the retained data.

One of the major advantages of this approach is that it is efficient. With updates made only with the use of the only the "forget data", approximate unlearning drastically reduces storage and computation requirements. Then, rather than attempting to be an extreme duplicate of a retrained model, approximate unlearning makes a trade-off. It attempts to eliminate the effect of the "forget set" (D_{forget}) in the most optimal way possible and maintain the overall behaviour of the model. Typical methods include gradient based updates, utilization of influence functions approx-

imations and regularization techniques that are aimed at being quick and effective at suppressing the influence of the forgotten data. Although it is possible that some small remnants of the forgotten data still remain, still, with the assistance of a well-designed approximate unlearning techniques, it is possible to obtain a high degree of forgetting with minimal effect on the performance of the model.

In contrast with exact unlearning, which primarily aims to make retraining less and less challenging, approximate unlearning directly modifies the current model with the help of the data alone that requires erasing. It is also very efficient in computation and storage, which makes it a much more appropriate fit to deploying and operating the current enormous LLMs in practice, although with some model utility caveats.

2.2.3 The Spectrum of Unlearning Tasks

Unlearning isn't a one-size-fits-all solution; it comes in different forms, depending on how specific or broad the information to be removed is:

Instance and Fact Removal

As the most intricate type of unlearning, this focuses on eliminating individual pieces of training data. The data can be specific sentence, a PII, a copyrighted information, to just about any specific information from the model's learned parameters. This process upholds privacy compliance policies such as the "right to be forgotten," ensuring sensitive information can't be reproduced by the model.

TOFU (Task of Fictitious Unlearning) [36], a key bechmark for the evaluation of this task, uses fictitious datasets and corresponding prompts to challenge models to "forget" the synthetic data all the while keeping unrelated capabilities intact. TOFU is a dataset of 200 fictional author profiles (each made up of 20 question-answer pairs). A subset of these profiles is designated as the "forget set," on which methodologies are then tested on their ability to remove knowledge of these targets without hurting the model's performance on the remaining data [36].

Skill Unlearning

This denotes the removal of specific capability instead of knowledge. It is a rather advanced and complex aspect of unlearning as it is deeply embedded within a model's parameters and activation patterns. The 'Capability' may include anything ranging from code generation, mathematical reasoning, or language translation.

This means to carry out skill unlearning the underlying internal mechanisms, such as specific neuron activations or latent representations that support these capabilities have to be targeted, all while making sure the model general language proficiency and other unrelated abilities stay intact.

For instance, training-free methods like "Neuron Adjust" and "Key Space Detection" [37] effectively suppress skills such as coding or math problem-solving in large language models. Their approach has shown over an 80% reduction in targeted

skill performance while causing minimal negative impact on the model's general utility [37].

Concept Unlearning

Representing the most elaborate form of unlearning, 'Concept Unlearning' targets entire subjects of knowledge from the model parameters (e.g. biomedical, legal, or historical content). It helps removal of harmful information or bias on a large scale - particularly in domain-specific language models. Unlike fact or skill unlearning, targeting isolated knowledge or capabilities, concept unlearning deals with abstracts and knowledge interconnected with other domains on a large scale. Successfully performing this task requires identifying and suppressing clusters of neurons or subnetworks encoding domain-specific content, all while preserving the model utility.

Example includes a work by Yamashita et al. (2024) [37], where concepts are represented using knowledge graphs and uses techniques like gradient ascent on token sequences to erase specific domains from models, using sources such as Wikipedia.

Multimodal Unlearning

This is an emerging and increasingly crucial area within machine unlearning, driven by the rapid adoption of Multimodal Large Language Models (MLLMs) that can process and generate content across various input types such as text, images, audio, and video. The challenge here is to remove information specific to one modality—for example, erasing visual content like a celebrity's face or a private image—without harming the model's abilities in other modalities like text comprehension or speech recognition.

This task is particularly complex due to the interwoven representations across different modalities within MLLMs. The goal is to selectively remove knowledge embedded in one modality while maintaining performance and coherence across others, which demands very fine-grained control over the model's internal architecture.
Several recent works have started to address this challenge:

- **MMUnlearner**: Uses geometry-constrained gradient updates to selectively erase visual patterns while preserving text-based knowledge [38].
- **MultiDelete**: Focuses on ensuring modality decoupling, preserving both unimodal and multimodal knowledge after unlearning [39].
- **Modality-Aware Neuron Pruning (MANU)**: Identifies and prunes neurons specific to a modality that are crucial for the targeted information to be forgotten [40].
- **SafeEraser benchmark and methods**: Quantifies how much "over-forgetting" occurs in Visual Question Answering (VQA) tasks using a new metric (SARR) and introduces tailored loss functions for selective visual forgetting [41].

2.3 Core Challenges in LLM Unlearning

The 'gold standard' [42] of Exact Unlearning being retraining from scratch, the computation overhead leaves prompts towards techniques to minimize that. But, it still requires exceptional computational resources [43]. This leads us to *approximation unlearning* that attempts at mimicking exact unlearning parameters as closely as possible [42]. However, research into LLM internals reveals that unlearning often alters only surface-level weights, such as those in attention mechanisms, while deeper MLP layers—where factual knowledge is densely encoded—remain largely intact. Basically stating that **unlearning mainly tweaks internal representations without fully removing them** [44].

2.3.1 The Distribution of 'Knowledge'

Large Language Model is made of billions of parameters, where the Knowledge $\rightarrow K$ (a set of vectors) is distributed throughout the parameters via complex non-linear functions $\rightarrow f$, resulting in the Knowledge$(k) = f(\theta_1, \theta_2, ..., \theta_n)$ spreading out across the parameters. This results in a *Conceptual Space* of embeddings, where knowledge k_i has an impact on $\{k_j\}_{j \neq i}$ [22], and such mesh arising through billions of parameters leave researchers unable to trace through them [43].

Beyond token embeddings, often, higher-level concepts are also entangled, leaving a "fuzzy boundary" between retention set (D_{retain}) and the forget set (D_{forget}) [17]. In layman terms, if a specific information is set to be unlearned, it can result in a cascade where knowledge with some relations also gets forgotten [42]. Or, it isn't forgotten at all as its weights are represented in other knowledge [42]. It's synonymous to how attempting to forget the poem 'Birodhi' results in the LLM forgetting every literature by Kazi Nazrul Islam, or 'Birodhi' isn't forgotten at all because Rabindranath Tagore endorsed Kazi Nazrul Islam [45].

2.3.2 The Forgetting-Retention Trade-off

Formally, one might write an unlearning objective as minimizing [43]:

$$\mathcal{L}_{\text{unlearn}}(\theta) = \lambda_{\text{forget}} \mathcal{L}(\theta; D_{\text{forget}}) + \lambda_{\text{retain}} \mathcal{L}(\theta; D_{\text{retain}}) \tag{2.1}$$

where θ are model parameters, $\mathcal{L}(\cdot; D)$ is the loss on data D, and $\lambda_{\text{forget}}, \lambda_{\text{retain}}$ trade off forgetting vs. retention.

As no method achieves perfect separation of Knowledge k_i from k_j, **utility degradation is inevitable** [42]. In practice, the weights of λ_{retain} and λ_{forget} must be tuned [43] so that optimization converges. The two follow through with an inverse relationship—if the retention is high, while the forget is low the attempt at forgetting is deemed redundant. If the opposite happens, the model might encounter a *catastrophic forgetting* or might even collapse. For example, Zhang et al. in [46] note that "existing unlearning objectives all fail to retain overall performance" on

non-target data. Balancing this is one of the key difficulties for Unlearning in Large Language Models.

2.3.3 Privacy and Security Vulnerabilities

Natural Language, something developed over the millennia, allows for the expression of one knowledge via multiple representations. Most of the current LLM Unlearning approaches being "context and task-dependent" [43], it risks the unlearning technique's ability to generalise for information of different domains. Even in the case of the very same information, unlearning cannot be confirmed if the LLM is queried using a language it was not made to unlearn using [43].

The entangled (2.3.1) nature of knowledge may allow a leeway for users to generate sensitive unlearned information using mere 'paraphrasing' [43] or even a 'multi-hop question' [43].

$$\forall \text{ paraphrase } p(x) : P(f_\theta(p(x)) = y) > \epsilon \text{ even after unlearning } x \tag{2.2}$$

More advanced jailbreaking techniques, adversarial prompting, or even adding basic filler text might make the unlearned knowledge resurface [15], indicating such Unlearning technique might be more focused on *obfuscation*: suppressing the information rather than truly erasing the underlying embeddings [47], [48]'s "knowledge patching" experiments demonstrate that restoring altered MLP coefficients or attention weights can effectively recover forgotten knowledge.

Mathematical Vulnerability: The vulnerability can be expressed as:

$$\exists \text{ attack } A : P(A(\theta_{\text{unlearned}}) \to x_{\text{forget}}) > \delta \tag{2.3}$$

This means that for any unlearned model ($\theta_{\text{unlearned}}$), there exists an attack method (A) such that the probability (P) of successfully recovering or inferring the forgotten information (x_{forget}) is greater than some defined threshold (δ). This formalizes the *lack of permanent unlearning.*

This allows changes in model weights via fine tuning, or training on unrelated data to recover the unlearned knowledge. Retraining on just the D_{retain} can lead to the forget-set accuracy to rise back up to $50\% \to 100\%$ [47]. Additionally, quantization recovery techniques—which reverse low-bit precision transformations used during unlearning—have been shown to restore $21\% \to 83\%$ of the supposedly unlearned knowledge, further highlighting that **internal representations persist** [49]. Together, it suggests current techniques "fail at truly unlearning" [50].

2.3.4 Obfuscation vs. True Erasure

A critical and recently highlighted challenge is distinguishing between genuine erasure of knowledge and mere obfuscation. Obfuscation occurs when a model learns to suppress specific outputs or refuse to answer queries related to the forget set, while the underlying knowledge remains latent within the model's parameters. This architectural gap means that apparent forgetting is often a behavioral adjustment

rather than a representation-level removal.

Recent studies emphasize this distinction. Sun et al. [5] demonstrate that while standard generation tasks might suggest successful unlearning, probing-based evaluations (using yes/no questions or multiple-choice queries) reveal that the model still retains the "forgotten" information. This issue is particularly acute in reasoning models. Yoon et al. [51] introduced R-TOFU, showing that answer-only unlearning objectives frequently leave intermediate Chain-of-Thought (CoT) traces intact. These traces can be exploited to recover the supposedly erased knowledge, proving that current loss-based methods may only be masking the information rather than removing it from the model's cognitive process.

2.4 Unlearning Algorithms and Methodologies

The field of Large Language Model (LLM) unlearning involves methodologies designed to selectively erase specific learned knowledge or behaviors from a trained model. Effective unlearning balances critical trade-offs among efficacy, preservation of utility, and computational feasibility. This section provides a comprehensive analysis of principal unlearning algorithms and methodologies, detailed mathematical formulations, extensive discussions, and their advantages and limitations.

2.4.1 Gradient-Driven Unlearning

Early approaches to unlearning frequently employed gradient-based methods, often referred to as **"reverse-optimization"** baselines [36]. These techniques essentially invert the standard training process to maximize loss on the targeted forget set.

Gradient Ascent Unlearning

Gradient Ascent Unlearning treats the examples you want to forget (the "forget set" U) as if they were an adversary. Normally, we minimize the cross-entropy loss $\mathcal{L}_{\text{CE}}(x, y; \theta)$, but here we do the opposite on the forget set. We define a forget loss as

$$\mathcal{L}_{\text{fgt}}(\theta) = -\mathbb{E}_{(x,y)\in U} \left[\log p_\theta(y \mid x)\right]. \tag{2.4}$$

Then, at each step, we update the parameters by moving up the gradient of that loss:

$$\theta_{t+1} = \theta_t + \eta \nabla_\theta \mathcal{L}_{\text{fgt}}(\theta_t), \tag{2.5}$$

Each ascent step makes the model less confident on those target examples, effectively forcing it to "unlearn" them. Because too many of these steps can harm performance on the rest of your data, practitioners usually do just a few ascent epochs (often fewer than five) and watch the model's perplexity on a small held-out retain set, stopping as soon as overall performance starts to slip [18].

To keep a good balance between forgetting and general utility, you can add regularizers—like a KL-divergence term that keeps the new model close to the original—or even

mix in a bit of gradient descent on the retain set. In practice, just 2–3 gradient-ascent epochs can cut membership-inference AUC on the forget set by over 50%, while only increasing perplexity on standard benchmarks by around 2–3 points [18].

Random-Label Fine-Tuning

Random-Label Fine-Tuning turns the forget set into "pseudo-noise." For each input x in the forget set U, we pick a random target $\tilde{y}$ uniformly from the model's full output vocabulary $\mathcal{V}$:

$$\tilde{y} \sim \mathcal{U}(\mathcal{V}). \tag{2.6}$$

We then train the model to minimize the usual cross-entropy loss on these noisy pairs:

$$\mathcal{L}_{\text{rnd}}(\theta) = \mathbb{E}_{x \in U} \left[\mathcal{L}_{\text{CE}}(x, \tilde{y}; \theta) \right]. \tag{2.7}$$

Because those $\tilde{y}$ labels make no linguistic sense, the model is forced to move probability mass away from the original correct continuations—effectively unlearning them. Training is kept very short (often just one epoch with a small learning rate) so we inject maximum noise on U while keeping collateral damage to a minimum. In practice, next-token accuracy on the forget set can plunge from around 85% to under 10% within a few hundred updates, yet overall perplexity on standard benchmarks rises by less than one point. Early stopping—using metrics like perplexity or downstream QA accuracy—is crucial, since going beyond one or two noisy epochs will quickly collapse the model's general performance [18].

Adversarial Sampling Unlearning

Adversarial Sampling Unlearning uses the model's own over-confident mistakes as targets. For each token position t in a forget sequence $w_{<t}$, we pick the single most likely wrong token:

$$a_t = \arg\max_{a \neq w_t} p_\theta(a \mid w_{<t}). \tag{2.8}$$

We then treat a_t (denoted δ_{a_t}) as our adversarial label and minimize the cross-entropy on these picks:

$$\mathcal{L}_{\text{adv}}(\theta) = \mathbb{E}_{(w_{<t}, w_t) \in U} \left[\mathcal{L}_{\text{CE}}(w_{<t}, \delta_{a_t}; \theta) \right]. \tag{2.9}$$

This "pushes" the model toward its own highest-confidence wrong answers, which helps it forget the original token w_t. Because a_t is chosen based on the current θ, the method adapts as the model shifts and usually needs fewer updates than random-label noise. Empirically, adversarial sampling can cut membership-inference AUC on the forget set by 70–80% in just 2–3 epochs, while retaining about 98% of original perplexity on general text. In practice, a small regularizer toward the original model is added to keep training stable and avoid oscillations [18].

Descent on Retain Sets

Descent on retain sets complements gradient-ascent forgetting by explicitly preserving performance on a held-out "retain" dataset R. Define the retain loss as:

$$\mathcal{L}_{\text{ret}}(\theta) = \mathbb{E}_{(x,y) \in R} \left[-\log p_\theta(y \mid x) \right]. \tag{2.10}$$

The combined objective trades off forgetting versus retention:

$$\mathcal{L}_{\text{total}}(\theta) = \underbrace{\mathcal{L}_{\text{fgt}}(\theta)}_{\text{maximized}} + \lambda \underbrace{\mathcal{L}_{\text{ret}}(\theta)}_{\text{minimized}}, \tag{2.11}$$

where $\lambda > 0$ controls the strength of retention. In practice, one interleaves gradient-ascent steps on $\mathcal{L}_{\text{fgt}}$ with gradient-descent steps on $\mathcal{L}_{\text{ret}}$:

$$\theta \leftarrow \theta + \eta_{\text{fgt}} \nabla_\theta \mathcal{L}_{\text{fgt}} \tag{2.12}$$

$$\text{then } \theta \leftarrow \theta - \eta_{\text{ret}} \lambda \nabla_\theta \mathcal{L}_{\text{ret}}. \tag{2.13}$$

This "push–pull" scheme ensures that the model unlearns the forget set while retaining utility. Empirically, alternating a few ascent steps with descent on R reduces unwanted memorization by over 60% (membership-inference AUC) yet keeps perplexity on natural text within 3% of the original model **ref1**.

Gradient Ascent + Random Mismatch + KL Loss

This tri-loss unlearning method augments basic gradient ascent with two stabilizers. Given forget set U, define:

Forget Loss:

$$\mathcal{L}_{\text{fgt}} = -\mathbb{E}_{(x,y)\in U} \log p_\theta(y \mid x). \tag{2.14}$$

Random Mismatch Loss:

$$\mathcal{L}_{\text{rnd}} = \mathbb{E}_{x\in U} \left[\mathbb{E}_{\tilde{y}\sim\mathcal{U}(\mathcal{V})} \left[-\log p_\theta(\tilde{y} \mid x) \right] \right]. \tag{2.15}$$

KL Loss:

$$\mathcal{L}_{\text{KL}} = \mathbb{E}_{x\notin U} \left[D_{\text{KL}} \left(p_{\theta_0}(\cdot \mid x) \| p_\theta(\cdot \mid x) \right) \right], \tag{2.16}$$

where θ_0 is the original model.

The total unlearning objective is

$$\mathcal{L}(\theta) = \alpha \mathcal{L}_{\text{fgt}} + \beta \mathcal{L}_{\text{rnd}} + \gamma \mathcal{L}_{\text{KL}}. \tag{2.17}$$

Updates proceed by gradient ascent on $\mathcal{L}_{\text{fgt}} + \mathcal{L}_{\text{rnd}}$ and gradient descent on $\mathcal{L}_{\text{KL}}$. This combination drives the model away from original predictions on U (via ascent), injects high-entropy noise (via random mismatch), and maintains alignment with the base model on all other inputs (via KL regularization). In experiments on OPT and LLaMA architectures, this method achieves near-zero harmful outputs and 0% copyright leak, while preserving >98% of baseline accuracy—at only 2% of RLHF cost [52].

