Logits-Level Balanced Machine Unlearning for LLM-Based Recommendation System

Chenchen Tan, Xinghao Li, Youyang Qu, *Member, IEEE*, Cunjian Chen, *Senior Member, IEEE*, Shujie Cui, and Longxiang Gao, *Senior Member, IEEE*

***Abstract*—As recommendation systems have become foundational to digital platforms, the integration of large language models (LLMs) into these systems offers transformative potential. In LLM-based recommendation (LLMRec) systems, LLMs excel at capturing nuanced user pReferences, greatly enhancing personalization and recommendation accuracy. However, LLMRec systems face data governance issues, such as data privacy, outdated data, poisoned data, and copyrighted data. All these data government scenarios require the removal of data and its impact from large models like LLMRec. This poses significant challenges as existing solutions struggle to accurately delete target data from such complex systems. Motivated by this, we propose an LLMRec unlearning system based on adapter-driven logits modification in this work. This mechanism addresses the unique characteristics of LLMRec's complex parameters and network structure, enabling precise and effective unlearning while maintaining recommendation performance. Our approach offers several key advantages: 1) using adapters reduces training costs during the unlearning process; 2) a knowledge distillation (KD)-powered logits modification mechanism ensures that the model can still engage in effective reasoning after unlearning, leveraging its existing knowledge logic to maintain high-quality recommendations; and 3) an additional adapter-based general knowledge retention module is incorporated to preserve the core language capabilities of the LLMRec system, ensuring that its foundational performance remains intact. Extensive experimental results show the effectiveness and efficiency of the proposed system. The code is released at: https://anonymous.4open.science/r/llmrecunlearning-1CB3**

***Index Terms*—Adapter, knowledge distillation (KD), large language model (LLM), machine unlearning, recommendation system.**

I. Introduction

LARGE language models (LLMs) have driven innovations in various fields, including healthcare [1], finance [2], and e-commerce [3]. LLM-based recommendation (LLMRec) models, which are fine-tuned on pretrained LLMs with vast amounts of user data for personalized recommendations, have the potential to revolutionize personalized recommendations across rising industries like e-commerce, entertainment, and social media [4], [5]. Benefiting from the foundational language understanding and reasoning capabilities of LLMs, LLMRec can effectively capture intricate user preferences within user data and recommendation content datasets, providing highly relevant and personalized suggestions [5]. However, as shown in Fig. 1, LLMRec systems face two primary challenges: 1) the inherent risk of memorizing sensitive information from training data that may inadvertently lead to privacy leaks and 2) the necessity to adapt to users' evolving preferences over time by dynamically updating recommendations.

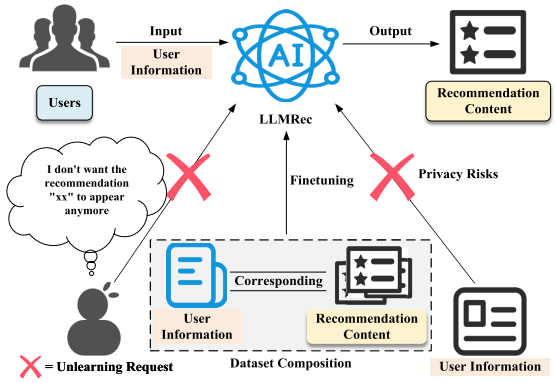

Fig. 1. Overview of the LLMRec unlearning scenario: user information and recommendation content composed dataset is used to fine-tune the LLMRec system, enabling personalized recommendations for users. However, when users request the removal of specific recommendations or in response to privacy concerns, unlearning should be performed to eliminate sensitive information from the model.