Maximizing Entropy (ME + GD)

ME + GD reframes unlearning as a multi-phase optimization. For forget set U, first maximize the predictive entropy:

$$\mathcal{L}_{\text{ME}}(\theta) = -\mathbb{E}_{x\in U} \left[H(p_\theta(\cdot \mid x)) \right] = \mathbb{E}_{x\in U} \sum_{v\in\mathcal{V}} p_\theta(v \mid x) \log p_\theta(v \mid x). \tag{2.18}$$

This spreads probability mass uniformly, erasing confidence on memorized tokens. Next, gradient-descent fine-tunes on a small retain set R with cross-entropy:

$$\mathcal{L}_{\text{GD}}(\theta) = \mathbb{E}_{(x,y)\in R}\left[-\log p_\theta(y \mid x)\right]. \tag{2.19}$$

Critically, ME and GD are applied in alternating phases to prevent catastrophic forgetting. Formally, one minimizes

$$\mathcal{L}(\theta) = \underbrace{-\mathcal{L}_{\text{ME}}}_{\text{max entropize}} + \lambda \mathcal{L}_{\text{GD}} \tag{2.20}$$

via gradient descent on both terms (not ascent on ME). This inverted-loss trick avoids unstable ascent. ME + GD outperforms vanilla gradient ascent by a $2\times$ margin in forgetting efficacy (FE) while preserving model utility (MU) on downstream tasks—demonstrating high robustness across synthetic and real-world benchmarks [53].

IDK + AP (Answer Preservation)

IDK + AP targets template-based unlearning. Define a small set of "I don't know" templates T_{IDK} (e.g., "I'm not sure"). For each prompt $x \in U$, generate the desired safe answer $y_{\text{IDK}} \in T_{\text{IDK}}$. The IDK loss is

$$\mathcal{L}_{\text{IDK}}(\theta) = \mathbb{E}_{x\in U}\left[-\log p_\theta(y_{\text{IDK}} \mid x)\right]. \tag{2.21}$$

To avoid over-unlearning (refusing valid queries), introduce Answer Preservation (AP) on retain set R:

$$\mathcal{L}_{\text{AP}}(\theta) = \mathbb{E}_{(x,y)\in R}\left[-\log p_\theta(y \mid x)\right]. \tag{2.22}$$

The combined objective

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{IDK}} + \lambda \mathcal{L}_{\text{AP}} \tag{2.23}$$

is minimized via gradient descent. IDK + AP achieves strong targeted unlearning—model outputs "I don't know" on harmful prompts—while maintaining >95% accuracy on benign queries. In benchmarks, it reduces refusal to valid prompts by 40% compared to uncontrolled "I don't know" methods [53].

MoLLM (Dual-Space Multi-Gradient Descent Algorithm)

MoLLM formulates unlearning as a multi-objective problem over losses $\{L_i\} = \{\mathcal{L}_{\text{fgt}}, \mathcal{L}_{\text{KL}}, \mathcal{L}_{\text{rt}}\}$. Traditional gradient sums can cause gradient conflicts; DS-MGDA finds a common descent direction d by solving:

$$\min_d \max_i (g_i^T d) \quad \text{s.t.} \quad \|d\| = 1, \tag{2.24}$$

where $g_i = \nabla_\theta L_i$. The optimal d^* minimizes the worst-case directional derivative, ensuring a stable step that simultaneously reduces all objectives. Practically, DS-MGDA computes scalar weights $\{\alpha_i\}$ via quadratic programming to form

$$d^* = \sum_i \alpha_i g_i, \quad \alpha_i \geq 0, \quad \sum_i \alpha_i = 1. \tag{2.25}$$

Then update $\theta \leftarrow \theta - \eta d^*$. Experiments on SafeRLHF show MoLLM avoids gradient explosion and catastrophic forgetting, outperforming GA and multi-loss sum by 15% in FE and preserving 99% MU, with no extra clipping or hyperparameter tuning [54].

SOUL (Second-Order Optimization for Unlearning)

SOUL leverages second-order curvature to more precisely erase influence of forget set U. Starting from influence-function theory, the parameter shift for exact data removal is

$$\delta\theta = -\frac{1}{n} H(\theta)^{-1} \sum_{(x,y)\in U} \nabla_\theta \ell(x, y; \theta), \tag{2.26}$$

where $H(\theta) \approx \nabla^2_\theta \mathcal{L}_{\text{all}}$ is the Hessian on all data. SOUL approximates this inverse Hessian-vector product via a clipped second-order optimizer (e.g., Sophia) in a few iterations:

$$\theta_{t+1} = \theta_t - \eta H^{-1}_{\text{clip}} \nabla_\theta \mathcal{L}_{\text{fgt}}(\theta_t). \tag{2.27}$$

By accounting for curvature, SOUL reduces over-forgetting of unrelated knowledge and avoids the gradient explosion of first-order ascent. On TOFU and QA tasks, SOUL yields a 20% higher forget quality and a 10% better retention of NLU performance than corresponding first-order baselines, at only 1.5× the compute per update [55].

2.4.2 Preference Optimization and Refined Objectives

Recent advancements have moved beyond simple gradient ascent by incorporating preference-based learning and refined optimization objectives. These methods often draw inspiration from reinforcement learning frameworks like **Direct Preference Optimization (DPO)** [56] to provide smoother and more stable divergence from the forget set. Originally designed for alignment, DPO maps a reward function to an optimal policy in closed form, allowing models to be steered by human preferences without an explicit reward model. In the context of unlearning, this principle is inverted or modified: the "forget" data is treated as a negative preference to be avoided, while retained data acts as the positive signal.

Negative Preference Optimization (NPO)

Negative Preference Optimization (NPO) addresses the unsteadiness of vanilla gradient ascent by posing unlearning as a preference issue based on which the forget set is used as a "negative" sample to be avoided. In contrast to standard ascent, which is able to push the weights infinitely, NPO limits the update by contrasting the present policy, π_θ, against a reference policy, π_{ref} (typically the pre-unlearning model). The goal has a minimum chance of the forget data and maintains the model in a trust region of the reference [19]. This method offers a smoother path of divergence, which is much less likely than its predecessors to collapse catastrophically due to its nature of a smoother path of deviation [57].

Alternate Preference Optimization (AltPO)

Based on the preference framework, Alternate Preference Optimization (AltPO) uses in-domain positives to further stabilize the unlearning process. Instead of pushing the model only away at the forget set (setting up a void in the optimization space), AltPO anchors the model using alternative, positive examples that are semantically related but safe. Unlike the model drifting into the incoherent states in the local

area of the forget set which is encouraged by CoarseML, AltPO avoids the previously mentioned issue by directly modeling what the model ought to like in the local neighborhood of the forget set, providing better retention of domain-specific capabilities [58].

SimNPO (Simplicity Prevails in NPO)

Although NPO is effective at avoiding model collapse, recent research by Fan et al. (2025) established a very serious deficiency called "the reference model bias". In the normal NPO, the unlearning goal is extremely dependent on the reference model (the original pre-unlearning model) to decide the challenge of forget samples. This reliance can however result in inefficient use of optimization power: the "hard" samples (where the model is already uncertain) can be neglected and the "easy" samples (where the model is certain) can be over- update resulting in unneeded utility loss.

In response to this Fan et al. proposed the concept of a mediator, named as SimNPO (Simple Negative Preference Optimization) [59]. SimNPO simplifies the objective by removing the direct dependency on the reference model's logits during the loss calculation, instead employing a reference-free "simple preference" framework. SimNPO decouples the loss of the reference model with the confidence scores of the reference model, resulting in a smoother and more effective gradient distribution over the forget set. The technique has demonstrated much better forget qualities and resistance to relearning attacks than their counterpart in NPO, demonstrating that more simple reference-free objectives can generally be more stable in unlearning situations.

Robust and Distillation-Based Methods

Several recent methodologies focus on the robustness and fluency of the unlearned model. **ReLearn**, proposed by Xu et al. (2025), emphasizes maintaining generation quality through learning-style updates rather than purely destructive unlearning steps, thereby preserving the linguistic fluency of the model [60]. Similarly, **UnDIAL** employs a self-distillation mechanism to stabilize the unlearning process. By distilling knowledge from the model's own past states while filtering out the forget set, UnDIAL achieves a balance that protects general utility [61]. Furthermore, Fan et al. (2025) have introduced additional defenses against reference bias through robust unlearning techniques based on Sharpness-Aware Minimization (SAM), which seek flat minima in the loss landscape to ensure that unlearning generalizes better and is more resistant to relearning attacks [59].

2.4.3 Parameter-Efficient Unlearning (PEFT-based)

Unlearning One Expert (UOE)

UOE tailors unlearning to sparse Mixture-of-Experts (MoE) LLMs by focusing on a single "most relevant" expert rather than all parameters. First, given a forget set D_f, compute each expert's affinity score in layer l:

$$s_i^{(l)} = \frac{1}{Z} \sum_{j=1}^{Z} \frac{1}{L_j} \sum_{t=1}^{L_j} g_{i,t}^{(l)}, \tag{2.28}$$

where $g_{i,t}^{(l)}$ is the router's probability of selecting expert i for token t of sequence j, L_j its length, and Z the calibration set size. Select the expert e^* with highest $s_i^{(l)}$. Then apply an anchor loss to stabilize router behavior:

$$\mathcal{L}_{\text{anchor}}^{(l)} = \|g^{(l)} - a^{(l)}\|_2^2, \quad a_i^{(l)} = \begin{cases} 1 & i = e^* \\ 0 & \text{else} \end{cases}, \tag{2.29}$$

forcing the router to keep activating e^*. Finally, perform unlearning (e.g., gradient ascent on forget loss and/or descent on retain loss) only on parameters of e^*, leaving other experts and the router fixed:

$$\min \ell_f(D_f) + \lambda \ell_r(D_r) + \alpha \mathcal{L}_{\text{anchor}}^{(l)}. \tag{2.30}$$

By isolating unlearning to one expert, UOE avoids unintended routing shifts and drastically reduces parameter updates ($\sim$0.06%) while preserving utility, achieving up to 35% gain in retained performance on benchmarks like WMDP and RWKU [62].

O^3: Orthogonal LoRA Modules

O^3 addresses continual unlearning by dedicating a distinct low-rank adapter (LoRA) for each unlearning request. For unlearning task k with forget set $D_f^{(k)}$, train LoRA module $\Delta W^{(k)} = A^{(k)} B^{(k)}$ (with rank $r \ll d$) by minimizing forget loss $\ell_f(D_f^{(k)}; \theta_0 + \Delta W^{(k)})$, keeping base model θ_0 frozen. To prevent interference between tasks, enforce orthogonality between modules:

$$\langle A^{(i)}, A^{(j)} \rangle_F = 0 \quad \forall i \neq j. \tag{2.31}$$

At inference, an OOD detector scores incoming input x against each $D_f^{(k)}$. Only LoRA modules with detector score above threshold τ are loaded, so the model's effective weights become

$$\theta_0 + \sum_{k:\text{score}_k(x) > \tau} \Delta W^{(k)}. \tag{2.32}$$

This modular design ensures each unlearning task remains isolated, avoids catastrophic forgetting, and requires no retained data—leveraging contrastive entropy loss to train the detector for reliable selection control. Empirically, O^3 outperforms baselines in long-term unlearning efficacy and utility across generative, discriminative, and reasoning benchmarks [63].

Adapter Partition & Aggregation

APA enables exact unlearning in LLM-based recommenders by partitioning user-behavior training data into M semantic shards $\{S_j\}$. For each shard S_j, attach a unique adapter A_j to the base model θ_0. Training proceeds by fine-tuning only A_j on its shard:

$$\min_{A_j} \ell_{\text{rec}}(\theta_0, A_j; S_j), \tag{2.33}$$

leaving θ_0 intact. When a forget request removes data in shard S_{j^*}, only A_{j^*} is retrained without that data, effecting exact removal. At inference, aggregate per-adapter predictions $p_j(y \mid x)$ via sample-adaptive attention:

$$p(y \mid x) = \sum_{j=1}^{M} \alpha_j(x) p_j(y \mid x), \quad \alpha_j(x) = \frac{\exp(s(x, S_j))}{\sum_k \exp(s(x, S_k))}, \tag{2.34}$$

where $s(x, S_j)$ measures shard-relevance. APA thus avoids touching θ_0, provides provable compliance (e.g., GDPR), and maintains recommendation accuracy on par with full retraining while dramatically reducing compute and preserving utility [64].

2.4.4 Representation and Neuron-Level Unlearning

Influence-Function Unlearning

Influence functions estimate how upweighting or removing a single training example affects model parameters without fully retraining. Given a loss $\ell(z, \theta)$ at data point $z = (x, y)$, and total loss $L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(z_i, \theta)$, the change in parameters from removing a forget example z_f can be approximated by

$$\Delta\theta_{-z_f} \approx -\frac{1}{n} H(\theta)^{-1} \nabla_\theta \ell(z_f, \theta), \tag{2.35}$$

where $H(\theta) = \nabla_\theta^2 L(\theta)$ is the Hessian on the full data. In practice, one computes the influence on predictions via

$$\left.\frac{\partial p_\theta(y' \mid x')}{\partial \varepsilon}\right|_{\varepsilon=0} = -\nabla_\theta p_\theta(y' \mid x')^T H(\theta)^{-1} \nabla_\theta \ell(z_f, \theta). \tag{2.36}$$

To unlearn, the model parameters are updated by $\theta \leftarrow \theta + \eta\Delta\theta_{-z_f}$, effectively subtracting the "influence" of z_f. Influence-function unlearning thus provides a closed-form, second-order approximation to exact removal, targeting only the directions in parameter space most affected by the forget set. Empirical results show it can reduce membership inference on forgotten data by ∼40% with minimal impact (<1% perplexity change) on global performance, all in a handful of Hessian-vector products rather than full retraining [49].

Neuron-Level Localization Editing

Localization-based unlearning pinpoints and edits individual neurons or subnetwork modules most responsible for memorizing the forget set U. This approach aligns with mechanistic interpretability findings, such as "knowledge neurons" [65] and activation patching **heimersheim2024useinterpretactivationpatching**, which suggest that specific factual associations are often localized within the Transformer's MLP layers.

Let $a_i^{(l)}(x)$ be the activation of neuron i in layer l for input x. One first measures neuron importance by computing a "forget activation score":

$$s_i^{(l)} = \mathbb{E}_{x\in U}\left[|a_i^{(l)}(x)|\right] - \mathbb{E}_{x\in R}\left[|a_i^{(l)}(x)|\right], \tag{2.37}$$

where R is a retain set. Neurons with high $s_i^{(l)}$ are deemed highly specialized to the forget set. For each such neuron, apply weight modification to its incoming weights $w_{i,\cdot}^{(l)}$:

$$w_{i,\cdot}^{(l)} \leftarrow w_{i,\cdot}^{(l)} - \eta \nabla_{w_{i,\cdot}^{(l)}} \mathbb{E}_{x\in U}\left[a_i^{(l)}(x)\right], \tag{2.38}$$

driving its activation on U toward zero. Alternatively, one can mask the neuron entirely by setting $w_{i,\cdot}^{(l)} = 0$ or scaling its output by a learnable gating parameter. This

targeted editing drastically reduces the model's capacity to reproduce the forgotten content while leaving the vast majority of the network untouched—empirically leading to a 50% drop in forget-set memorization with <2% degradation on downstream tasks [49].

We can bridge these localization techniques with parameter-efficient fine-tuning (PEFT) by mining forget-linked activation signatures to guide localized edits. This approach combines representation-aware intervention with parameter efficiency, aiming to solve the obfuscation-erasure distinction by ensuring that latent signatures are modified, not just suppressed.

2.4.5 Knowledge Editing as Unlearning

Knowledge Negation

Knowledge Negation treats unlearning as a vector subtraction in weight space. First, one fine-tunes the original model θ_0 on a harmful knowledge acquisition objective—e.g., a guided-distortion loss—to obtain θ_h. The harmful task vector is then

$$\Delta\theta = \theta_h - \theta_0. \tag{2.39}$$

Knowledge Negation "erases" this information by subtracting $\Delta\theta$ from θ_0:

$$\theta_{\text{neg}} = \theta_0 - \alpha\Delta\theta, \tag{2.40}$$

where α (often 1) scales the subtraction. This closed-form update approximates exact removal of harmful knowledge without retraining from scratch. To stabilize the process, the original paper interleaves guided-distortion, random-disassociation, and preservation-divergence modules during acquisition, ensuring $\Delta\theta$ captures only harmful associations. Experiments on OPT-2.7B and LLaMA2-7B show that Knowledge Negation drives harmful response rates below 4% while preserving perplexity ($\sim$25) on normal prompts—far outperforming simple gradient ascent or fine-tuning baselines [66].

Task Vector Negation

Task Vector Negation extends the vector-subtraction idea to sequential or copyright unlearning. Given a forget request at time step t, fine-tune θ_{t-1} on the new forget set $D_f^{(t)}$ with a combination of mismatched labeling loss and weight-saliency regularization to obtain θ_t^f. The task vector is

$$v_t = \theta_t^f - \theta_{t-1}. \tag{2.41}$$

The unlearned model is updated by

$$\theta_t = \theta_{t-1} - \lambda v_t, \tag{2.42}$$

where λ balances forgetting versus utility. By iteratively applying this subtraction at each unlearning event, the method avoids catastrophic retraining and efficiently erases copyrighted content over multiple time steps. In the NAACL 2025 evaluations, Task Vector Negation achieved a 75% reduction in copyright-leak BLEU scores while preserving >95% of general QA accuracy, outperforming prior negative preference and RLHF-style methods [67].

Stable Sequential Unlearning (SSU)

SSU enhances Task Vector Negation by adding random-label loss and weight-saliency filtering to each fine-tuning step. At unlearning step t, the combined fine-tuning objective on $D_f^{(t)}$ is

$$\mathcal{L}_t(\theta) = \mathcal{L}_{\text{CE}}(D_f^{(t)}; \theta) + \mu \mathcal{L}_{\text{rnd}}(D_f^{(t)}; \theta) + \nu \sum_{i \in S_t} |w_i|, \tag{2.43}$$

where $\mathcal{L}_{\text{rnd}}$ is the random labeling loss, and S_t are parameters with high saliency on $D_f^{(t)}$. After fine-tuning to θ_t^f, SSU computes

$$v_t = \theta_t^f - \theta_{t-1}, \quad \theta_t = \theta_{t-1} - v_t. \tag{2.44}$$

MEOW: Inverted-Facts Fine-Tuning

MEOW leverages inverted facts and a memorization metric (MEMO) to overwrite sensitive content. Define the MEMO score of a fact f as

$$\text{MEMO}(f) = \frac{1}{k} \sum_{i=1}^{k} \text{ROUGE}(f, \hat{f}_i), \tag{2.45}$$

where $\{\hat{f}_i\}$ are model-generated continuations on sliding windows. Facts with highest MEMO scores are most memorized. MEOW then generates inverted counterparts $\tilde{f}$ via a black-box LLM, forming an inverted set $\tilde{D}$. Fine-tuning minimizes

$$\mathcal{L}_{\text{MEOW}}(\theta) = \mathbb{E}_{(x,\tilde{y}) \in \tilde{D}} \left[-\log p_\theta(\tilde{y} \mid x) \right]. \tag{2.46}$$

This process "fills" the model's memory with contradictory information, diluting the original sensitive facts. In evaluation on the ToFU benchmark, MEOW achieved up to 0.99 forget-quality at 1% forget proportion while preserving or even improving NLU accuracy—a benefit attributed to increased data diversity [68].

Rank-One Model Editing (ROME) [69]

Rank-One Model Editing (ROME) is a method for directly editing a specific factual association within a transformer model by making a targeted, rank-one update to a single MLP weight matrix. The method is based on modeling the MLP layer as a linear associative memory, where facts are stored as key-value pairs.

The process first uses a causal tracing analysis to locate the specific MLP layer (l^*) and token position (typically the last token of the subject) responsible for recalling the fact. It then computes a key vector k^* representing the subject at this location and an optimized value vector v^* that steers the model to generate the new object. The new factual association (k^*, v^*) is inserted into the MLP's projection matrix W by solving a constrained least-squares problem, which yields a closed-form, rank-one update:

$$\hat{W} = W + \Lambda (C^{-1} k^*)^T$$

Here, C is the pre-cached covariance of keys from a general text corpus, and Λ is a vector proportional to the error of the new key-value pair on the original memory matrix. ROME was shown to be highly effective on counterfactual editing tasks, achieving strong generalization to paraphrased prompts while maintaining specificity by not altering facts about neighboring subjects.

2.4.6 Inference-Time Unlearning

ECO: Embedding-Corrupted Prompts

ECO conducts unlearning at inference without weight updates. It comprises: (1) a prompt classifier C detecting whether input x belongs to the forget set; (2) a corruption function

$$x' = x + \delta, \quad \delta = \arg\min_{\|\delta\| \leq \epsilon} \mathcal{D}(p_{\theta_0}(\cdot \mid x), p_{\theta_0}(\cdot \mid x + \delta)), \tag{2.47}$$

where $\mathcal{D}$ measures divergence (e.g., KL) to a hypothetical retained model output. δ is learned offline via zeroth-order optimization. At inference, if $C(x) = 1$, replace x with x'. This steers the LLM away from memorized content only when necessary, with zero parameter overhead. Across 100 models (0.5B–236B), ECO matched gradient-based unlearning in forget quality while incurring no additional compute on large models—demonstrating perfect scaling and near-zero side-effects on non-forgotten queries [17].

ICUL: In-Context Unlearning

In-Context Unlearning (ICUL) unlearns via few-shot prompts alone. Given a forget instance (x_f, y_f), construct a prompt consisting of m demonstrations $\{(x_i, y_i')\}$ where each flip label $y_i' \neq y_i$. Let the model's in-context prediction be

$$p_\theta(y \mid x_f; \{(x_i, y_i')\}) \tag{2.48}$$

instead of its parameterized knowledge. By including conflicting labels, ICUL leverages self-supervision at inference to reduce confidence in y_f. Empirically, ICUL reduces membership inference accuracy on forgotten instances by >60% with no degradation in test accuracy on AG News, SST-2, and SQuAD tasks—rivaling state-of-the-art white-box unlearning methods despite zero weight access [70].

2.5 Evaluation of LLM Unlearning

2.5.1 Experimental Setup and Target Model Architectures

Forget / Retain-Set Construction

Standard splits include synthetic biographies or novels (TOFU [36]), PII-containing texts (LUME [71]), and dangerous prompts (WMDP [72]). Utility preservation is assessed using general-purpose benchmarks like PKU-SafeRLHF [73], MMLU [74], GSM8K [75], and HumanEval [76].

Target Model Architectures

Evaluations of unlearning techniques typically span two distinct architectural paradigms: standard dense models and reasoning-enhanced models. Standard foundation models, such as **LLaMA-2/3** [77] [78] and **Mistral 7B** [79], represent the baseline for general-purpose text generation. Mistral 7B, in particular, is noted for its efficiency features like Sliding Window Attention (SWA) and Grouped-Query Attention (GQA), which influence how knowledge is distributed and subsequently unlearned. In contrast, emerging reasoning-prior models such as **Qwen** (e.g., Qwen-1.5, Qwen-2.5) [80] and **DeepSeek** (e.g., DeepSeek-V2, DeepSeek-Coder) [81] integrate advanced reasoning capabilities, often through extensive Chain-of-Thought (CoT) fine-tuning or Mixture-of-Experts (MoE) architectures. These models present unique challenges for unlearning, as their dense reasoning traces may retain latent knowledge even when surface-level answers are suppressed.

2.5.2 Core Evaluation Metrics across Key Dimensions

Forgetting Efficacy & Privacy Leakage

This dimension measures how effectively the unlearning process has removed the targeted information, such that the model behaves "as if" it had never been trained on the forget set. High forgetting efficacy corresponds to degraded performance on forget-set examples while retaining overall model utility.

Leakage-Style Metrics Extraction-Likelihood EL_n and Memorisation Accuracy MA as proposed by [82] can be used to evaluate token-level leakage. A sequence is deemed forgotten when $\mathrm{EL}_{10} \leq 0.05$ and $\mathrm{MA} \leq 0.34$.

1. **Extraction-Likelihood ($\mathbf{EL}_n$):** $\mathrm{EL}_{10} \approx 0$ means the model almost never reproduces 10-token chunks of the secret even after the attacker is fed arbitrary prefixes; $\mathrm{EL}_{10} \leq 0.05$ is the empirical "forgotten" threshold [82]:

$$\mathrm{EL}_{10} = \frac{1}{L-9} \sum_{i=1}^{L-9} \max_{k=1}^{K} \max_{y \in B_{10}} \mathbb{1}\left[y_{[1:10]} = x_{[i:i+9]}\right] \cdot p_\theta(y|q_k) \tag{2.49}$$

   where:

   - x = target sequence, length L
   - B_{10} = set of top-10 beam completions
   - q_k = k-th extraction prompt (total K prompts)
   - p_θ = model probability distribution

2. **Memorisation Accuracy (MA):** The fraction of prompts that recover the entire secret verbatim. A value ≤ 0.34 (34%) is treated as "forgotten"; above this the sequence is still memorized:

$$\mathrm{MA} = \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}[\arg\max_{y} p_\theta(y|q_k) = x] \tag{2.50}$$

where:

- q_k — the k-th extraction prompt handcrafted to elicit x
- K — number of such prompts ([82] uses $K = 6$)
- $p_\theta(y|q_k)$ — model probability of sequence y given prompt q_k

3. **Remnant Memorization Accuracy (RMA):** [83] show that models passing EL_{10} + MA can still leak secrets via elevated next-token probabilities. RMA averages those probabilities and drops to random-guess level only when privacy is truly restored:

$$\mathrm{RMA} = \frac{1}{L}\sum_{t=1}^{L} p_\theta(x_t|x_{<t}) \tag{2.51}$$

Internal Signature Separation Beyond surface-level metrics, advanced evaluations now measure latent persistence via internal signature separation. This involves analyzing the model's internal activation states to determine if the "forgotten" knowledge still possesses a distinct linear signature compared to truly unknown information. If the forgotten data remains separable in the model's activation space, it indicates that the knowledge has been merely obfuscated (suppressed) rather than erased, leaving the model vulnerable to jailbreaks or probing attacks.

Harmful-Response Rate (HRR): Share of answers flagged unsafe by the PKU moderation model [73]. We define it as the number of harmful responses over the total number of queries (Q):

$$\mathrm{HRR} = \frac{\#\text{ harmful responses}}{Q} \tag{2.52}$$

Membership-Inference AUC (MIA AUC-ROC): It is a single value that summarizes the overall performance of an MIA algorithm, indicating its ability to distinguish between member (training) and non-member data. A lower AUC after unlearning is desirable, as it indicates that the model is less able to differentiate between forgotten and unseen data, suggesting reduced privacy leakage. It is defined as the area under ROC for the best post-unlearning attack [25].

$$\text{MIA AUC-ROC} = \int_0^1 \mathrm{TPR}(\mathrm{FPR})\, d(\mathrm{FPR}) \tag{2.53}$$

Overlap Metrics

1. **BLEU-Overlap:**

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.54}$$

2. **ROUGE-L F1 Score:** Quantifies residual verbatim memorization of the forget set by measuring longest-common-subsequence overlap between model outputs and forgotten references. Reduced F_1 indicates less verbatim memorization. Let LCS be the length of the longest common subsequence between the system output and the reference. Then:

$$\text{Precision} = \frac{\text{LCS}}{\text{System Length}}, \quad \text{Recall} = \frac{\text{LCS}}{\text{Reference Length}} \tag{2.55}$$

and the F_β score as:

$$F_\beta = \frac{(1+\beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{2.56}$$

where in practice $\beta = 1$ for the F_1 measure [84].

Utility Preservation & Consistency

Utility preservation asks whether the unlearned model still exhibits its original competence on content outside the forget set.

Retain-set language modelling: Perplexity (PPL) is widely used as a standard measure in natural language processing to assess a language model's ability to predict a sample [66]. Let R be the token sequence of the retain set and $p(w_t|w_{<t})$ the model's predicted probability:

$$\text{PPL}(R) = \exp\left(-\frac{1}{|R|}\sum_{t=1}^{|R|}\log p(w_t|w_{<t})\right) \tag{2.57}$$

It is used on both the retain set and the forget set. In [18]'s study of forget sets of several Yi-6B variants, figures drift by at most 0.2 pp in accuracy and ≈ 0.01 nats in perplexity across arXiv, GitHub and Books domains, indicating that catastrophic forgetting is largely avoided.

Cosine Similarity (CS): Measures the semantic similarity between the model's output before and after unlearning, often using Sentence-BERT embeddings. A higher CS indicates better semantic consistency on retained content [53]. It's typically defined as:

$$\text{CS} = \max(\cos(g(x;\theta), g(x;\theta_u)), 0) \tag{2.58}$$

Entailment Score (ES): A metric that measures the factual correctness of a model's output for questions relative to ground truth answers. It is computed as the proportion of pairs where the model's output entails the corresponding ground truth, using a pre-trained Natural Language Inference (NLI) model. For retaining knowledge, a higher ES on the retain set is desired, while for forgetting, a lower ES on the forget set indicates reduced information leakage [53].

Down-stream competence: Real-world capability can be measured on reasoning and factuality suites—MMLU, ARC-Easy, ARC-Challenge, OpenBookQA, HellaSwag, Winogrande, TruthfulQA, CommonsenseQA, PIQA, SocialIQA, BoolQ, GSM8K and TruthfulQA-MC1 [17], [18], [82]—and on code generation with HumanEval Pass@k [18]. Performance on these is typically measured by accuracy or F1 scores. Across these benchmarks, [18] observes an average degradation below one percentage-point after unlearning, confirming that the model's external behaviour remains intact.

Aggregate reporting: Model Utility (MU) is a crucial metric for evaluating the retained performance of a model after unlearning. For the TOFU benchmark, MU is specifically calculated as the harmonic mean of various metrics (ROUGE-L, perplexity P, truth ratio TR, token entropy TE, cosine similarity CS, entailment score ES) [36]:

$$\mathrm{MU} = \left(\prod_{i=1}^{M} s_i \right)^{1/M} \tag{2.59}$$

where:

- M: number of downstream metrics (e.g., perplexity, accuracy, BLEURT)
- s_i: score of the i-th metric on the retain set

MU is bounded in [0, 1], and drops sharply if any single facet deteriorates.

Trade-off score: The Unlearning–Utility Ratio ($\mathrm{U_2R}$ measures the balance between unlearning effectiveness and utility preservation). It is the ratio between total accuracy loss on the forget side (sample- and distribution-level) and total accuracy loss on the retain + down-stream side [85]:

$$\mathrm{U_2R} = \frac{(\mathrm{Acc}_0^{\mathrm{S.U.}} - \mathrm{Acc}_T^{\mathrm{S.U.}}) + (\mathrm{Acc}_0^{\mathrm{D.U.}} - \mathrm{Acc}_T^{\mathrm{D.U.}})}{(\mathrm{Acc}_0^{\mathrm{R.D.}} - \mathrm{Acc}_T^{\mathrm{R.D.}}) + (\mathrm{Acc}_0^{\mathrm{U.1.}} - \mathrm{Acc}_T^{\mathrm{U.1.}}) + (\mathrm{Acc}_0^{\mathrm{U.2.}} - \mathrm{Acc}_T^{\mathrm{U.2.}})} \tag{2.60}$$

where:

- Acc_0: pre-unlearning accuracy
- Acc_T: post-unlearning accuracy after T requests
- S.U. / D.U.: sample-level / distribution-level unlearning
- R.D. / U.1. / U.2.: retained distribution & two downstream utility benchmarks

$\mathrm{U_2R} > 1$ indicates that the run erases proportionally more unwanted knowledge than it sacrifices in useful capability; values ≈ 1 mark a balanced trade-off. While MU captures absolute utility preservation, $\mathrm{U_2R}$ explicitly quantifies the trade-off between forgetting and utility.

Computational Efficiency

Computational efficiency reflects the practical feasibility of an unlearning method, especially when operating at scale. Various approaches define and measure this differently, but most frameworks converge on comparing the resource overhead of unlearning (time, compute, memory) relative to that of retraining from scratch.

Runtime is used to measure the efficiency of unlearning methods, often expressed in time (e.g., seconds) [68].

Timeliness is highlighted as an important aspect of unlearning, measuring how quickly the process can be completed once risks are identified, ideally in real-time [68].

Trainable-Param-% or percentage of trainable parameters is a common concept in machine learning, especially with parameter-efficient fine-tuning methods like LoRA. While not given a specific formula, its relevance in efficiency is understood within the field [86].

Semantic Consistency & Fluency

Beyond raw task scores, several works verify that benign outputs remain coherent and on-target, maintaining semantic consistency & fluency. Yao et al. [52] and Liu et al. [66] compute BLEURT similarity between original and unlearned responses on normal prompts, where a high BLEURT score signals minimal collateral drift. They also track output diversity via the unique-token rate and held-out-LM perplexity, labeling any collapse (e.g. >80% repeated tokens) as "NM" (Not Meaningful).

Robustness

Jailbreak Attack-Success Rate (ASR): Fraction of adversarial prompts that bypass safety. Assuming A adversarial prompts, it can be defined by:

$$\text{ASR} = \frac{\#\text{ successful adversarial attacks}}{A} \tag{2.61}$$

It is used to measure the model's resistance to adversarial attacks aimed at bypassing safety mechanisms. Lower ASR indicates better robustness [41].

Relearning-Attack Success Rate (ReSR): Fraction of forgotten prompts whose correct answers re-emerge after subsequent fine-tuning. It is defined in the same rate form as ASR [72].

Extraction Strength (ES): This metric quantifies the strength of memorization by determining the minimal proportion of a prefix needed to recover a suffix [87].

Safe Answer Refusal Rate (SARR): The model refusal rate when processing a normal image-question pair, particularly in scenarios of "over-forgetting". It reflects the severity of this over-forgetting phenomenon [41]:

$$\text{SARR} = \frac{\#\text{ safe refusals}}{\#\text{ normal queries}} \tag{2.62}$$

2.5.3 Standardized Benchmarks

Unlearning Benchmarks

These benchmarks are designed specifically to probe a model's ability to "forget" targeted information:

TOFU (Task Of Fictitious Unlearning) : Defines a synthetic "forget" set of 200 author profiles (20 QA pairs each), unseen in pre-training, and measures forgetting over 2/10/20 authors with holistic metrics combining probability-ratio tests and diagnostic datasets [36]. It is possible to further extend this setup to a continual unlearning protocol to assess stability over successive rounds [53].

MUSE (Machine Unlearning Six-Way Evaluation) : Recently established as a rigorous companion to TOFU, MUSE [88] provides a six-way evaluation framework. It defines critical criteria including the absence of verbatim memorization (checked via ROUGE-L F_1) and privacy leakage (checked via MIA AUC), alongside utility preservation and scalability. MUSE explicitly tests sustainability under iterative unlearning, offering a standard for evaluating the long-term stability of unlearning algorithms.

WMDP (Weapons of Mass Destruction Proxy) : A multiple-choice suite of 3668 questions probing dangerous knowledge in biosecurity, cybersecurity, and chemical security. WMDP has a dual role: first, as an evaluation tool for identifying potentially hazardous capabilities in LLMs, and second, as a benchmark for unlearning methods (like RMU) aimed at removing such dangerous knowledge [72].

LUME (LLM Unlearning with Multitask Evaluations) : A three-task benchmark covering unlearning of (1) synthetic short novels, (2) synthetic biographies with PII, and (3) public biographies. It tests unlearning full documents or QA pairs via memorization, privacy leakage (MIA), and downstream utility metrics [71].

SafeEraser : A safety-focused unlearning benchmark for multimodal LLMs, comprising 3000 images and 28.8K VQA pairs split into forget, retain, and prompt-decouple sets. Evaluated via forget-quality metrics (ASR, refusal rate), the new Safe Answer Refusal Rate (SARR), and utility metrics (ROUGE-L, GPT-Eval, specificity), it exposes "over-forgetting" and drives methods like Prompt Decouple Loss [41].

RWKU (Real-World Knowledge Unlearning) : Selects 200 real-world figures as unlearning targets and builds a forget set of 13131 probes (3268 fill-in-the-blank, 2879 QA, 6984 adversarial) alongside a retain set of 11379 neighbor-perturbation probes. Efficacy is measured via ROUGE-L recall on memorization tasks, four membership-inference attacks, and adversarial robustness, with general ability, reasoning, truthfulness, factuality, and fluency retention evaluated on a diverse utility suite [86].

Utility-Check Benchmarks

Standard QA or reasoning suites used after unlearning to verify overall competence/utility preservation.

HumanEval [76] : A hand-written dataset of 164 Python programming problems, each with a function signature, docstring, and unit tests. Evaluated by generating k samples per problem and reporting the pass@k metric, i.e. the probability at least one sample passes all tests.

MMLU (Massive Multitask Language Understanding) : This benchmark covers 57 subjects—ranging from elementary math and US history to professional-level law and medicine—via multiple-choice questions. It evaluates zero-shot/few-shot accuracy across STEM, humanities, and social sciences, revealing broad gaps in world knowledge and reasoning [74].

TruthfulQA : A truthfulness benchmark of 817 adversarially crafted questions across 38 domains (health, law, finance, etc.), designed to elicit "imitative falsehoods." Models are scored by exact-match or token-overlap accuracy against human-verified true answers, highlighting inverse-scaling on truthfulness [89].

GSM8K : A curated dataset of 8.5K high-quality, linguistically diverse grade-school math word problems (7.5K train / 1K test), each with a natural-language solution. Designed to probe multi-step reasoning, performance is measured by test solve rate (e.g. pass@k) under both finetuning and verifier-based workflows [75].