A recent study has shown that autoregressive language models like the GPT family are particularly susceptible to data extraction attacks. The sensitive user information embedded in the training data can be memorized and reproduced, risking privacy leakage during inference [6], [7]. For instance, Carlini et al. [6] demonstrated that training data extraction attacks could recover specific training examples from models like GPT-2. The attack could successfully retrieve verbatim sequences containing personally identifiable information, including names and contact details. Such vulnerabilities make it necessary for LLMRec systems to incorporate mechanisms for "forgetting" targeted sensitive user data, especially

Received 2 December 2024; revised 11 June 2025 and 30 October 2025; accepted 23 January 2026. This work was supported in part by the Qilu University of Technology (Shandong Academy of Sciences) Major Project under Grant 2025ZDZX02, in part by Shandong Provincial Natural Science Foundation Projects under Grant ZR2023QF152, and in part by Shandong Provincial University Youth Innovation and Technology Support Program under Grant 2022KJ291. *(Corresponding author: Longxiang Gao.)*

Chenchen Tan, Xinghao Li, Cunjian Chen, and Shujie Cui are with the Faculty of Information Technology (IT), Monash University, Clayton, VIC 3800, Australia (e-mail: chenchen.tan@monash.edu; xinghao.li@monash.edu; cunjian.chen@monash.edu; shujie.cui@monash.edu).

Youyang Qu and Longxiang Gao are with the Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China, and also with Shandong Provincial Key Laboratory of Computing Power Internet and Service Computing, Shandong Fundamental Research Center for Computer Science, Jinan 250014, China (e-mail: youyang.qu@data61.csiro.au; gaolx@sdas.org).

Digital Object Identifier 10.1109/TNNLS.2026.3660137

when users request their data to be removed or updated. One potential solution to ensure the removal of sensitive data from recommendation models is retraining the LLMRec from scratch, referred to as *exact unlearning*. This method guarantees that any traces of the specified data points are completely removed, rendering the model's behavior identical to a version that was never exposed to the unlearned data [8]. However, given the size and complexity of LLMRec, which often contains billions of parameters, this approach is highly resource-intensive and has unacceptable costs.

To reduce the reliance of LLMs on specific data without full retraining, recent research has proposed *approximate unlearning* techniques, which can be broadly categorized by training mechanism.

1) Objective-tuning-based methods directly update model parameters using modified objectives, such as gradient ascent (GA) [9], negative preference optimization (NPO) [10], and inverted hinge loss (IHL) [11] to push the model away from memorized outputs.
2) Knowledge distillation (KD)-based methods train a student model to imitate a modified teacher that excludes target knowledge [12], [13].

However, these methods are less suitable for LLMRec systems, where unlearning must balance two competing goals: removing sensitive associations while preserving high-quality generation. Existing approaches often over-suppress the model [9], [10], degrading recommendation fluency, or apply rigid suppression objectives that lack semantic control, making them less compatible with generation tasks [11], [13]. In addition, methods like [12] rely heavily on large retained datasets to recover performance, which may be impractical in real-world unlearning scenarios. These limitations highlight the need for a more flexible and data-efficient unlearning solution tailored to the characteristics of LLMRec systems.

In this work, we define knowledge unlearning for LLMRec to achieve a balance between effective unlearning and performance retention. As shown in Fig. 1, the LLMRec training dataset is composed of two key elements: user information and recommendation content. These two components correspond to distinct unlearning needs within the system. The first is to mitigate the risk of privacy leaks inherent in autoregressive language models like the GPT family. The second need is to unlearn the user's original recommendation preferences and generate new ones. Our approach is designed to address both aspects in a unified framework.

We propose an efficient unlearning approach combining neural network-based adapter modules with a KD-powered logit modification mechanism. Our method starts with GA to induce divergence between the model's knowledge and the target data. This step sets a foundation for a teacher–student framework that enables controlled knowledge transfer. The KD framework guides a student model in learning the modified output distribution of the teacher (original) model. Through this design, the model unlearns the target data while preserving critical generation capabilities, thus avoiding catastrophic forgetting. Unlike full knowledge inversion through GA, which compromises the entire knowledge structure, our framework enables more selective and semantically aligned unlearning. This balance is particularly important in LLMRec, where unlearning must preserve recommendation fluency. In addition, unlike prior distillation-based unlearning methods [12], our approach does not require access to the entire retained dataset to recover model performance, enabling more efficient and targeted unlearning. To make the unlearning process computationally efficient, we employ adapter modules that can be strategically inserted into key modules of the LLMRec architecture. These adapters, a type of neural network layer, allow us to restrict updates to only a subset of the model's parameters. Finally, an adapter-based performance maintenance module activates when performance drops below a predefined threshold, ensuring that the mapping between user information and recommended content remains intact. This balanced approach supports efficient, scalable unlearning, making it well-suited for large LLMRec systems by maintaining both privacy protection and recommendation performance. In summary, we have made the following contributions to this research.

1) *KD-Powered Logit Modification Unlearning Mechanism:* We propose an LLMRec unlearning method that enables effective unlearning without full knowledge inversion or complete access to the training dataset, preserving model utility. GA initially drives divergence between the model's existing knowledge and the target data. After that, a frozen teacher (original) model with modified logits guides the student model to learn alternative outputs for the target knowledge.
2) *Adapter-Based Unlearning for Fintuning Efficiency:* The adapters are introduced to key modules in the Transformer architecture to streamline unlearning updates, making the approach scalable for LLMRec systems.
3) *Performance Maintenance Module:* We design a maintenance module to fine-tune the over-unlearned LLMRec on a subset of remaining data to retain recommendation quality with minimal computational costs.
4) *Experimental Validation on Diverse Datasets:* We conduct comprehensive experiments on multiple datasets, including hybrid datasets that combine real-world and synthetic data, as well as large-scale recommendation datasets, such as Yelp, Movielens, and Steam. Results demonstrate that our method consistently balances the unlearning effectiveness and model utility across different scenarios.

The rest of this article is organized as follows. Related works are presented in Section II. Section III introduces the preliminaries of the GA method for LLM unlearning. The details of the proposed unlearning structure in the LLMRec system are presented in Section IV. Section V conducts a comprehensive experiment to complete the performance evaluation of the proposed unlearning method. The limitations of the proposed method are discussed in Section VI. Finally, Section VII summarizes and concludes this article.

II. Related Works

This section reviews relevant works on machine unlearning in the context of LLMs, with a focus on recent strategies

developed for general-purpose LLMs and their extensions to LLMRec. We begin with a brief background on LLM privacy concerns and the motivation for unlearning, followed by a discussion of representative methods across different paradigms. Finally, we highlight the unique challenges of unlearning in LLMRec settings, including personalization, generation quality, and deployment efficiency.

A. Background on LLMs and Machine Unlearning

LLMs, such as GPT-4 [14] and Gemini [15], have demonstrated strong generalization across natural language processing (NLP) tasks. However, their training on massive user-generated content raises privacy concerns, as studies have shown they may memorize and regurgitate sensitive information [6], [16]. To address such risks, machine unlearning has emerged as a promising direction. Existing unlearning approaches are broadly categorized into exact unlearning, which retrains models to erase specific samples, and approximate unlearning, which modifies parameters or predictions without full retraining [17], [18]. Due to the high training costs associated with deep learning models, recent research trends have focused on designing efficient approximate unlearning schemes to effectively remove target training data while balancing computational efficiency and unlearning effectiveness [19], [20], [21].

B. Approaches to Machine Unlearning for LLMs

One line of research focuses on inverse-objective optimization. Jang et al. [9] applied GA to maximize the negative log-likelihood of target tokens, but this disrupted related representations and harmed general utility. Zhang et al. [10] proposed NPO, which stabilizes LLM unlearning optimization better than GA. Yao et al. [22] improved GA by generating template responses and adding mismatch loss. More recently, Wang et al. [23] proposed FLAT, which adjusts loss using template responses and requires only forget data, removing dependence on retain sets or reference models.