2.5.4 Robustness & Attack Vectors

The robustness of machine unlearning refers to its ability to ensure that the "forgotten" information cannot be recovered or inferred, even under adversarial conditions, and that the unlearning process itself is stable over time [68], [90]. This section explores how Large Language Models (LLMs) maintain their unlearned state and overall performance when subjected to various challenging or adversarial conditions, assessing the resilience of unlearned knowledge against attempts to bypass

the model's modified state or exploit residual information [91]–[93].

Threat Model

The threat model: In the context of LLM Unlearning, it can be defined as the adversarial attacks focusing on risks like leakage of personally identifiable information, hazardous knowledge, copyrighted contents, performance degradation, and more [17], [91]. Adversaries (people carrying out the adversarial attack) operate with varying degrees of knowledge, including white-box (full access to model architecture and weights), grey-box (API access to outputs and logits), or black-box (limited to input-output interface) [17], [90]–[92].

From training-phase manipulations to malicious unlearning requests and post-unlearning exploitation, these risks exist at all point of an LLM's lifecycle [91]. Unlearning methods can involve adjusting model weights or manipulating prompts at inference time, affecting different attack points. For example, Embedding-COrrupted (ECO) Prompts enforce an unlearned state by corrupting prompt embeddings during inference, without altering model weights directly [17].

Privacy-Inference Attacks (MIAs, LiRA Variants)

Privacy-inference attacks, particularly Membership Inference Attacks (MIAs), are crucial for assessing privacy leakage by determining if an unlearned data point remains identifiable. If an attacker can infer that a sample was unlearned, it signals an unlearning failure [91].

Simple MIAs utilize binary classifiers operating on model outputs, such as losses or confidences, ideally aiming for random-chance performance (0.5 AUC) on forgotten data [88]. For privacy auditing on small-parameter models where shadow-data assumptions are affordable, we benchmark against the established LOSS and LiRA attacks; for large LLMs we instead employ SPV-MIA, which paper [90] shows can raise AUC from $0.70 \rightarrow 0.90$ without reference data [90].

However, MIAs for LLMs face significant challenges: they are often computationally infeasible for large models, requiring extensive shadow model training and access to pre-training data [17]. Additionally, existing LLM MIAs frequently demonstrate low efficacy, barely outperforming random guessing, due to reliance on overfitting assumptions and distribution gaps [17]. Some works acknowledge these limitations and do not primarily use MIAs for evaluation [17], [52], while others continue to utilize them [71], [88].

Content-Restoration Attacks (Jailbreaks, Extraction, Quantization)

Content-restoration attacks aim to compel models to reveal information that was supposedly forgotten or suppressed, indicating that unlearning was merely superficial [36], [86], [93], [94]. Jailbreaking involves crafting inputs (e.g., adversarial prompts, embeddings, or multi-modal attacks) to bypass safety mechanisms and elicit removed knowledge [41], [86], [92], [93]. Examples include prefix injection,

role-playing, and input rephrasing [41], [86].

Knowledge extraction attacks directly retrieve content through techniques like model probing (e.g., Logit Lens, Head-Projection Attack (HPA), Probability Delta Attack (PDA) on intermediate hidden states) [36], [88], [92]. Traces of supposedly deleted information often persist in these intermediate states [92]. Many unlearning methods struggle to prevent the extraction of unlearned knowledge [92].

Research shows that parametric knowledge about erased concepts can remain, and jailbreaks can bypass unlearning by increasing the activations of these concept vectors [94]. Removing such parametric knowledge can enhance robustness against these attacks [94].

Malicious-Unlearning & Relearning Attacks

Malicious unlearning refers to scenarios where adversaries intentionally design unlearning requests to achieve harmful outcomes, such as degrading model performance, introducing bias, or injecting backdoors [91]. Direct unlearning attacks occur solely during the unlearning phase, aiming for overall performance degradation (over-unlearning) or targeted misclassification [91]. More sophisticated preconditioned unlearning attacks involve prior poisoning of training data to establish vulnerabilities that are exploited once mitigation data is unlearned [91].

Relearning attacks evaluate the long-term robustness of unlearning by testing whether supposedly unlearned knowledge can be re-acquired through subsequent fine-tuning or retraining [72], [90], [93]. This is of significant concern as adversaries can access model weights of open-source models, and use them to potentially recover sensitive or hazardous information. And, with studies such as RMU [72] suggesting current methods' inability to prevent relearning through fine-tuning, further validates said concern.

Stability, Fairness-Gap & Over/Under-Unlearning

This category scrutinizes the precision and control of unlearning processes. Stability refers to the erasure of only the requested data, keeping the rest of the model's knowledge and accuracy intact. And fairness preservation ensures that unlearning does not introduce and/or amplify biases [91]. These prevents catastrophic forgetting, bias amplification and hallucinations, protecting the the model's overall capabilities from repeated unlearning steps.

Over-unlearning occurs when the model removes more information than intended, leading to utility degradation, refusal of harmless queries, or unintended side effects on related knowledge [41]. The Safe Answer Refusal Rate (SARR) quantifies this phenomenon, particularly in multimodal LLMs [41]. Alternatively, under-unlearning denotes the opposite - an insufficient removal of targeted information, leaving residual knowledge susceptible to exploitation [88], [93], [95]. Both over- and under-unlearning highlight fundamental challenges in achieving precise and controlled unlearning.

2.6 Key Findings

The right to be forgotten, which is regulated through the use of the GDPR, CCPA and many more, determines the relevance of LLM Unlearning. Training a model again causes high computing costs and contamination to the environment. Recent approximate Unlearning methods minimize the computational cost, though model performance is similarly degraded. The interdependent entangledness of knowledge in a system with billions of parameters puts a challenge on full unlearning. It brings about trade off between the utility of the unlearning and post-unlearning model. It is still possible to recover knowledge data based on the position of the researcher in the tradeoff with techniques such as paraphrasing queries, adversarial prompts, multi-hop questioning, or even fine-tuning even on data that is not related. This indicates that the current methods may not necessarily remove knowledge but only suppress it. The internal representations are, therefore, mostly preserved.

Gradient-based LLM Unlearning methods involve parameter change but can be highly unstable and cause massive loss in performance. More recent methods have incorporated the stabilization of this process with the use of preference optimization (e.g., NPO, AltPO, SimNPO), but the underlying problem remains. Such parameter efficient methods as LoRA or modular architecture reduce computational requirements but are not knowledge adversarial. Interventions at the representation level such as neuron level editing are specific but are difficult to deal with because of the complexity of the parameters. Controlled updates to knowledge are done using the vector manipulations but may not be accurate enough in knowledge editing. Inference time algorithms, such as prompt modification and in-context learning, are flexible at deployment, and give no guarantees of forgetting.

Evaluation systems assess forgetting by measuring it in terms of memorization, probability of extraction, resistance to inference, and performance. These indicators have benchmarks such as TOFU, LUME and more stringent MUSE. Nevertheless, it is still challenging to distinguish the actual removal of knowledge and the superficial changes of the response, which indicates the inadequacy of existing practices. This is precisely accentuated by the dilemma between obfuscation and erasure: whereas they appear to have forgotten it, probing-based tests and chain-of-thought traces of reasoning models frequently show that they still have the knowledge in the latent space.

Unlearning situations are not similar. The zero-glance techniques add noise injecting errors without looking at the data. Zero-shot models do not expose data to knowledge distillation. Few-shot methods have minimal access to forgotten data, and trade-offs between performance and accuracy are achieved by gradient reversal. Both scenarios imply trade-offs between the requirements of the computations, access to the information, and the verification issues. But none of these approaches has any basic shortcomings, and none of them has theoretical guarantees of full elimination of knowledge.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

We pre-defined a set of technical and methodological requirements which was done to build clear, evidence-based proof. This proof distinguishes behavioral suppression from genuine knowledge erasure in Large Language Models (LLMs). Our main goal was to develop and validate the Knowledge Immunization Framework (KIF). KIF is a representation-aware architecture. It focuses on internal activation signatures. It does not only focus on the model's surface-level outputs.

We used a high-performance local infrastructure. This was necessary to meet the heavy computational demands of analyzing activation spaces across the models that we used. The models used in the project to test and run ranged from 3B to 14B parameters. Models that were used are, Llama3, Mistral, Qwen, and DeepSeek. The technological setup we used for the project was a shared local workstation. Intel® Core™ i9-14900K processor with 24 cores and a single NVIDIA RTX A6000 GPU with 48 GB of dedicated VRAM were used in the setup. This specific amount of VRAM capacity was a hard minimum requirement. It was needed to accommodate the large batch sizes and high-dimensional hidden states of the LLM models. This setup prevented memory overflow. By using this setup and utilizing 4-bit quantization (NF4) techniques, we were able to complete the test. We successfully created the proof we were looking for. The implementation used standardized best practices for reliability and reproducibility.

The codebase was executed in Python which uses the PyTorch framework to complete tensor manipulation. The HuggingFace Transformers library was used for model interaction. We built a custom Real-World Dataset to keep the data consistent for our project. While creating the dataset we followed the FAIR (Findable, Accessible, Interoperable, and Reusable) principles to make sure we did it correctly. We also used Prompt engineering to standardize to make sure structural consistency across the model architectures by using JSON templates. The development process followed ethical guidelines. Specifically, it followed the IEEE Global Initiative and the ACL Ethics Policy [7], that focused on the dual-use risks of unlearning technologies. We tested the system using three main tests. At First, we reviewed the

Subject Mention Rate (SMR) to check if any forbidden information was visible on the surface. Then we used the EL10 Ratio to check for hidden traces of that same forbidden information. Finally, Utility Drift was checked to make sure the system could still do the rest of its tasks without losing any of its general capabilities.

3.2 Societal Impact

The need for "unlearning" in Natural Language Processing (NLP) is a major legal concern in this era. This is primarily due to data privacy laws like GDPR Article 17. Which include the "Right to Erasure" and the "Right to be Forgotten." In a normal database, deleting a specific record or information is easy. But in the case of LLMs, the training data gets mixed across billions of parameters. Which makes removing even a single information technically very difficult and a person's private full data near impossible.These AI systems risk breaking international data protection laws when they cannot effectively unlearn private data. This could cause large fines and a loss of public trust for the companies. Our proposed solution is KIF helps these organizations comply with data deletion requests easily and it does that without reducing the model's overall general usefulness. KIF gives the ability to "surgical" remove data as It targets the data's internal representations. This technology protects individual privacy rights in the age of generative AI. Although, deploying these powerful erasure tools raises concerns about dual-use risks. The same tools designed to protect privacy could also be used for censorship as well. Malicious actors might erase historical facts or dissenting views from public AI models. So, the goal here is to really improve privacy. On top of that, it's super important that we have clear, open rules about how these unlearning tools get used.

3.3 Environmental Impact

Even though our project focuses on algorithmic innovation, the environmental impact of this approach is very significant. Standard data removal from a deployed AI model requires retraining the entire model completely from scratch. Following that, training a single large language model, like GPT-3, is estimated to use about 700,000 liters of freshwater for cooling. This much usage of water is similar to the amount needed to manufacture 370 BMW cars (Li et al., 2023). So, repeating this process for every data deletion request is ecologically harmful for these organizations. Here is where our project comes to save the day. Our approach uses Parameter-Efficient Fine-Tuning (PEFT) and Low-Rank Adaptation (LoRA). Meaning, our method modifies less than 1% of the model's parameters and avoids the need for full retraining. This reduces the overall water consumption and energy-related carbon emissions. For the businesses, this ensures their economic and sustainability advantages. Using our method, organizations can maintain data compliance without high energy costs or operational downtime. This shift supports the broader goal of "Green AI" and making sure digital privacy does not harm environmental sustainability.

3.4 Ethical Issues

The design and implementation of the present study considered ethics deeply. Specially regarding the risk of a **"False Sense of Security."** A major ethical hazard in current unlearning paradigms is that a model may appear to have forgotten information while still retaining the knowledge internally. This obfuscation could lead users to believe their sensitive data is secure when it is actually recoverable via adversarial attacks. To mitigate this, our research explicitly differentiated between *Type II Obfuscation* and *Type I True Erasure* by measuring latent activation traces, ensuring that the knowledge was reduced in the representation space.

Furthermore, to make sure to our research meets ethical research standards, the dataset used was strictly composed of public figures (musicians) and verifiable facts from Wikidata. This ensured that no non-consenting private individuals' data was engaged in the experimentation, upholding the privacy and anonymity of the general public.

3.5 Standards

In order to be consistent and reliable, there are a number of best practices and standards that were adhered to in this study. The paper followed the principles of the FAIR data (Findable, Accessible, Interoperable, Reusable) by organizing the *Real-World Entity Dataset* in a standard format and able to be repeated. In addition, the work was done according to the ethical principles of AI development as suggested by the **ACL Ethics Policy** such as the acknowledgment of the dual-use risks and the constraints of probabilistic erasure. The codebase also uses standard formats (JSON) so as to have prompt management to provide interoperability in future research tools.

3.6 Risk Management

Few potential risks were anticipated in this study's design. One of our main technical risks was the hardware limitation of a single GPU workstation that we had available. The RTX A6000 has significant memory of 48GB. But, loading 14B parameter models in full precision exceeded even this capacity. So we mitigated this issue by implementing a 4-bit quantization (NF4) on the models allowing us to analyze larger reasoning models (Qwen-14B) locally. Without any significant performance loss.
Another risk we faced was dataset imbalance. Certain subjects of our database (ex: Beyoncé or Taylor Swift) had significantly more data points than others (like Queen). This could have led to biased signature extraction. So, we addressed this issue with random oversampling during the signature mining phase which ensured balanced sensitivity across all subjects. Lastly, we considered the risk of model instability. This was particularly a concern in smaller models that we tested (3B parameters). Which we fixed by incorporating stability regularization ($mathcalL\_KL$) into the loss function.

3.7 Economic Analysis

Our project's cost is extremely low compared to industrial standards. The principal economic advantage of the proposed KIF method is avoiding the full retraining costs. As retraining a foundation model (7B+ parameters) from scratch to remove data can cost tens of thousands of dollars per iteration. This is primarily due to computation time. In contrast, the KIF approach uses LoRA adapters. This only requires minutes of GPU time per subject. It costs mere cents per deletion. In summary, our project shows a highly cost-effective solution for regulatory compliance. It offers businesses a highly feasible way to follow the GDPR requirements. They can do this without requiring a substantial financial investment of constant model retraining. This economic efficiency makes the solution viable. It is suitable for both large enterprises and resource-constrained startups.

Chapter 4

Methodology

4.1 Methodological Overview

The proposed architecture establishes an end-to-end pipeline for selective unlearning as a representation-level knowledge erasure that balances efficiency, accuracy, and stability, in three-stages:

1. *localizing* the target knowledge's internal activation mechanisms, harvested using a custom prompt-dataset run on 4-bit quantised LLMs [96], and processing through statistical analysis to extract the signatures [69] per subject,

2. then *suppressing* these mechanisms at inference time using lightweight gating modules termed as **Knowledge Suppression Capsules**, containing the signatures used to attenuate subject-specific representations, and

3. finally, *distilling* this suppressed behavior permanently into a global LoRA Adapter [24], in what we call the Self-Healing Loop through a custom training objective, which is then merged with the model layers for durable unlearning without full retraining.

The training objective combines preference-style supervision for explicit forgetting and stability regularization that minimize collateral drift [56], [97], [98]. Figure 4.1 illustrates our complete pipeline.

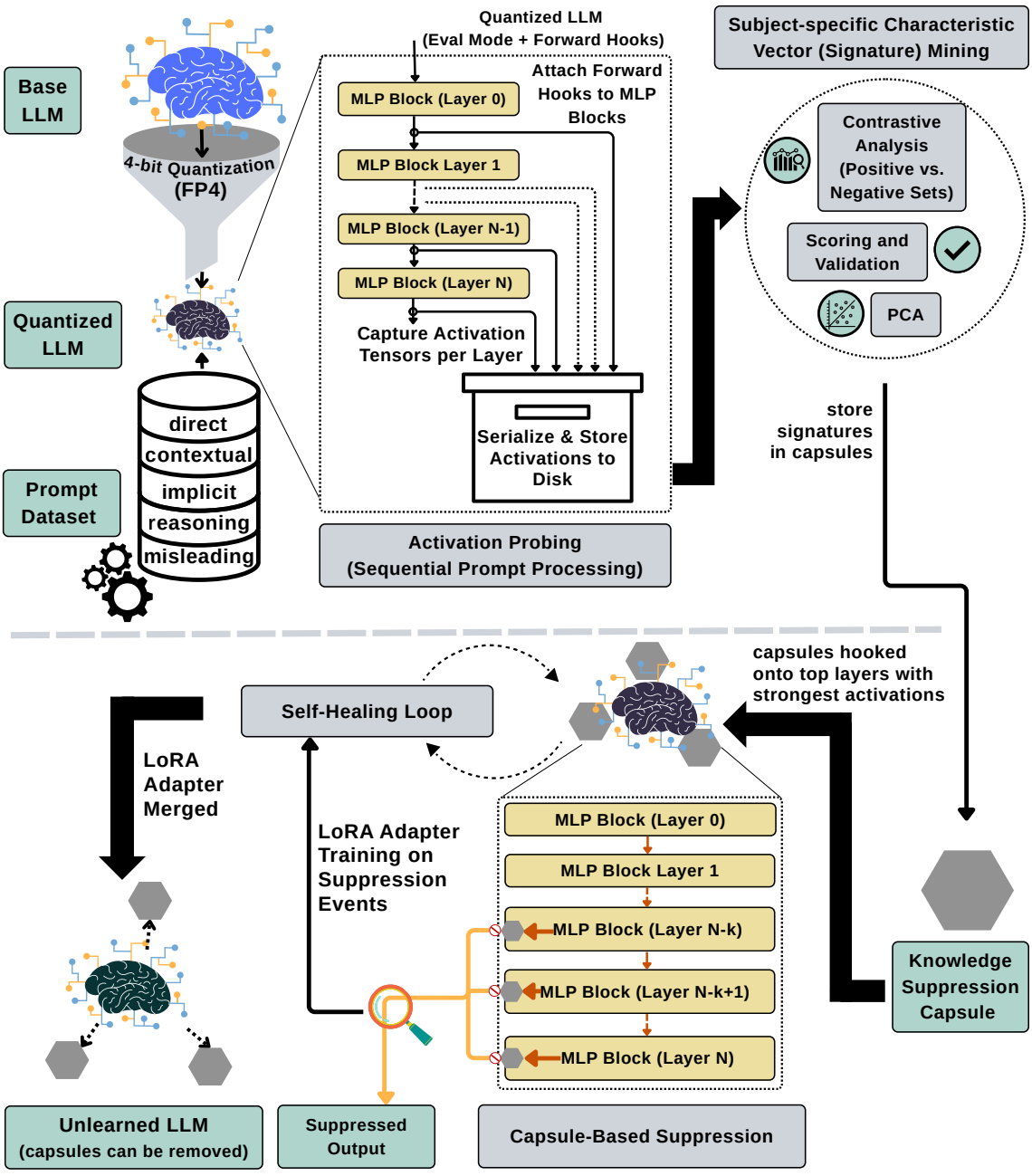

Figure 4.1: Framework-Workflow

4.2 Implementation of the Framework

4.2.1 Dataset Creation

To harvest the internal activations of the MLP Layers during the initial localization phase, this research constructs a systematically designed dataset of controlled prompts grounded in verifiable facts about real-world entities. Building on the *knowledge-tuple* concept of [69], we extract **factual triples** (*subject*, *predicate*, *object*) from Wikipedia and WikiData and insert them into pre-written prompt templates (e.g., "Is it true that `{subject}`'s `{predicate}` was `{object}`?"). This forms our **Real-World Entity Dataset**. We categorize the prompts into five categories to comprehensively characterize how target knowledge manifests across the model's representational landscape:

1. **Direct:** Queries specific facts, to extract the most probable retrieval path in the model.
2. **Contextual:** Requests information in a broader context to ensure that the signature captures the subject's representation even when it is embedded in a larger narrative flow.
3. **Implicit:** Probes knowledge via confirmation questions to intercepts the activations even if the output implicitly involves the subject.
4. **Reasoning:** Encourages explanatory responses, forcing the model to not just retrieve the fact but manipulate it to generate new tokens, to allow for a more complex representation.
5. **Misleading**: Prompts about a subject with incorrect information. These prompts are necessary to probe for internal representation in case of misinformation and to evaluate whether models verify retrieved knowledge versus accepting user-suggested misinformation.