To address instability, researchers have also explored logit-level interventions. Cha et al. [11] proposed an IHL that suppresses target tokens while encouraging alternatives. Ji et al. [24] introduced unlearning from logit difference, subtracting logits with an assistant model trained on forget data. Yuan et al. [25] suggested maximizing entropy or aligning logits with rejection templates for distribution-level control.

Architectural and neuron-level strategies provide another direction. Chen and Yang [12] used distilled adapters to selectively forget knowledge. Shi et al. [26] introduced constrained knowledge unlearning, constraining gradient updates on target knowledge neurons to unlearn harmful data. Liu et al. [27] developed selective disentanglement unlearning, disentangling biased knowledge from reasoning ability using shadow-model guided saliency.

C. Unlearning in LLMRec Systems

In the context of LLMRec, a few studies have begun to address domain-specific challenges. Wang et al. [28] proposed E2URec, which enhances efficiency by updating a small set of LoRA parameters under a KD framework. Hu et al. [29] introduced adapter partition and aggregation, retraining only affected adapters while preserving recommendation quality at reduced cost. Despite these advances, unlearning in LLMRec still struggles with efficiency–utility tradeoffs. KD-based methods require access to the full dataset, GA-based methods often fail to converge, and adapter retraining incurs high storage costs when unlearning discrete samples.

Overall, existing works on LLM unlearning differ in their primary focus: some aim to aggressively remove data traces [9], [10], [30], while others emphasize preserving model utility during the unlearning process [11], [22], [24]. At the same time, concerns have been raised that many approaches achieve only surface-level unlearning, with sensitive information still embedded in deeper representations and vulnerable to adversarial recovery [30], [31]. However, recent studies also indicate that LLMs already struggle with maintaining logical consistency and reasoning quality [32], which implies that unlearning mechanisms may further exacerbate these weaknesses if not carefully designed. Taken together, these observations suggest that unlearning must balance multiple, and sometimes conflicting, objectives, including unlearning effectiveness, robustness, and utility preservation.

Importantly, the unlearning requirements vary substantially across different LLM downstream tasks. For example, the factual answering task needs an unlearned LLM to answer exact facts or refuse to answer, but the generation task or recommendation task needs the LLM to continually generate creative and reasonable answers after unlearning. Thus, beyond generic challenges, unlearning solutions need to be tailored to the demands of specific applications. In particular, existing methods have not yet accounted for the unique characteristics of LLMRec systems, such as personalization, interaction history, and recommendation quality, which motivates our work.

III. Preliminaries on GA-Based LLM Unlearning

LLMRec is fine-tuned on generative pretrained language models, whose core training objective is to maximize the likelihood of token sequences based on their surrounding context. The objective of an LLM is to maximize the log-likelihood of the tokens given their context. Given a target dataset $\mathcal{D} = \{(\mathbf{x}_{-i}, y_i)\}_{i>=1}^{N}$, the loss function for standard LLM training is

$$\mathcal{L}_{\text{NLL}} = -\sum_{i=1}^{N} \log P(y_i|\mathbf{x}_{-i}, \theta). \tag{1}$$

Here, $\mathcal{L}_{\text{NLL}}$ is the negative log-likelihood of the target token y_i; $\mathbf{x}_{-i}$ represents the context for predicting y_i, which may include tokens before and/or after y_i depending on the model type; N is the number of data samples in $\mathcal{D}$; θ represents the model parameters.

The GA-based method for unlearning in LLMs involves training the model with an objective that is the inverse of the original ground-truth prediction, i.e., minimizing the model's likelihood on each token in the target dataset $\mathcal{D}_f = \{(\mathbf{t}_{-i}, h_i)\}_{i>=1}^{F}$. Here, the target dataset is the part of the data in the training dataset $\mathcal{D}$ that should be unlearned in this model,

and the rest of the data forms a remaining dataset (RD) $\mathcal{D}_r$. The loss function can be described as

$$\mathcal{L}_{\text{GA}} = \sum_{i=1}^{F} \log P(h_i|\mathbf{t}_{-i}, \theta) \tag{2}$$

where $\mathcal{L}_{\text{GA}}$ is the log-likelihood of target token h_i; $\mathbf{t}_{-i}$ refers to a context sequence of tokens except h_i; F is the number of data samples in $\mathcal{D}_f$.

While this approach effectively achieves knowledge unlearning, it negatively impacts the user experience and makes the model produce extremely opposite to the original outputs after unlearning. It may result in inappropriate or irrelevant outputs, even unreadable text, decreasing the quality of personalized content, and overall user satisfaction. However, directly applying GA in LLMRec leads to the training objective function never converging. Originally, the LLMRec training process had reached a convergence state where the objective function minimized the negative log-likelihood on the target data. However, GA introduces a loss function with an inverted goal, aiming to minimize the positive original loss. This results in the current loss trending toward negative infinity, meaning the model will keep training without ever converging. As a result, the system will suffer from severe performance degradation, unstable outputs, and increasingly irrelevant and unpredictable recommendations. These outcomes are particularly problematic in LLMRec systems, where generating relevant and natural suggestions is crucial.

IV. Knowlege Distillation-Powered Logits Modification-Based LLMRec Unlearning

To meet the fine-grained unlearning requirements of LLMRec, we propose a KD-based approach that modifies the output logits of the model. Instead of directly inverting knowledge or applying hard constraints, our method enables controlled adjustment of token probabilities to suppress sensitive outputs. This allows the model to avoid generating target content while preserving overall performance and coherence. By distilling from a modified teacher distribution, our approach provides a stable unlearning signal and reduces the risk of catastrophic forgetting.

A. Enhanced Logits Modification Unlearning via Teacher–Student Architecture

As illustrated in Fig. 2, we begin by freezing the current LLMRec model to serve as the teacher. The logits of the teacher model are then edited to reduce the probabilities of tokens directly corresponding to the target data. In addition, tokens with high semantic similarity, which are measured by cosine similarity in embedding space, are also suppressed. This design ensures that the teacher's predictions are less biased toward the original sensitive outputs, aligning better with the logic of generative recommendation. A student model (target LLMRec) is trained to distill this modified distribution, learning to avoid sensitive outputs while maintaining fluency and personalization. The teacher–student framework offers two key advantages: 1) it provides a stable training target

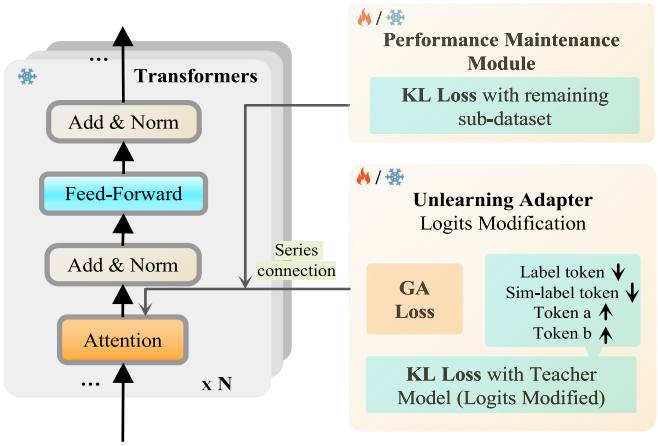

Fig. 2. Illustration of the LLMRec unlearning system structure. We design two types of adapters embedded within each Transformer block for this unlearning structure. The unlearning adapter enables unlearning the target data, while the performance maintenance module preserves model performance. The performance maintenance module remains frozen unless the recommendation system's performance drops below a defined threshold. The unlearning adapter removes target data in two steps: 1) GA weakens the model's memory of specific data by setting tailored degrees, epochs, and weights, creating divergence in the model's knowledge and 2) a frozen teacher model (original LLMRec model) with modified logits guides the student model to adjust probabilities, reducing those of target tokens and their similar tokens (Sim-label tokens) while enhancing unrelated high-probability tokens.

unaffected by continuous updates and 2) it avoids the instability often observed in optimization-based suppression. By guiding the student to mimic a carefully edited output space, the model unlearns specific associations without disrupting unrelated knowledge.