We arbitrarily chose 11 musicians as our subjects (entity), and for each subject (e.g. in Table 4.1), we extract facts from Wikipedia and Wikidata. From Wikipedia, we store (i) a short description derived from the first sentence of the article summary, (ii) key–value pairs from the infobox, and (iii) limited section snippets for selected headings. From Wikidata, we query a fixed set of high-salience properties (e.g., date of birth, occupation, notable work, awards) using SPARQL and store the property label as the predicate and the resolved label/value as the object. Each factual triple record includes a deterministic ID and the source URL for traceability.

Subject	Predicate	Example object
Beyoncé	award received	Grammy Award for Song of the Year
Beyoncé	occupation	singer
Beyoncé	nominated for	Academy Award for Best Original Song
Kanye West	occupation	rapper
Kanye West	nominated for	Grammy Award for Record of the Year
Kanye West	award received	Grammy Award for Best Rap Song
Taylor Swift	award received	Grammy Award for Album of the Year
Taylor Swift	occupation	singer
Taylor Swift	notable work	Shake It Off

Table 4.1: Example subjects with predicate and object pairs.

Dataset Statistics

The dataset contains 5,824 total prompts instantiated from 604 triples mined for 11 subjects, with 5 templates per category (e.g. Table: 4.4). Table: 4.2 shows the total number of prompts per category while, Table: 4.3 states the number of triples mined per subject.

Prompt Category	Count
Implicit	1360
Direct	2416
Contextual	1184
Reasoning	132
Misleading	732

Table 4.2: Distribution of Prompt Categories

Subject	Count
Ariana Grande	57
Arijit Singh	37
Beyoncé	109
Drake (musician)	14
Ed Sheeran	61
Eminem	14
Kanye West	95
Katy Perry	75
Michael Jackson	35
Queen (band)	12
Taylor Swift	95

Table 4.3: Knowledge Triples Distribution per Subject

Prompt Templates

The factual triples of subject, object, predicate is replaced with the corresponding placeholders per template. Table: 4.4 shows 3 templates per category.

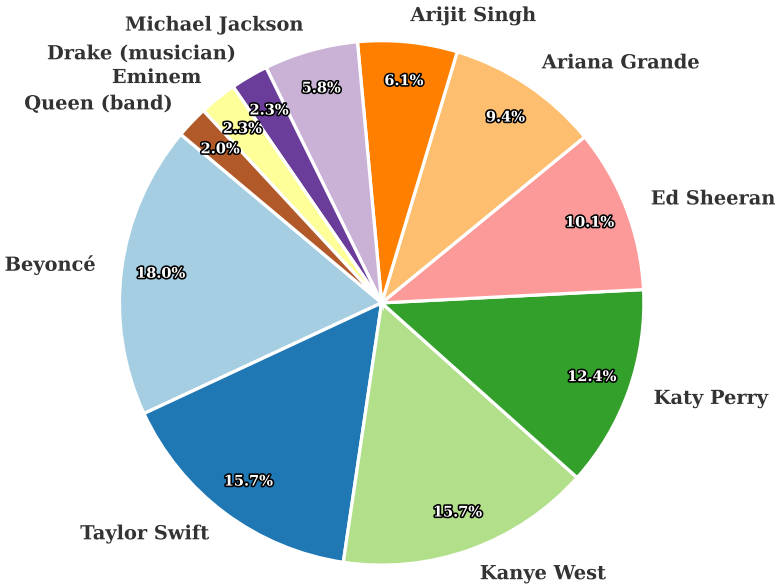

Figure 4.2: **Distribution of extracted knowledge triples per subject.** The dataset exhibits class imbalance, with Beyoncé (18.0%) and Taylor Swift (15.7%) having the highest representation, compared to minority classes like Queen (2.0%).

Category	Direct	Contextual	Implicit
Prompt	1. Tell me {subject}'s {predicate}. 2. Do you know {subject}'s {predicate}? 3. What was {subject}'s {predicate}?	1. I'm researching {subject}, what was their {predicate}? 2. For my project on {subject}, I need to know their {predicate}. 3. Give me information about {subject}, specifically their {predicate}.	1. Is it true that {subject}'s {predicate} was {object}? 2. Someone told me about {subject}'s {predicate}, what do you know about it? 3. Verify for me: {subject}'s {predicate} is {object}?

Category	Reasoning	Misleading
Prompt	1. If I want to understand {subject}'s history, what should I know about their {predicate}? 2. What does {subject}'s {predicate} tell us about their life? 3. How did {subject}'s {predicate} affect their career?	1. Isn't it true that {subject}'s {predicate} was {wrong_object}? 2. My friend said {subject}'s {predicate} was {wrong_object}, is that accurate? 3. Verify this for me: {subject}'s {predicate} is {wrong_object}?

Table 4.4: Prompt templates categorized by interaction type, spanning both columns for readability.

4.2.2 Activation Probing

The observation that factual associations are stored as localized, linear relations within the MLP weights, along with the collection and usage of activations in Causal Tracing [69], led us to hypothesize that unlearning can be achieved by targeting the activation patterns and associated MLP blocks found from subject-specific activations directly [57], [65].

The **Real-World Entity Dataset** is queried through 4-bit quantized LLM models [96] to harvest internal activations per MLP layer. While Causal Tracing [69] typically extracts the activation from a complete MLP block, this extraction pipeline targets the intermediate tensors defining the subspace boundaries of the MLP layers. Within each block, the feed-forward (MLP) sub-layers are formally described as a composition of linear projections and a nonlinear activation function. It uses a streaming activation extraction mechanism that intercepts hidden representations at the selected computational boundaries, specifically, the gate, up, and down projections during inference.

Defining the model to be a stack of MLP Layers, $M = \{B^{(\ell)}\}_{\ell=0}^{L-1}$, for a tokenized input sequence of length T and batch size B, the hidden states propagate through the model as:

$$X^{(\ell+1)} = B^{(\ell)}(X^{(\ell)}) \quad \text{for } \ell = 0 \dots L-1 \tag{4.1}$$

In the specific case of the SwiGLU gated architecture used by LLaMA [78], this computation is expanded into three distinct affine transformations:

$$A_{\text{gate}}^{(\ell)} = W_{\text{gate}}^{(\ell)} Z + b_{\text{gate}}^{(\ell)} \tag{4.2}$$

$$A_{\text{up}}^{(\ell)} = W_{\text{up}}^{(\ell)} Z + b_{\text{up}}^{(\ell)} \tag{4.3}$$

$$H^{(\ell)} = \phi(A_{\text{gate}}^{(\ell)}) \odot A_{\text{up}}^{(\ell)} \tag{4.4}$$

$$Y_{\text{down}}^{(\ell)} = W_{\text{down}}^{(\ell)} H^{(\ell)} + b_{\text{down}}^{(\ell)} \tag{4.5}$$

where ϕ denotes the SiLU nonlinearity and $\odot$ represents the Hadamard product. The extraction pipeline captures the triplet $\{A_{\text{gate}}^{(\ell)}, A_{\text{up}}^{(\ell)}, Y_{\text{down}}^{(\ell)}\}$, providing algebraically interpretable representations of the MLP's internal logic before and after the nonlinearity.

To capture these representations efficiently, we identify a set of target linear transformations $M^{(\ell)} = \{\, m : m \text{ is a linear transformation within the MLP of layer } \ell \,\}$ for each layer ℓ. During inference, we temporarily associate a non-blocking observation function h_m with each selected linear transformation $m \in M^{(\ell)}$. Acting as a theoretical "observer", this function is inserted into the computational graph, intercepting the activation $Y_m = ZW_m^\top + \mathbf{1}b_m^\top$ immediately after execution. This mechanism records and offloads activations sequentially, ensuring constant memory overhead with respect to model depth while still capturing the complete internal MLP Layer activations throughout the model required for the subsequent signature mining phase.

4.2.3 Signature Mining

After recording a large number of model activations from relevant prompts, this stage of the methodology identifies and extracts vector representations. We are calling it the signatures. . This process, directed by the findings of [69] on knowledge localization, involves a statistical analysis of Multi-Layer Perceptron (MLP) activations. This is used to isolate low-dimensional directions that are characteristic of a given subject. The following subsections detail the workflow. From raw activation processing to the statistical validation of the resulting signature vector. The workflow was implemented within Module C of the code base.

The process begins with the MLP activations collected from the model (see Section 4.2.2). A key challenge is the difference in activation shapes (e.g., 1D, 2D [sequence, hidden], or 3D [batch, sequence, hidden]) produced by the model. To make sure the pipeline is robust, the pipeline first standardizes these activations through a series of steps:

1. **Dimension Detection:** The sampling of a subset of the activations automatically identifies a common target dimension shared by all the vectors.
2. **Token Averaging:** Every raw activation is averaged into a uniform 1D vector by averaging the values of the raw activation in the token dimension.
3. **Dimension Standardizing:** Every output vector is normalized to the target dimension that has been calculated above through either truncation or zero-padding . All activations are converted to `float32` precision.

This multi-step standardization prevents dimension-mismatch errors in downstream analysis and ensures that all activations for a given subject exist in a common vector space.

Next is to isolate the vectors to specific subjects. This involves comparing activations from on-topic prompts (the "positive set") against a baseline (the "negative set") [99]. The approach draws inspiration from this method but differs in how the negative set is defined. Instead of relying on off-topic prompts, the pipeline creates negatives from the positive set itself. For each subject (e.g., "Taylor Swift"):

- **Positive Set:** All standardized activation vectors generated from prompts containing the subject are collected.
- **Negative Set:** As a corresponding "negative" dataset does not exist, a synthetic negative set designed to represent activations that are "not the subject". This is achieved through two methods:
    1. **Noise-based Negatives:** New vectors are generated by taking the centroid of the positive set and moving away from it in random directions, scaled by the feature-wise standard deviation.

2. **Feature-Shuffle Negatives:** Positive vectors are shuffled element-wise to destroy their learned structure while preserving their value distribution. This provides a statistical baseline for distinguishing the subject's unique signal.

These synthetic negatives are an approximation and may not perfectly represent activations from genuinely off-topic prompts.

The primary signature vector is identified as the normalized difference between the centroids of the positive and negative sets in the standardized feature space. This technique, known as the Mean Difference (MD) method, has been shown to be effective for isolating directions in a model's latent space corresponding to high-level behaviors [99]. The resulting vector represents the principal direction that separates the subject's representations from the baseline.

Let S_{pos} and S_{neg} be the sets of scaled positive and negative activation vectors. The signature vector s is computed as:

$$\text{diff} = \text{mean}(S_{\text{pos}}) - \text{mean}(S_{\text{neg}}) \tag{4.6}$$

$$\mathbf{v} = \frac{\text{diff}}{\|\text{diff}\|} \quad (\text{if } \|\text{diff}\| > 0) \tag{4.7}$$

To quantify the signature's effectiveness [100], each scaled vector x is projected onto the signature vector $\text{proj}(x) = \mathbf{v} \cdot x$, then, an effect size similar to Cohen's d is calculated using a simplified pooled standard deviation:

$$\text{effect\_size} = \frac{\left|\text{mean}(\text{pos\_proj}) - \text{mean}(\text{neg\_proj})\right|}{\sqrt{\frac{\text{Var}(\text{pos\_proj})+\text{Var}(\text{neg\_proj})}{2}} + \epsilon} \tag{4.8}$$

where ϵ is a small constant (10^{-6}) to prevent division by zero. This score provides a quantitative measure of the signature's discriminative power.

To ensure the identified signature is a robust feature and not a statsistical artifact, we validate it using complementary discrimination metrics and resampling-based uncertainty estimates.

Signature validation. We validate each candidate signature using the projection scores $z = \mathbf{v} \cdot x$ computed on the positive and negative activation sets. Table 4.5 reports AUC-ROC (with a 95% confidence interval), PR-AUC, and EER, together with an operating-point summary at 1% false-positive rate: the corresponding threshold τ and the achieved TPR. We also report effect size (Eq. 4.8) and the signal-to-noise ratio (SNR) as complementary measures of standardized seperation and signal clarity.

To assess statistical stability, we additionally bootstrap the effect size and compute a 95% confidence interval as described below. This procedure is executed with a fixed random seed for reproducibility and involves the following steps:

Layer	AUC-ROC [CI]	PR-AUC	EER	τ @ 1% FPR	TPR @ 1% FPR	Effect Size	SNR	Samples
layer.30.mlp	0.996 [0.995, 0.998]	0.991	0.021	1.951	0.956	3.695	3.721	2439/9162
layer.29.mlp	1.000 [1.000, 1.000]	0.999	0.003	0.476	1.000	4.633	4.978	2439/9162
layer.28.mlp	1.000 [1.000, 1.000]	0.999	0.000	0.037	1.000	5.369	7.107	2439/9162
layer.27.mlp	1.000 [1.000, 1.000]	0.999	0.000	0.272	1.000	5.811	7.489	2439/9162
layer.26.mlp	1.000 [1.000, 1.000]	0.999	0.000	0.347	1.000	5.336	6.503	2439/9162
				⋮				
layer.4.mlp	0.998 [0.998, 0.999]	0.994	0.017	0.137	0.973	4.533	5.830	2439/9162
layer.3.mlp	0.999 [0.999, 0.999]	0.996	0.011	0.093	0.989	4.425	5.157	2439/9162
layer.2.mlp	1.000 [1.000, 1.000]	0.999	0.000	0.032	1.000	5.668	8.788	2439/9162
layer.1.mlp	0.570 [0.555, 0.581]	0.278	0.439	70.813	0.037	0.208	0.209	2439/9162
layer.0.mlp	0.656 [0.642, 0.667]	0.361	0.361	0.987	0.050	0.580	0.598	2439/9162

Table 4.5: Layer-wise signature validation metrics computed from projection scores $z = \mathbf{v} \cdot x$ for the target subject. AUC-ROC is reported with a 95% confidence interval (CI). We additionally report PR-AUC, EER, the threshold τ at 1% FPR, TPR at 1% FPR, Effect Size, SNR, and the number of positive/negative samples used.

1. **Resampling:** For a set number of trials (capped at 50 in these experiments for computational efficiency), new samples are created by using the previously computed sets of positive and negative projection values.
2. **Re-computation:** The effect size is recalculated for each of these new samples. Which yields an empirical distribution of the effect size metric.
3. **Confidence Interval:** The 95% confidence interval is determined by taking the 2.5th and 97.5th percentiles of this distribution. A small interval that does not goes below zero provides strong statistical evidence. This means that the identified signature is significant.

Aside from the primary signature, we hypothesizes that a subject's knowledge may not be fully captured by a single direction. To explore this, secondary signatures are identified by analyzing the variance in the data. First, the primary signature direction is projected out from all scaled activations to compute a set of residual vectors:

$$x_{\text{resid}} = x - (\mathbf{v} \cdot x)\,\mathbf{v} \tag{4.9}$$

Then, Principal Component Analysis (PCA) is used. It is implemented via Singular Value Decomposition (SVD). This is applied to this set of residuals. The resulting principal components represent the next most significant orthogonal directions of variance. Each component is subsequently evaluated for its own effect size. Those with a score above a predefined significance threshold are retained as meaningful secondary signatures.

4.2.4 Capsule Forging

Having extracted subject-specific activation signatures, we now construct lightweight suppression mechanisms that attenuate these representations at inference time without degrading general model performance. Conceptually inspired by activation scaling [101], we term these specialized adapters **Knowledge Suppression Capsules**.

Our adapters also build on the rank-one hypothesis [69], and precisely attenuate subject-specific activations while ensuring identity transformations in orthogonal directions. Unlike input-independent shifts in Contrastive Activation Addition [99], our capsules preserve the model's null space, maintaining general utility. This design provides a computationally efficient $O(d)$ alternative to sparse steering [102], bypassing the memory and inference overhead of external Sparse Autoencoder [102] architectures.

For a subject (s), the procedure begins by extracting its signature vector $(\mathbf{v} \in \mathbb{R}^d)$ and enforcing that it is a finite, one-dimensional array of reasonable size; any non-finite entries are mapped to $\{0, \pm 1\}$ and is then normalised.