In exploring self KD (SKD) [13] for unlearning in the LLMRec system, we find that directly supervising the student model using pseudo labels derived from the edited teacher outputs via a cross-entropy (CE) loss does not lead to the most effective unlearning. Although the teacher logits are modified to suppress target information, the resulting pseudo labels often remain close to the original targets. In autoregressive model-based LLMRec, where predictions are highly sensitive to sequential context, this minimal difference between edited and original labels fails to generate sufficient gradient signals to drive meaningful unlearning. As a result, the student model continues to retain strong associations with the target data, and the intended unlearning effect is significantly weakened.

To address this challenge, we propose a two-step strategy. In addition to refining the logits modification scheme, we first use GA, as described in (2) to ensure that LLMRec gradually diverges from its original knowledge structure. The student model, after GA processing, will lose a part of its capability, which increases the knowledge gap between the student and teacher models. This larger gap makes it easier for the student model to relearn the teacher model's modified knowledge from scratch. In this way, this method magnifies the differences between the two models, which is in line with the knowledge transfer process of KD.

The logits modification strategy of the teacher model outputs, as shown in Fig. 2, is represented as

$$z_i'^{\text{teacher}} = \text{lm}\left(z_i^{\text{teacher}}\right) \tag{3}$$

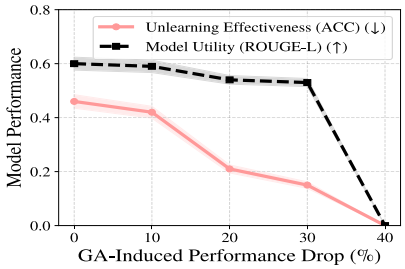

Fig. 3. Impact of GA-induced performance degradation on unlearning effectiveness and model utility in LLaMA-7B.

where $z_i^{\text{teacher}} = f(\mathbf{t}_{-i}, \theta_{\text{teacher}})$, which is the output logits of teacher model. $\text{lm}(z_i)$ can be considered a hyperparameter that encompasses the logits modification for the ground truth (label), similar tokens to the ground truth, and high-probability substitute tokens.

The predicted probabilities for the student and teacher models are computed using the softmax function

$$P(h_i|\mathbf{t}_{-i}, \theta_{\text{student}}) = \text{softmax}\left(z_i^{\text{student}}\right) \tag{4}$$

$$P\left(\hat{h}_i|\mathbf{t}_{-i}, \theta_{\text{teacher}}\right) = \text{softmax}\left(z'^{\text{teacher}}_i\right). \tag{5}$$

To enforce the student model to mimic the output distribution of the teacher model on the target unlearning dataset (TUD) $\mathcal{D}_f$, we apply a Kullback–Leibler (KL) divergence loss between their predicted distributions. The KL divergence loss for the KD-based logits modification approach can be written as

$$\mathcal{L}_{\text{KL},\mathcal{D}_f} = \mathbb{E}_{(\mathbf{t}_{-i},h_i)\in\mathcal{D}_f}\left[D_{\text{KL}}\left(P\left(\hat{h}_i|\mathbf{t}_{-i}, \theta_{\text{teacher}}\right) \| P(h_i|\mathbf{t}_{-i}, \theta_{\text{student}})\right)\right]. \tag{6}$$

The final unlearning loss combines the weighted GA loss and KL loss, as follows:

$$\mathcal{L}_{\text{unlearning}} = \alpha \cdot \mathcal{L}_{\text{GA}} + (1-\alpha)\cdot \mathcal{L}_{\text{KL}}. \tag{7}$$

Both GA loss and KL loss play essential roles in this two-step unlearning strategy. The GA loss ensures that the model unlearns its original associations by forcing it to diverge from previously learned knowledge. In contrast, the KL loss helps maintain the overall output distribution by ensuring that the probabilities of related tokens are adjusted according to the teacher model's modified logits. This balance ensures that the unlearning process does not degrade the model's recommendation and language reasoning abilities. Our approach operates at the output distribution level by modulating the token-level logits, a mechanism shared by most Transformer-based language models. This design ensures compatibility across different LLM architectures without requiring modifications to their core components.