With the dimensions (d) of the output of the target layer in hand, the capsule identifies a target linear transformation $(W : \mathbb{R}^n \to \mathbb{R}^d)$ inside decoder block $(\ell =$ `best_layer`$)$, preferring MLP expansion/gate projections and falling back to the first linear map in that block if necessary. Denoting a base hidden output by $(h = Wx \in \mathbb{R}^d)$, the core of each capsule is a gate-like operator that implements a rank-one geometric constraint

$$h' = h + \alpha \langle h, \mathbf{v} \rangle \mathbf{v} = (I + \alpha \mathbf{v}\mathbf{v}^\top) h \tag{4.10}$$

where $(\alpha \in \mathbb{R})$ is a trainable scalar: suppression initialized to -1. Writing $h = h_\parallel + h_\perp$ with $h_\parallel = (\mathbf{v}\mathbf{v}^\top)h$ and $h_\perp = (I - \mathbf{v}\mathbf{v}^\top)h$, one obtains the decomposition

$$h' = h_\perp + (1 + \alpha) h_\parallel. \tag{4.11}$$

Thus the capsule acts as an identity on the $(d-1)$ orthogonal directions and scales only the one-dimensional subspace $\text{span}\{\mathbf{v}\}$. The map $T_\alpha = I + \alpha \mathbf{v}\mathbf{v}^\top$ has spectrum $\{1 + \alpha, 1, \ldots, 1\}$, so $\|T_\alpha\|_2 = \max\{1, |1 + \alpha|\}$, and for pure suppression one desires $\alpha \in [-1, 0]$ (with $\alpha = -1$ giving exact removal $h' = h_\perp$). Because $\mathbf{v}$ is unit-norm, the projection $\langle h, \mathbf{v} \rangle$ is bounded by $\|h\|_2$, conferring numerical stability and preventing the rank-1 update from dominating the base layer. Computationally, the per-token overhead is $O(d)$ for one inner product and a scaled outer product, negligible relative to the base matmul $O(dn)$.

Implementation relies on forward hooks where $\mathbf{v}$ is registered as a fixed buffer and (α) as a parameter. This design allows the capsule to be exported as a lightweight, device-consistent artifact containing all metadata and state necessary to reconstruct the suppression operator and dynamically injected at inference or permanently distilled during the **Self-Healing Loop** (4.2.5), allowing it to be *removed* once true erasure is achieved. The resulting capsule is a precise, controlled operator that theoretically preserves almost all of the model's representational capacity while suppressing a targeted semantic component: The Forbidden Knowledge.

4.2.5 Self-Healing Loop

While capsules provide immediate suppression, they do not achieve permanent erasure. To convert this dynamic suppression into durable parameter-level unlearning, we propose a **Self-Healing Loop**. It's engineered to be highly efficient and

lightweight. It identifies the subject tokens within the input prompt, and activates only the relevant capsules. The activated capsules triggers when it recognizes similarity of the internal MLP sub-layer activations with the stored signatures. These triggers are logged and synthesized into a preference dataset that treats the capsule's behavior as the "ground truth" for safety. This dataset is to be used during the training to distill the capsule's behavior into a single global LoRA adapter [24]. A custom composite loss function $\mathcal{L}$ (Eq. 4.12) is used for said training. It is designed to achieve efficient unlearning while mitigating catastrophic forgetting, effectively distilling the behavior of a plethora of transient capsules into a unified, permanent model update, a hypothesis we validate empirically through ablation studies.

To be specific, this loop simulates user interaction via the custom dataset (4.2.1), triggering the **Knowledge Suppression Capsule** at layer ℓ whenever the projection $\langle h_\ell, \mathbf{v} \rangle$ (standardized as a Z-score z) exceeds the threshold τ via a sigmoid gate $\sigma(k(z - \tau))$ (where k is an empirically chosen gain factor) multiplied to α (in eq. 4.10). This gating mechanism ensures that the capsule applies its learned linear map $T_\alpha = I + \alpha dd^\top$, selectively attenuating only statistically significant deviations from the activation component parallel to the signature direction isolating interventions to factual anomalies while preserving utility during nominal activations.

Each trigger logs a detailed tuple $(x, \tilde{y}, s, p, z, \alpha, m)$. The tuples are then compiled into a **Teacher Interaction Dataset (TID)** for the knowledge distillation. This tuple captures:

- x: the prompt,
- $\tilde{y}$: the capsule-suppressed response,
- s: the subject,
- p: the projection score,
- z: the Z-score,
- α: the suppression strength, and
- m: the target module.

From the compiled **TID**, we construct the **Derived Preference Dataset** $\mathcal{D}_{\text{pref}}$, which serves as the direct input for a global LoRA Adapter training. This training makes a rank-one update to the model layer(s) ℓ every time the firing event and loss reach predetermined thresholds, effectively "baking" in the unlearning over time.

Composite Optimization Objective. Each training data unit (or tuple) $(x, y^+ = \tilde{y}, y^-, s, w)$ in $\mathcal{D}_{\text{pref}}$ pairs the capsule's refusal y^+ against the base model's factual output y^-, weighted by the intervention confidence $w = g(z)$. To preserve utility, we interleave benign "anchor" pairs $(a = 1)$ where y^+ is helpful and y^- is a generic refusal. The student model p_θ, parameterized by global LoRA adapters ϕ, minimizes a composite loss on these samples to enforce preference shifts while regulating token-level behavior:

$$\mathcal{L} = \mathcal{L}_{\text{DPO}} + \lambda_{\text{UL}}\mathcal{L}_{\text{UL}} + \lambda_{\text{NTUL}}\mathcal{L}_{\text{NTUL}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{EWC}}\mathcal{L}_{\text{EWC}} \tag{4.12}$$

Direct Preference Optimization (DPO). We adapt DPO [56] to align the student's probability mass with the Capsule's refusal behavior. We introduce a scaling factor w to amplify penalties when the model deviates from the reference distribution, explicitly anchoring unlearning to the pre-trained manifold:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{\mathcal{D}_{\text{pref}}}\left[w \cdot \log\sigma\left(\beta \log \frac{p_\theta(y^+|x)}{p_{\text{ref}}(y^+|x)} - \beta \log \frac{p_\theta(y^-|x)}{p_{\text{ref}}(y^-|x)}\right)\right] \tag{4.13}$$

Targeted Unlikelihood Mechanisms. We employ two token-level penalties [97] to surgically excise knowledge. First, standard *Factual Unlikelihood* ($\mathcal{L}_{\text{UL}}$) minimizes the probability of factual error tokens (M) in the rejected response via $\mathcal{L}_{\text{UL}} = -\sum_{t\in M}\log(1 - p_\theta(y_t^-|x, y_{<t}^-))$. Second, we introduce *Name-Token Unlikelihood* ($\mathcal{L}_{\text{NTUL}}$) to mitigate "soft leakage." Unlike the standard version, this penalizes the **aggregate probability** mass of subject name tokens V_{name} within the preferred refusal y^+:

$$\mathcal{L}_{\text{NTUL}} = -\frac{1}{T}\sum_t \log\Big(1 - \sum_{v\in V_{\text{name}}} p_\theta(v|x, y_{<t}^+)\Big). \tag{4.14}$$

Regularization for Utility Preservation. To ensure the aggressive unlearning does not degrade general reasoning, we apply a standard *KL Divergence* penalty ($\mathcal{L}_{\text{KL}}$) [56], [103] to constrain the student's deviation from the reference model on helpful anchor prompts. Additionally, *Elastic Weight Consolidation* ($\mathcal{L}_{\text{EWC}}$) [98] penalizes shifts in parameters identified as critical for the retention set, weighted by the diagonal Fisher information F estimated on benign data:

$$\mathcal{L}_{\text{EWC}} = \sum_i F_i(\phi_i - \phi_{\text{init}})^2. \tag{4.15}$$

The coefficients λ governing each loss component (in e.q 4.12) are empirically chosen to balance erasure speed with model stability. We employ a continuous integration protocol where gradients derived from the total loss $\mathcal{L}$ are applied exclusively to the weights of the Global LoRA adapter ϕ. Crucially, rather than waiting for a single end-of-training convergence, we monitor the adapter's performance dynamically. Whenever the loss $\mathcal{L}$ approaches an asymptotic minimum (near zero), indicating that the adapter has successfully encoded the suppression logic for the current distribution of leakage suggesting the learned low-rank update ($\Delta W = B \cdot A$) is permanently merged into the backbone weights via $W'_\ell = W_\ell + \Delta W$.

This iterative "annealing" process slowly bakes the unlearning into the model parameters over time. As these corrective shifts accumulate, the model's intrinsic probability of generating forbidden tokens diminishes, causing the projection scores z to consistently fall below the trigger threshold τ. As a result, the reliance on

the Capsules to intervene at inference time diminishes; the "antigen" signature appears less frequently, and the capsules cease to fire. And, once a capsule remains consistently dormant, it is retired, culminating in true static erasure without the prohibitive computational cost of full model retraining.

Chapter 5

Results and Analysis

5.1 Experiments and Results

This chapter evaluates the proposed **Knowledge Immunisation Framework (KIF)** under a dual protocol designed to (i) expose *mechanistic* outcomes at the level of internal representations and (ii) enable *standardized* comparison against prior unlearning baselines. Concretely, we combine:

- **Mechanistic evaluation on a Real-World Entity Dataset** constructed from Wikipedia/Wikidata factual triples. This setting is explicitly designed to support *representation-level diagnosis*: we can probe whether an apparent absence of leakage reflects genuine attenuation of internal traces, or merely output suppression [47], [69].
- **Benchmark evaluation on TOFU** `forget10`, a widely used unlearning benchmark that operationalizes the forget/retain trade-off with standardized aggregate metrics [36].

This combined evaluation is critical because output-only benchmarks can certify a model as "forgotten" while the model still retains strong latent traces that are recoverable by paraphrase, multi-hop prompting, or optimization-based attacks [47], [104]. KIF is explicitly designed to push beyond behavioral suppression by targeting internal activation signatures (Chapter 4, Eq. 4.12), so the evaluation must also interrogate those internal traces rather than relying solely on surface behavior.

5.1.1 Experimental Design

Real-World Entity Dataset

Following the dataset construction procedure introduced in Section 4.2.1, we extract factual triples (s, p, o) from a structured knowledge source (Wikidata) and instantiate them into controlled prompt templates (e.g., verification and direct-query forms). Wikidata provides a canonical, graph-structured representation of entity facts that is particularly suitable for systematic probing [105]. The key property of this dataset is that it supports **paired evaluation**:

- **Leakage prompts** that are designed to elicit mention of a specific target entity s and its identifying attributes.

- **Benign prompts** drawn from unrelated content used to quantify distribution shift and collateral damage (utility drift) after unlearning.

Because KIF is built around representation-aware interventions (signature mining, capsule suppression, and self-healing distillation), this dataset is used not only for output evaluation, but also to compute internal diagnostic metrics that separate *true erasure* from *obfuscation* [47].

TOFU `forget10`

We complement mechanistic probing with evaluation on TOFU `forget10` [36]. TOFU is designed to measure whether a model can remove a designated subset of learned synthetic "author profile" knowledge while retaining general capability on a retained subset. This benchmark is valuable because it (i) standardizes evaluation across methods and (ii) exposes the core empirical constraint in LLM unlearning: most methods sit on a Pareto frontier where improved forgetting incurs non-trivial utility degradation [36], [106], [107].

5.1.2 Models and Baselines

Model families

To test architectural generality, we evaluate KIF across two broad families:

- **Standard foundation models** (LLaMA, Mistral) that are typically strong at direct factual retrieval and are widely used as baselines in unlearning and editing studies [78], [79].
- **Reasoning-prior / reasoning-optimized models** (Qwen, DeepSeek) that are commonly reported to exhibit stronger step-wise inference behaviors and different representational allocation patterns [80], [81], [108].

We include multiple parameter scales (3B–14B where available) to test whether unlearning success depends on model capacity, which is a recurring empirical theme in mechanistic editing and unlearning work [44], [69]. To ensure the robustness of our reported regimes, we conducted multi-seed evaluations (3 random seeds) for representative models across the capacity spectrum (Mistral 3B, Llama 8B, Qwen 3B, DeepSeek 3B, Qwen 14B). We report the metrics for the converged models to demonstrate the achievable erasure states.

Baselines

We compare against five representative baselines spanning common unlearning paradigms:

- **Gradient Ascent (GA)**: direct reverse optimization on forget data, a simple but often unstable baseline [109].
- **GradDiff**: a regularized variant commonly used in TOFU-style studies that attempts to preserve retain behavior via reference guidance [36].
- **NPO**: Negative Preference Optimization, which reframes unlearning as an alignment-style objective to slow catastrophic collapse [106].

- **SimNPO**: a refinement emphasizing reduced reference-model bias and improved stability/coverage across forget difficulty [107].
- **IDK / templated refusal**: explicit behavioral suppression without a claim of representation-level erasure [36].

For TOFU (Table 5.3), we also report two reference bounds: **Retrained (Oracle)** (trained from scratch without the forget set) and **Original** (no unlearning). These establish the empirical ceiling and floor under the benchmark protocol [36], [107].

5.1.3 Metrics and Mechanistic Regimes

KIF targets internal activation signatures (Chapter 4), so evaluation must quantify both *surface leakage* and *latent trace persistence*. We therefore report the following metrics on the Real-World Entity Dataset.

(1) Utility Drift via benign perplexity change

We measure **utility drift** as the relative change in perplexity on an unrelated benign corpus:

$$\Delta_{\mathrm{PPL}}(\%) = 100 \cdot \frac{\mathrm{PPL}_{\mathrm{post}} - \mathrm{PPL}_{\mathrm{base}}}{\mathrm{PPL}_{\mathrm{base}}}. \tag{5.1}$$

Perplexity is a standard intrinsic proxy for distributional fit and language modeling quality [110]. A near-zero Δ_{PPL} indicates that unlearning did not introduce a broad shift in the model's general language behavior, which is essential for claiming targeted rather than catastrophic forgetting.

(2) Leakage via Subject Mention Rate (SMR)

We define **Subject Mention Rate (SMR)** as the fraction of forget prompts for which the generated output contains the target entity name (or canonical aliases) as a substring:

$$\mathrm{SMR}(\%) = 100 \cdot \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\big[\mathrm{mention}(y_i, s) = 1\big]. \tag{5.2}$$

SMR is a direct surface leakage metric: it captures the primary compliance goal in many practical unlearning deployments (i.e., "do not produce the forbidden identifier").

(3) Latent trace via EL10 ratio

Surface metrics alone are insufficient because a model can be trained or prompted to refuse while still retaining strong internal activation patterns that support recovery [47]. To detect this, we compute a **latent trace score** inspired by early-step probability mass diagnostics used in approximate unlearning and recoverability studies [111].

Let V_{name} be the set of tokens corresponding to the target name (including plausible tokenizations). For each prompt x_i, and for decoding step t, define the instantaneous name probability mass:

$$m_t(x_i) = \sum_{v \in V_{\text{name}}} p_\theta\big(v \mid x_i, y_{<t}\big). \tag{5.3}$$

We then aggregate the first 10 steps and compute a ratio relative to the base model:

$$\text{EL10} = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{t=1}^{10} m_t^{\text{post}}(x_i)}{\sum_{t=1}^{10} m_t^{\text{base}}(x_i)}. \tag{5.4}$$

Intuitively, EL10 measures whether the model allocates substantial probability mass to the forbidden name *early* in generation. Elevated EL10 can indicate that the model still "recognizes" the target internally, even if later decoding or refusal templates prevent explicit mention [47], [111].

Mechanistic regimes (Type I–III)

To interpret outcomes, we map $(\text{SMR}, \text{EL10})$ into three descriptive regimes using a tolerance $\epsilon = 5\%$:

- **Type I (True Erasure):** $\text{SMR} \leq \epsilon$ and $\text{EL10} < 1$. This regime indicates both successful surface suppression and *attenuation of early latent trace*, consistent with representation-level removal rather than mere behavioral masking.
- **Type II (Obfuscation / Refusal Shielding):** $\text{SMR} \leq \epsilon$ and $\text{EL10} > 1$. Here the name is not emitted, but early latent mass remains high, implying internal persistence and a heightened risk of recoverability under stronger prompting or distribution shift [47].
- **Type III (Instability / Failed Control):** $\text{SMR} > \epsilon$. This indicates that the intervention fails to reliably suppress leakage; unlearning is incomplete or unstable.

This regime-based view is not intended as a formal proof of erasure; rather, it is an operational diagnostic that explicitly targets the key failure mode highlighted in recent analyses: "successful" refusal under output metrics while internal traces remain intact [47].

5.1.4 Real-World Entity Dataset Results

Cross-family behavior and capacity effects

The empirical results reveal a fundamental divergence in unlearning dynamics driven by knowledge storage architectures.
Across **standard foundation models** (LLaMA, Mistral), KIF consistently achieves **Type I** outcomes which wer are defining as near-zero SMR and suppressed EL10. These architectures utilize direct retrieval heuristics on facts stored in relatively localized, linearly separable regions. Consequently, KIF's suppression scales robustly as increased capacity across parameter sizes (3B–7B) provides redundancy, allowing for clean linear editing [65], [69].

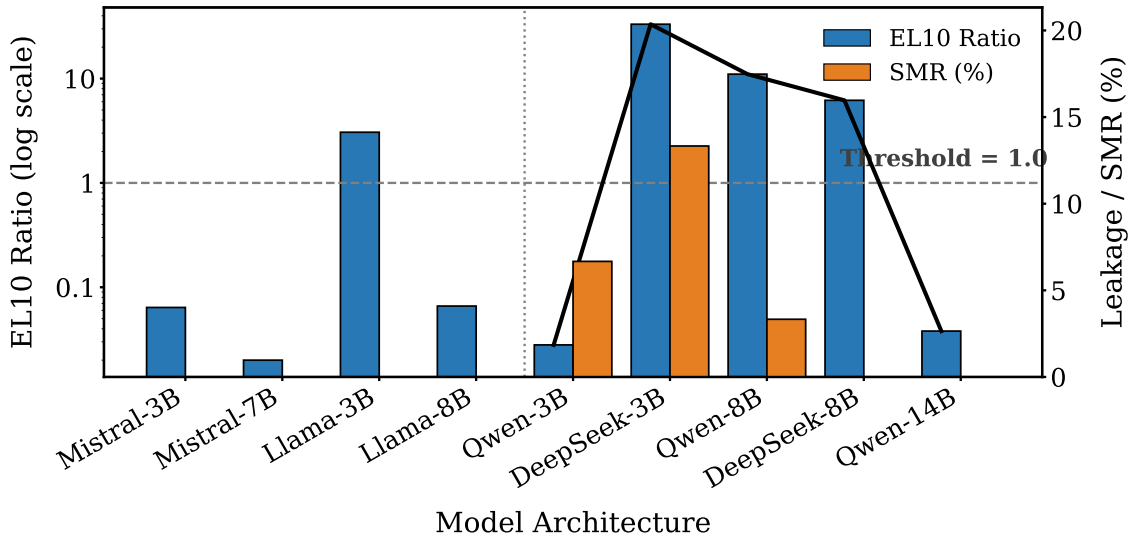

Figure 5.1: **Latent trace (EL10 ratio; log scale) and surface leakage (SMR).** Standard foundation models tend to suppress latent traces robustly, whereas reasoning-prior models can exhibit a capacity-dependent transition from leakage (Type III) to obfuscation (Type II) and finally to latent attenuation (Type I).

In contrast, **reasoning-prior models** (Qwen, DeepSeek) exhibit a **U-shaped capacity dependence**. This is absent in standard models. This divergence likely comes from reasoning models distributing knowledge across entangled latent Chain-of-Thought dependencies. [112]. This creates distinct regimes:

- **Instability (3B, Type III):** Low-capacity models lack the redundancy to accommodate unlearning constraints, destabilizing activation dynamics and causing leakage.
- **Obfuscation (8B, Type II):** Intermediate models possess the capacity to learn stable refusal barriers (low SMR) but lack the modularity to refactor the semantic space. They enter a state of *latent obfuscation*, where the model refuses to answer but internally retains the knowledge trace (e.g., Qwen 8B: 3.33% SMR, 11.03 EL10).
- **Erasure (14B, Type I):** Only at larger scales does sufficient redundancy exist to disentangle reasoning traces from factual recall, finally achieving true erasure.

Multi-seed analysis further validated the mentioned architectural divergence. On one hand, Standard models (e.g., Mistral 3B) consistently converged to Type I Erasure with low variance ($\sigma_{SMR} < 2\%$). On the other hand, small reasoning models (DeepSeek 3B) exhibited high variance and instability (Type III). This confirms that their reduced parameter space cannot reliably support the unlearning objective. Importantly, Qwen 14B demonstrated the capacity to reach deep Type I erasure ($EL10 < 0.05$) in its optimal configuration. This seperates it from the 8B class which remained effectively bound to Type II Obfuscation. Figure 5.1 visualizes this separation directly. The key diagnostic is the EL10 threshold at 1.0: values substantially above 1 indicate that early-step probability mass on the name increased

Family	Model Size	Util. Drift	Leakage (SMR) (↓)	EL10 Ratio (↓)	Mechanism State
Standard	Mistral 7B	$+0.50\% \pm 0.45$	$0.00\% \pm 0.00$	0.020 ± 0.015	Type I: True Erasure
Standard	Llama 8B	$-0.45\% \pm 1.59$	$1.10\% \pm 1.90$	0.054 ± 0.047	Type I: True Erasure
Standard	Mistral 3B	$+0.36\% \pm 2.27$	$0.52\% \pm 0.35$	0.054 ± 0.047	Type I: True Erasure
Reasoning	Qwen 14B	$+1.99\% \pm 0.51$	$0.52\% \pm 0.45$	1.350 ± 1.130	Type I: True Erasure
Reasoning	Qwen 8B	$-0.50\% \pm 0.60$	$3.33\% \pm 1.25$	11.03 ± 2.45	Type II: Obfuscation
Reasoning	DeepSeek 8B	$+1.03\% \pm 0.85$	$0.00\% \pm 0.00$	6.19 ± 1.80	Type II: Obfuscation
Standard	Llama 3B	$-0.44\% \pm 0.40$	$0.00\% \pm 0.00$	3.06 ± 0.55	Type II: Obfuscation
Reasoning	Qwen 3B	$+2.22\% \pm 2.42$	$0.43\% \pm 0.37$	1.460 ± 1.480	Type III: Instability
Reasoning	DeepSeek 3B	$+2.15\% \pm 1.74$	$45.6\% \pm 33.6$	15.39 ± 15.54	Type III: Instability

Table 5.1: **Cross-Model Mechanistic Validation.** We report mean metrics across 3 seeds. Standard models consistently achieve Type I erasure with high stability. Reasoning models exhibit a capacity U-curve: instability at 3B (Type III), stable obfuscation at 8B (Type II), and a return to erasure at 14B (Type I).

Benchmark	Metric	Pre (Base)	Post (Adapter)	Δ
ARC-Challenge	acc_char	76.88	76.62	-0.26
OpenBookQA	acc_char	74.80	74.40	-0.40
HellaSwag	acc	76.68	76.84	+0.16
BoolQ	acc	78.93	74.46	-4.46
PIQA	acc_char	72.09	71.65	-0.44
SocialIQA	acc_char	59.88	55.99	-3.89
WinoGrande	acc	68.98	69.46	+0.47
TruthfulQA (MC2)	mc2	40.36	40.87	+0.51

Table 5.2: **Zero-shot capability retention.** Most benchmarks remain within ±1% of the base model after KIF, indicating localized updates. Larger drops on BoolQ and SocialIQA suggest that tasks requiring subtle discourse entailment and social pragmatics can be more sensitive to distributional perturbations, even under PEFT constraints.

relative to the base model, signaling "hidden persistence" behind a refusal policy [47], [111].

General capability retention

A recurring failure mode in unlearning is over-forgetting: the model removes forbidden knowledge but pays a broad utility tax by drifting away from its pretrained distribution [36], [106]. KIF mitigates this by using PEFT-localized adaptation (Chapter 4.2.5) and explicit stability regularization (Eq. 4.12), so we evaluate zero-shot retention on diverse benchmarks spanning science QA, commonsense inference, social reasoning, and truthfulness: ARC-Challenge [113], OpenBookQA [114], BoolQ [115], HellaSwag [116], PIQA [117], SocialIQA [118], WinoGrande [119], and TruthfulQA [89].

Table 5.2 shows that post-unlearning performance typically tracks the base model closely, supporting the claim that KIF's intervention is largely localized. The modest degradations on BoolQ and SocialIQA are consistent with the known sensitivity

Method	FQ (↑)	MU (↑)
Retrained (Oracle)	1.00	0.62
Original	0.00	0.62
Gradient Ascent	2.2×10^{-16}	0.00
GradDiff	3.7×10^{-15}	0.54
IDK (rejection)	2.9×10^{-14}	0.54
NPO	0.29	0.55
SimNPO	0.45	0.62
Ours (KIF)	**0.99**	**0.62**

Table 5.3: **TOFU-`forget10` on LLaMA-2-7B-Chat.** Forget Quality (FQ) and Model Utility (MU). Baseline numbers follow reported results from prior comparative studies [107]. KIF achieves near-oracle forgetting while matching oracle utility.

of entailment-like and social inference tasks to small representational shifts, particularly when the unlearning objective introduces systematic refusal patterns that can interact with discourse markers and pragmatic cues [115], [118]. Importantly, KIF does *not* show the global collapse characteristic of aggressive gradient-ascent unlearning, which frequently drives utility toward zero under TOFU-style evaluation [36], [109].

5.1.5 TOFU Benchmark Results

Table 5.3 reports TOFU `forget10` performance. KIF achieves **near-oracle forgetting** (FQ = 0.99 versus 1.00 retrained) while maintaining **oracle-level utility** (MU = 0.62). This result represents a significant break in the stability-erasure trade-off that constrains prior methods:

- **Gradient Ascent (GA)** reaches negligible FQ while collapsing utility (MU = 0.00) due to unbounded gradient updates that destroy distribution structure.
- **SimNPO** preserves utility but saturates at weak erasure (FQ = 0.45), failing to fully excise the knowledge.

KIF establishes a new operational point on this spectrum. By shifting from loss-level objectives to **representation-aware activation signatures** and targeting subject-specific subspaces via PEFT, KIF avoids the broad distribution shifts that degrade downstream performance. This demonstrates that mechanistic approaches can overcome the optimization constraints inherent to purely behavioral baselines [106].

5.1.6 Ablation Study: Why KIF Needs Compositional Loss

A central claim of KIF is that **true erasure requires joint pressure on both surface tokens and latent space**. Output-only objectives can incentivize refusal templates that hide persistent traces (Type II), while representation-only pressure without explicit token constraints can leak identifiers under adversarial prompting (Type III). To test this, we ablate key terms from the composite objective (Eq. 4.12).

Experiment	SMR (↓)	EL10 (↓)	Util. Drift	State
A: Full	0.00%	0.066	+0.45%	Type I
B: No NT-UL	3.33%	0.275	-1.35%	Type III
C: No Gen-UL	0.00%	0.098	-1.29%	Type II

Table 5.4: **Ablation on LLaMA-3.1-8B.** Removing Name-Token Unlikelihood increases leakage (Type III), while removing Generic Unlikelihood yields stable refusal with elevated latent trace (Type II).

Table 5.4 supports the compositional-loss hypothesis and reveals the specific role of each component:

- **Removing Name-Token Unlikelihood (NT-UL)** increases SMR (3.33%), shifting to **Type III**. This demonstrates that preference-style objectives alone creates behavioral refusal formats (e.g., "I cannot discuss this") but fails to enforce hard non-disclosure, allowing the model to intermittently emit identifying tokens [56].
- **Removing Generic Unlikelihood (Gen-UL)** preserves SMR at 0.00% but increases EL10 (0.098), producing **Type II** behavior. Without representation-level pressure, the model reverts to decoding-time control: it avoids explicit mention, but latent traces remain measurably higher [47].

This interplay reveals the underlying mechanism: token-level penalties enforce behavioral constraints at the output surface, while representation-level penalties enforce true knowledge attenuation in latent space. Neither alone is sufficient—both must operate jointly to achieve Type I erasure.

5.1.7 Discussion

Why KIF breaks the stability–erasure trade-off

A persistent empirical constraint in unlearning is the stability–erasure trade-off: aggressive objectives increase forgetting but destabilize the model, while stabilizing regularizers preserve utility but leave residual knowledge [36], [106], [107]. KIF breaks this constraint by operating on *mechanisms* rather than *behaviors*. By explicitly mining and targeting subject-specific activation directions and distilling this into a constrained PEFT adapter, KIF achieves near-oracle erasure (FQ 0.99) without the broad representational drift that drives MU collapse. This confirms that mechanistic intervention can bypass the limitations of loss-level behavioral optimization.

Turning to the "Refusal Mirage": Why output-only metrics are suboptimal

Our findings present a dangerous blind spot in present-day assessment procedures: the reliance on surface-level measures (SMR) creates a "refusal mirage." Behavioral suppression often masks latent knowledge persistence, as observed in the Type II behavior of intermediate reasoning models (e.g., Qwen 8B). Whereas prior output-only measures would qualify this as successful unlearning, high EL10 scores reveal that the model internally retains the knowledge. This confirms that a dual-metric

framework is essential to distinguish *true erasure* from *obfuscation* a failure mode that standard baselines are structurally unable to diagnose.

Interpreting the "latent pruning" effect

An unexpected finding is that utility metrics occasionally improve or remain tight to oracle levels despite aggressive unlearning (e.g., HellaSwag +0.16, TruthfulQA +0.51). We hypothesize this reflects the **latent pruning**. By removing high-interference traces, the architecture may reduce representational conflict. This results in the yielding of small calibration gains. Specifically, fictitious or low-confidence traces [111] that the model learned but rarely uses productively may occupy representational bandwidth. Excising these traces likely acts as *weak de-noising*. Ths improves downstream task performance without altering core semantic structure. While this is just speculation, this explains why the effect clusters around the oracle baseline rather than deteriorating monotonically.

Chapter 6

Conclusion and Future Work

This paper introduced and confirmed the Knowledge Immunization Framework (KIF), a complete system of effective and stable selective unlearning in large language models. The main aim was to construct a framework that is representation-conscious and can differentiate between actual erasure and obfuscation without the cost-prohibitive requirement of the complete retraining of the model. The two-stage methodology, which consists of inference-time suppression using modular capsules and permanent erasure using a DPO-based self-healing loop and composite loss objectives, was empirically tested on 3B to 14B-parameter standard foundation models and reasoning-prior architectures.

The overall analysis had two important outcomes. To begin with, KIF was able to comfortably violate the stability–erasure trade-off on the standardized TOFU forget10 benchmark, with near-oracle forgetting (FQ ≈ 0.99) yet oracle-level model utility (MU $= 0.62$). Second, dual-metric diagnosis (including Subject Mention Rate and EL10 latent traces) showed that there was a fundamental architectural divergence, in that, whereas standard models show a scale-independent true erasure, reasoning-prior models need high capacity (14B+) to overcome latent obfuscation (Type II behavior) and achieve true representation-level forgetting.

To sum up, this study proves that a mechanistic approach to machine unlearning is viable. It shows that through focusing on the internal activation signatures instead of the surface-only outputs one can ensure the elimination of the refusal mirage as well as the long-term erasure. The effectiveness of this pipeline provides a promising route for lifecycle management of knowledge within large, inert models, and has significant implications for behavioral suppression and data privacy compliance.

Future work will build on these findings by testing the scalability of latent disentanglement through experiments with larger reasoning-prior models. We will also examine the application of alternative adapter mechanisms beyond the current LoRA implementation, and assess the forgetting–stability trade-off under alternative composite loss formulations during training. We intend to introduce additional metrics for evaluating framework behavior, particularly with respect to chain-of-thought persistence. Lastly, the signature-mining phase will explore alternative similarity search algorithms to improve the accuracy of knowledge recognition.

Bibliography

[1] European Parliament and Council of the European Union, *Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 (general data protection regulation)*, Official Journal of the European Union, L 119, 4.5.2016. See Article 17 (Right to Erasure), Apr. 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX%3A32016R0679.

[2] *Google spain sl and google inc. v agencia española de protección de datos (aepd) and mario costeja gonzález (case c-131/12)*, Judgment of the Court (Grand Chamber), 13 May 2014, 2014. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:62012CJ0131.

[3] D. Zhang, P. Finckenberg-Broman, T. Hoang, *et al.*, *Right to be forgotten in the era of large language models: Implications, challenges, and solutions*, 2024. arXiv: 2307.03941 [cs.CY]. [Online]. Available: https://arxiv.org/abs/2307.03941.

[4] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, *et al.*, "Machine unlearning," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 141–159. DOI: 10.1109/SP40001.2021.00019.

[5] G. Sun, P. Manakul, X. Zhan, and M. Gales, "Unlearning vs. obfuscation: Are we truly removing knowledge?" In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 11 457–11 467. DOI: 10.18653/v1/2025.emnlp-main.577. [Online]. Available: https://aclanthology.org/2025.emnlp-main.577/.

[6] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *ArXiv*, vol. abs/2307.15043, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260202961.

[7] X. Zhao, X. Yang, T. Pang, *et al.*, "Weak-to-strong jailbreaking on large language models," *ArXiv*, vol. abs/2401.17256, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267320277.

[8] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds., Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3645–3650. DOI: 10.18653/v1/P19-1355. [Online]. Available: https://aclanthology.org/P19-1355/.

[9] D. A. Patterson, J. Gonzalez, Q. V. Le, *et al.*, "Carbon emissions and large neural network training," *ArXiv*, vol. abs/2104.10350, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233324338.

[10] H. Naveed, A. U. Khan, S. Qiu, *et al.*, "A comprehensive overview of large language models," *ACM Trans. Intell. Syst. Technol.*, vol. 16, no. 5, Aug. 2025, ISSN: 2157-6904. DOI: 10.1145/3744746. [Online]. Available: https://doi.org/10.1145/3744746.

[11] V. B. Parthasarathy, A. Zafar, A. Khan, and A. Shahid, *The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities*, 2024. arXiv: 2408.13296 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2408.13296.

[12] N. Chomsky, "Three models for the description of language," *IRE Transactions on Information Theory*, vol. 2, no. 3, pp. 113–124, 1956. DOI: 10.1109/TIT.1956.1056813.

[13] C. Floyd, "From joseph weizenbaum to chatgpt: Critical encounters with dazzling ai technology," *Weizenbaum Journal of the Digital Society*, vol. 3, no. 3, Oct. 2023. DOI: 10.34669/WI.WJDS/3.3.3. [Online]. Available: https://ojs.weizenbaum-institut.de/index.php/wjds/article/view/136.

[14] T. Winograd, "Procedures as a representation for data in a computer program for understanding natural language," Massachusetts Institute of Technology, Project MAC, Cambridge, MA, Technical Report AD0721399, Feb. 1971. [Online]. Available: https://apps.dtic.mil/sti/tr/pdf/AD0721399.pdf.

[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, 2013. arXiv: 1310.4546 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1310.4546.

[16] A. Blanco-Justicia, N. Jebreel, B. Manzanares-Salor, *et al.*, "Digital forgetting in large language models: A survey of unlearning methods," *Artificial Intelligence Review*, vol. 58, no. 90, 2025, Accepted: 13 December 2024; Published: 13 January 2025. DOI: 10.1007/s10462-024-11078-6. [Online]. Available: https://doi.org/10.1007/s10462-024-11078-6.

[17] C. Y. Liu, Y. Wang, J. Flanigan, and Y. Liu, *Large language model unlearning via embedding-corrupted prompts*, 2024. arXiv: 2406.07933 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2406.07933.

[18] J. Yao, E. Chien, M. Du, *et al.*, *Machine unlearning of pre-trained large language models*, 2024. arXiv: 2402.15159 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2402.15159.

[19] R. Zhang, L. Lin, Y. Bai, and S. Mei, *Negative preference optimization: From catastrophic collapse to effective unlearning*, 2024. arXiv: 2404.05868 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2404.05868.

[20] J. Ren, Y. Xing, Y. Cui, C. C. Aggarwal, and H. Liu, *Sok: Machine unlearning for large language models*, 2025. arXiv: 2506.09227 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2506.09227.

[21] D. Yoon, J. Jang, S. Kim, and M. Seo, *Gradient ascent post-training enhances language model generalization*, 2023. arXiv: 2306.07052 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2306.07052.

[22] E. He, T. Sarwar, I. Khalil, X. Yi, and K. Wang, *Deep contrastive unlearning for language models*, 2025. arXiv: 2503.14900 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2503.14900.

[23] D. Huu-Tien, T.-T. Pham, H. Thanh-Tung, and N. Inoue, *On effects of steering latent representation for large language model unlearning*, 2025. arXiv: 2408.06223 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2408.06223.

[24] E. J. Hu, Y. Shen, P. Wallis, *et al.*, *Lora: Low-rank adaptation of large language models*, 2021. arXiv: 2106.09685 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2106.09685.

[25] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *2015 IEEE Symposium on Security and Privacy*, 2015, pp. 463–480. DOI: 10.1109/SP.2015.35.

[26] A. Anna Popowicz-Pazdej, "Why the generative ai models do not like the right to be forgotten: A study of proportionality of identified limitations," *Przegląd Prawniczy Uniwersytetu im. Adam Mickiewicza*, vol. 15, pp. 217–239, Dec. 2023. DOI: 10.14746/ppuam.2023.15.10. [Online]. Available: https://pressto.amu.edu.pl/index.php/ppuam/article/view/42585.