To ensure this two-step process works effectively, it is critical to manage the degree of divergence introduced during the GA phase. Excessive perturbation may cause irreversible damage to the model's structure, making the subsequent KL-based recovery ineffective. On the other hand, insufficient divergence may leave the target knowledge intact, reducing the overall unlearning performance. Therefore, a careful balance between degradation and recoverability must be maintained when transitioning from GA to KL training. As shown in Fig. 3, when GA causes less than 10% performance drop on the retained dataset, the unlearning effectiveness remains high (above 0.4), indicating that the target knowledge is not unlearned. In contrast, a moderate drop of 20%–35% significantly reduces the model's accuracy on the unlearning dataset, while still maintaining acceptable utility. However, when GA-induced degradation exceeds 40%, the model exhibits near-zero utility and irrecoverable collapse. These findings emphasize the importance of monitoring and limiting performance degradation during GA to enable stable and efficient unlearning in the subsequent KL phase.

B. Convergence Analysis

The total unlearning loss function in (7) is defined as the weighted sum of the GA loss and the KL divergence loss. Initially, the GA loss dominates with a larger weight, but as training progresses, its coefficient α gradually decreases to zero. Consequently, during the convergence phase, only the KL divergence term influences the optimization process. We analyze the convergence behavior of the unlearning loss from both theoretical and empirical perspectives.

1) Theoretical Convergence Analysis:

a) Distribution space convexity: When the teacher distribution P^t is fixed, the KL divergence $D_{\text{KL}}(P^t \parallel P^s)$ is convex with respect to the student distribution P^s over the probability simplex. By Gibbs' inequality, it admits a unique global minimum attained at $P^s = P^t$. This ensures that, in the distribution space, the optimization landscape has a single optimal solution.

b) Parameter space nonconvexity: The student model's distribution P^s_θ is parameterized by deep neural networks. The mapping $\theta \mapsto P^s_\theta$ introduces nonconvexity to the loss function, precluding global optimality guarantees. Nonetheless, under standard smoothness assumptions, gradient descent-type algorithms are known to converge to stationary points [33].

c) Stationary point convergence: Assume the KL loss $\mathcal{L}_{\text{KL}}(\theta)$ is lower bounded and L-smooth (i.e., its gradient is L-Lipschitz), and that the stochastic gradient has bounded variance σ^2, i.e.,

$$\mathbb{E}\left[g_t\right] = \nabla\mathcal{L}_{\text{KL}}(\theta_t), \qquad \mathbb{E}\|g_t - \nabla\mathcal{L}_{\text{KL}}(\theta_t)\|^2 \le \sigma^2.$$

Let η be the learning rate. Choosing

$$\eta = \min\left(\frac{1}{L}, \frac{c}{\sqrt{T}}\right), \quad c = \sqrt{\frac{4\Delta_0}{L\sigma^2}}, \quad \Delta_{T_0} = \mathcal{L}_{\text{KL}}\left(\theta_{T_0}\right) - \mathcal{L}^*_{\text{KL}}$$

where $\mathcal{L}^*_{\text{KL}} = \inf_\theta \mathcal{L}_{\text{KL}}(\theta)$ denotes the global minimum value of the KL objective. The stochastic gradient descent (SGD) iterates $\{\theta_t\}_{t=T_0}^T$ satisfy

$$\min_{T_0\le t\le T} \mathbb{E}\|\nabla\mathcal{L}_{\text{KL}}(\theta_t)\|^2 \le \frac{2\left(\mathcal{L}_{\text{KL}}\left(\theta_{T_0}\right) - \mathcal{L}^*_{\text{KL}}\right)}{\eta T} + \frac{\eta L\sigma^2}{2}. \tag{8}$$