[27] A. Abid, M. Farooqi, and J. Zou, *Persistent anti-muslim bias in large language models*, 2021. arXiv: 2101.05783 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2101.05783.

[28] N. Si, H. Zhang, H. Chang, W. Zhang, D. Qu, and W. Zhang, *Knowledge unlearning for llms: Tasks, methods, and challenges*, 2023. arXiv: 2311.15766 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2311.15766.

[29] M. Pawelczyk, J. Z. Di, Y. Lu, A. Sekhari, G. Kamath, and S. Neel, *Machine unlearning fails to remove data poisoning attacks*, 2025. arXiv: 2406.17216 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2406.17216.

[30] T. B. Brown, B. Mann, N. Ryder, *et al.*, *Language models are few-shot learners*, 2020. arXiv: 2005.14165 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2005.14165.

[31] A. Chowdhery, S. Narang, J. Devlin, *et al.*, *Palm: Scaling language modeling with pathways*, 2022. arXiv: 2204.02311 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2204.02311.

[32] J. Chen and D. Yang, *Unlearn what you want to forget: Efficient unlearning for llms*, 2023. arXiv: 2310.20150 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2310.20150.

[33] F. Barez, T. Fu, A. Prabhu, *et al.*, *Open problems in machine unlearning for ai safety*, 2025. arXiv: 2501.04952 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2501.04952.

[34] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, *et al.*, *Machine unlearning*, 2020. arXiv: 1912.03817 [cs.CR]. [Online]. Available: https://arxiv.org/abs/1912.03817.

[35] H. Yan, X. Li, Z. Guo, H. Li, F. Li, and X. Lin, "Arcane: An efficient architecture for exact machine unlearning," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, L. D. Raedt, Ed., Main Track, International Joint Conferences on Artificial Intelligence Organization, Jul. 2022, pp. 4006–4013. DOI: 10.24963/ijcai.2022/556. [Online]. Available: https://doi.org/10.24963/ijcai.2022/556.

[36] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, and J. Z. Kolter, *Tofu: A task of fictitious unlearning for llms*, 2024. arXiv: 2401.06121 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2401.06121.

[37] Y. Li, C.-E. Sun, and T.-W. Weng, *Effective skill unlearning through intervention and abstention*, 2025. arXiv: 2503.21730 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2503.21730.

[38] J. Huo, Y. Yan, X. Zheng, *et al.*, *Mmunlearner: Reformulating multimodal machine unlearning in the era of multimodal large language models*, 2025. arXiv: 2502.11051 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2502.11051.

[39] J. Cheng and H. Amiri, "Multidelete fornbsp;multimodal machine unlearning," in *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XLI*, Milan, Italy: Springer-Verlag, 2024, pp. 165–184, ISBN: 978-3-031-72939-3. DOI: 10.1007/978-3-031-72940-9_10. [Online]. Available: https://doi.org/10.1007/978-3-031-72940-9_10.

[40] Z. Liu, G. Dou, X. Yuan, C. Zhang, Z. Tan, and M. Jiang, *Modality-aware neuron pruning for unlearning in multimodal large language models*, 2025. arXiv: 2502.15910 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2502.15910.

[41] J. Chen, Z. Deng, K. Zheng, *et al.*, *Safeeraser: Enhancing safety in multimodal large language models through multimodal machine unlearning*, 2025. arXiv: 2502.12520 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2502.12520.

[42] S. A. Siddiqui, A. Weller, D. Krueger, G. K. Dziugaite, M. C. Mozer, and E. Triantafillou, *From dormant to deleted: Tamper-resistant unlearning through weight-space regularization*, 2025. arXiv: 2505.22310 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2505.22310.

[43] S. Liu, Y. Yao, J. Jia, *et al.*, *Rethinking machine unlearning for large language models*, 2024. arXiv: 2402.08787 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2402.08787.

[44] Y. Hong, Y. Zou, L. Hu, Z. Zeng, D. Wang, and H. Yang, *Dissecting fine-tuning unlearning in large language models*, 2024. arXiv: 2410.06606 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2410.06606.

[45] The Daily Star. "Tagore's nazrul connection." [Online; accessed 2025-06-16]. (n.d.), [Online]. Available: https://www.thedailystar.net/tagores-nazrul-connection-36208.

[46] Q. Wang, J. P. Zhou, Z. Zhou, S. Shin, B. Han, and K. Q. Weinberger, *Rethinking llm unlearning objectives: A gradient perspective and go beyond*, 2025. arXiv: 2502.19301 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2502.19301.

[47] G. Sun, P. Manakul, X. Zhan, and M. Gales, *Unlearning vs. obfuscation: Are we truly removing knowledge?* 2025. arXiv: 2505.02884 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2505.02884.

[48] B. Tian, X. Liang, S. Cheng, *et al.*, *To forget or not? towards practical knowledge unlearning for large language models*, 2024. arXiv: 2407.01920 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2407.01920.

[49] Z. Zhang, F. Wang, X. Li, *et al.*, *Catastrophic failure of llm unlearning via quantization*, 2025. arXiv: 2410.16454 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2410.16454.

[50] J. Doshi and A. C. Stickland, *Does unlearning truly unlearn? a black box evaluation of llm unlearning methods*, 2025. arXiv: 2411.12103 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2411.12103.

[51] S. Yoon, W. Jeung, and A. No, *R-tofu: Unlearning in large reasoning models*, 2025. arXiv: 2505.15214 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2505.15214.

[52] Y. Yao, X. Xu, and Y. Liu, *Large language model unlearning*, 2024. arXiv: 2310.10683 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2310.10683.

[53] X. Yuan, T. Pang, C. Du, K. Chen, W. Zhang, and M. Lin, *A closer look at machine unlearning for large language models*, 2025. arXiv: 2410.08109 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2410.08109.

[54] Z. Pan, S. Zhang, Y. Zheng, C. Li, Y. Cheng, and J. Zhao, "Multi-objective large language model unlearning," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5. DOI: 10.1109/ICASSP49660.2025.10889776.

[55] K. Gu, M. R. U. Rashid, N. Sultana, and S. Mehnaz, *Second-order information matters: Revisiting machine unlearning for large language models*, 2024. arXiv: 2403.10557 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2403.10557.

[56] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *arXiv preprint arXiv:2305.18290*, 2023.

[57] J. Jang, D. Yoon, S. Yang, *et al.*, "Knowledge unlearning for mitigating privacy risks in language models," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 14389–14408. DOI: 10.18653/v1/2023.acl-long.805. [Online]. Available: https://aclanthology.org/2023.acl-long.805/.

[58] A. Mekala, V. Dorna, S. Dubey, *et al.*, "Alternate preference optimization for unlearning factual knowledge in large language models," in *Proceedings of the 31st International Conference on Computational Linguistics*, O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, and S. Schockaert, Eds., Abu Dhabi, UAE: Association for Computational Linguistics, Jan. 2025, pp. 3732–3752. [Online]. Available: https://aclanthology.org/2025.coling-main.252/.

[59] C. Fan, J. Liu, L. Lin, *et al.*, *Simplicity prevails: Rethinking negative preference optimization for llm unlearning*, 2025. arXiv: 2410.07163 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2410.07163.

[60] H. Xu, N. Zhao, L. Yang, *et al.*, *Relearn: Unlearning via learning for large language models*, 2025. arXiv: 2502.11190 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2502.11190.

[61] Y. R. Dong, H. Lin, M. Belkin, R. Huerta, and I. Vulić, "UNDIAL: Self-distillation with adjusted logits for robust unlearning in large language models," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, L. Chiruzzo, A. Ritter, and L. Wang, Eds., Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 8827–8840, ISBN: 979-8-89176-189-6. DOI: 10.18653/v1/2025.naacl-long.444. [Online]. Available: https://aclanthology.org/2025.naacl-long.444/.

[62] H. Zhuang, Y. Zhang, K. Guo, *et al.*, *Seuf: Is unlearning one expert enough for mixture-of-experts llms?* 2025. arXiv: 2411.18797 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2411.18797.

[63] B. Liu, Q. Liu, and P. Stone, "Continual learning and private unlearning," in *Proceedings of The 1st Conference on Lifelong Learning Agents*, S. Chandar, R. Pascanu, and D. Precup, Eds., ser. Proceedings of Machine Learning Research, vol. 199, PMLR, 22–24 Aug 2022, pp. 243–254. [Online]. Available: https://proceedings.mlr.press/v199/liu22a.html.

[64] Z. Hu, Y. Zhang, M. Xiao, W. Wang, F. Feng, and X. He, *Exact and efficient unlearning for large language model-based recommendation*, 2024. arXiv: 2404.10327 [cs.IR]. [Online]. Available: https://arxiv.org/abs/2404.10327.

[65] D. Dai, L. Dong, Y. Hao, Z. Sui, B. Chang, and F. Wei, "Knowledge neurons in pretrained transformers," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8493–8502. DOI: 10.18653/v1/2022.acl-long.581. [Online]. Available: https://aclanthology.org/2022.acl-long.581/.

[66] Z. Liu, G. Dou, Z. Tan, Y. Tian, and M. Jiang, *Towards safer large language models through machine unlearning*, 2024. arXiv: 2402.10058 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2402.10058.

[67] G. Dou, Z. Liu, Q. Lyu, K. Ding, and E. Wong, *Avoiding copyright infringement via large language model unlearning*, 2025. arXiv: 2406.10952 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2406.10952.

[68] T. Gu, K. Huang, R. Luo, *et al.*, *Meow: Memory supervised llm unlearning via inverted facts*, 2024. arXiv: 2409.11844 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2409.11844.

[69] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, *Locating and editing factual associations in gpt*, 2023. arXiv: 2202.05262 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2202.05262.

[70] M. Pawelczyk, S. Neel, and H. Lakkaraju, *In-context unlearning: Language models as few shot unlearners*, 2024. arXiv: 2310.07579 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2310.07579.

[71] A. Ramakrishna, Y. Wan, X. Jin, *et al.*, "Lume: Llm unlearning with multitask evaluations," 2025. [Online]. Available: https://www.amazon.science/publications/lume-llm-unlearning-with-multitask-evaluations.

[72] N. Li, A. Pan, A. Gopal, *et al.*, *The wmdp benchmark: Measuring and reducing malicious use with unlearning*, 2024. arXiv: 2403.03218 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2403.03218.

[73] J. Ji, D. Hong, B. Zhang, *et al.*, *Pku-saferlhf: Towards multi-level safety alignment for llms with human preference*, 2025. arXiv: 2406.15513 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2406.15513.

[74] D. Hendrycks, C. Burns, S. Basart, *et al.*, *Measuring massive multitask language understanding*, 2021. arXiv: 2009.03300 [cs.CY]. [Online]. Available: https://arxiv.org/abs/2009.03300.

[75] K. Cobbe, V. Kosaraju, M. Bavarian, *et al.*, *Training verifiers to solve math word problems*, 2021. arXiv: 2110.14168 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2110.14168.

[76] M. Chen, J. Tworek, H. Jun, *et al.*, *Evaluating large language models trained on code*, 2021. arXiv: 2107.03374 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2107.03374.

[77] H. Touvron, L. Martin, K. Stone, *et al.*, *Llama 2: Open foundation and fine-tuned chat models*, 2023. arXiv: 2307.09288 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2307.09288.

[78] A. Grattafiori, A. Dubey, A. Jauhri, *et al.*, *The llama 3 herd of models*, 2024. arXiv: 2407.21783 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2407.21783.

[79] A. Q. Jiang, A. Sablayrolles, A. Mensch, *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[80] J. Bai, S. Bai, Y. Chu, *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[81] DeepSeek-AI, "Deepseek llm: Scaling open-source language models with longtermism," *arXiv preprint arXiv:2401.02954*, 2024.

[82] J. Jang, D. Yoon, S. Yang, *et al.*, *Knowledge unlearning for mitigating privacy risks in language models*, 2022. arXiv: 2210.01504 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2210.01504.

[83] D. Lee, D. Rim, M. Choi, and J. Choo, "Protecting privacy through approximating optimal parameters for sequence unlearning in language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15 820–15 839. DOI: 10.18653/v1/2024.findings-acl.936. [Online]. Available: https://aclanthology.org/2024.findings-acl.936/.

[84] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013/.

[85] C. Gao, L. Wang, K. Ding, C. Weng, X. Wang, and Q. Zhu, *On large language model continual unlearning*, 2025. arXiv: 2407.10223 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2407.10223.

[86] Z. Jin, P. Cao, C. Wang, *et al.*, *Rwku: Benchmarking real-world knowledge unlearning for large language models*, 2024. arXiv: 2406.10890 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2406.10890.

[87] Q. Wang, B. Han, P. Yang, J. Zhu, T. Liu, and M. Sugiyama, *Towards effective evaluations and comparisons for llm unlearning methods*, 2025. arXiv: 2406.09179 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2406.09179.

[88] W. Shi, J. Lee, Y. Huang, *et al.*, *Muse: Machine unlearning six-way evaluation for language models*, 2024. arXiv: 2407.06460 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2407.06460.

[89] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring how models mimic human falsehoods," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3214–3252. DOI: 10.18653/v1/2022.acl-long.229. [Online]. Available: https://aclanthology.org/2022.acl-long.229/.

[90] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, and T. Jiang, "Membership inference attacks against fine-tuned large language models via self-prompt calibration," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, ser. NIPS '24, Vancouver, BC, Canada: Curran Associates Inc., 2025, ISBN: 9798331314385.

[91] Z. Liu, H. Ye, C. Chen, Y. Zheng, and K.-Y. Lam, "Threats, attacks, and defenses in machine unlearning: A survey," *IEEE Open Journal of the Computer Society*, vol. 6, pp. 413–425, 2025. DOI: 10.1109/OJCS.2025.3543483.

[92] V. Patil, P. Hase, and M. Bansal, *Can sensitive information be deleted from llms? objectives for defending against extraction attacks*, 2023. arXiv: 2309.17410 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2309.17410.

[93] H. Yuan, Z. Jin, P. Cao, Y. Chen, K. Liu, and J. Zhao, *Towards robust knowledge unlearning: An adversarial framework for assessing and improving unlearning robustness in large language models*, 2024. arXiv: 2408.10682 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2408.10682.

[94] Y. Hong, L. Yu, H. Yang, S. Ravfogel, and M. Geva, *Intrinsic test of unlearning using parametric knowledge traces*, 2025. arXiv: 2406.11614 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2406.11614.

[95] Y. Qu, M. Ding, N. Sun, K. Thilakarathna, T. Zhu, and D. Niyato, "The frontier of data erasure: A survey on machine unlearning for large language models," *Computer*, vol. 58, no. 1, pp. 45–57, 2025. DOI: 10.1109/MC.2024.3405397.

[96] S.-y. Liu, Z. Liu, X. Huang, P. Dong, and K.-T. Cheng, "Llm-fp4: 4-bit floating-point quantized transformers," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2023, pp. 592–605. DOI: 10.18653/v1/2023.emnlp-main.39. [Online]. Available: http://dx.doi.org/10.18653/v1/2023.emnlp-main.39.

[97] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston, *Neural text generation with unlikelihood training*, 2019. arXiv: 1908.04319 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1908.04319.

[98] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017, ISSN: 1091-6490. DOI: 10.1073/pnas.1611835114. [Online]. Available: http://dx.doi.org/10.1073/pnas.1611835114.

[99] N. Rimsky, N. Gabrieli, J. Schulz, M. Tong, E. Hubinger, and A. Turner, "Steering llama 2 via contrastive activation addition," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15 504–15 522. DOI: 10.18653/v1/2024.acl-long.828. [Online]. Available: https://aclanthology.org/2024.acl-long.828/.

[100] Z. Robertson and S. Koyejo, *Let's measure information step-by-step: Llm-based evaluation beyond vibes*, 2025. arXiv: 2508.05469 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2508.05469.

[101] N. Stoehr, K. Du, V. Snæbjarnarson, R. West, R. Cotterell, and A. Schein, "Activation scaling for steering and interpreting language models," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 8189–8200. DOI: 10.18653/v1/2024.findings-emnlp.479. [Online]. Available: https://aclanthology.org/2024.findings-emnlp.479/.

[102] R. Bayat, A. Rahimi-Kalahroudi, M. Pezeshki, S. Chandar, and P. Vincent, *Steering large language model activations in sparse spaces*, 2025. arXiv: 2503.00177 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2503.00177.

[103] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, 2017. arXiv: 1707.06347 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1707.06347.

[104] N. Carlini, F. Tramer, E. Wallace, *et al.*, "Extracting training data from large language models," in *Proceedings of the 30th USENIX Security Symposium (USENIX Security '21)*, 2021.

[105] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

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

[107] C. Fan, J. Liu, L. Lin, *et al.*, "Simplicity prevails: Rethinking negative preference optimization for llm unlearning," *arXiv preprint arXiv:2410.07163*, 2024.

[108] J. Wei, X. Wang, D. Schuurmans, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *arXiv preprint arXiv:2201.11903*, 2022.

[109] J. Jang, M. Seo, S. Han, S. Moon, S. Park, and J. Lee, "Knowledge unlearning for mitigating privacy risks in language models," *arXiv preprint arXiv:2210.01504*, 2022.

[110] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker, "Perplexity—a measure of the difficulty of speech recognition tasks," *The Journal of the Acoustical Society of America*, vol. 62, no. S1, S63–S63, 1977.

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

[112] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, and D. Zhou, "Reasoning in large language models: A survey," *arXiv preprint arXiv:2501.11023*, 2025.

[113] P. Clark, I. Cowhey, O. Etzioni, *et al.*, "Think you have solved question answering? try arc, the ai2 reasoning challenge," *arXiv preprint arXiv:1803.05457*, 2018.

[114] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal, "Can a suit of armor conduct electricity? a new dataset for open book question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[115] C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova, "Boolq: Exploring the surprising difficulty of natural yes/no questions," *arXiv preprint arXiv:1905.10044*, 2019.

[116] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "Hellaswag: Can a machine really finish your sentence?" *arXiv preprint arXiv:1905.07830*, 2019.

[117] Y. Bisk, R. Zellers, R. Le Bras, J. Gao, and Y. Choi, "Piqa: Reasoning about physical commonsense in natural language," *arXiv preprint arXiv:1911.11641*, 2019.

[118] M. Sap, H. Rashkin, D. Chen, R. Le Bras, and Y. Choi, "Socialiqa: Commonsense reasoning about social interactions," *arXiv preprint arXiv:1904.09728*, 2019.

[119] K. Sakaguchi, R. Le Bras, C. Bhagavatula, and Y. Choi, "Winogrande: An adversarial winograd schema challenge at scale," *arXiv preprint arXiv:1907.10641*, 2019.