This standard result for nonconvex SGD implies an $O(1/\sqrt{T})$ convergence rate toward an ϵ-stationary point. Hence, after $T = \mathcal{O}(1/\epsilon^2)$ iterations, the model parameters lie near such a stationary region [34]. The above guarantee applies to the KL minimization phase ($t \ge T_0$) of (7), where $\alpha = 0$, while

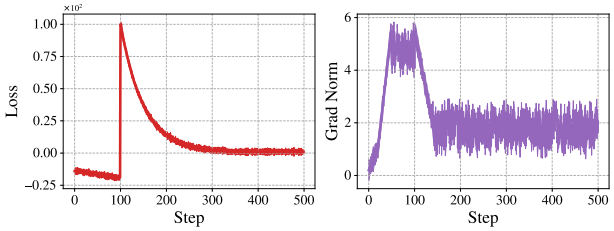

Fig. 4. Training loss and gradient norm of the proposed unlearning method work in LLaMA-7B for unlearning 256 samples.

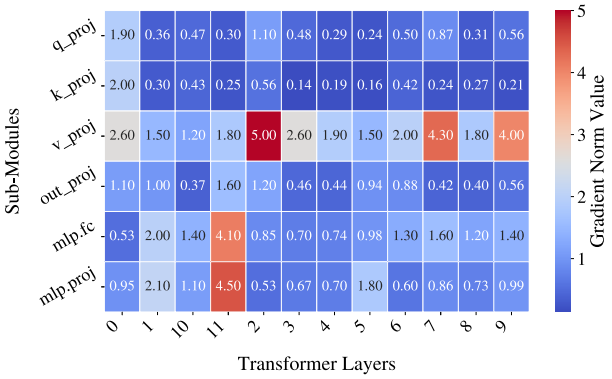

Fig. 5. Heatmap of gradient norms across transformer layers during full-parameter updates with the proposed method. The data rows represent the average gradient norms across submodules within each layer. Higher gradient norm values indicate areas with a greater impact of processing the targeted unlearning.

the preceding GA phase ($t < T_0$) primarily serves to induce divergence from the target knowledge.

2) Empirical Validation of Convergence: We empirically verify the convergence behavior of our two-stage unlearning method by tracking the training loss and gradient norm throughout the optimization process, as shown in Fig. 4. In the first stage, GA encourages the model to diverge from the memorized target outputs, resulting in a steady decrease in negative loss. Upon transitioning to the second stage, where the KL divergence is minimized based on a teacher model, we observe a temporary spike in both the loss and the gradient norm due to the abrupt change in the optimization direction. Despite this transient instability, the training process quickly stabilizes. The loss rapidly decreases and converges to a plateau, while the gradient norm exhibits bounded oscillations without divergence. This behavior indicates that the optimizer enters and remains in a stationary region, consistent with our theoretical analysis of convergence to an ϵ-stationary point. Notably, the loss and gradient remain well-behaved throughout training, despite the inherent nonconvexity of the KL divergence to model parameters. These results confirm that our decoupled two-phase design avoids conflicting gradient signals and supports stable convergence in practice. The empirical patterns align with the theoretical guarantees, reinforcing the effectiveness of our framework in achieving reliable and efficient unlearning.

C. Adapter Design and Selective Activation Strategy

To enable efficient and controllable unlearning in LLMRec systems, we incorporate bottleneck-style adapter modules into the target model, following the architecture proposed by Houlsby et al. [35]. Each adapter consists of a down-projection layer to 64 dimensions, a ReLU activation, and an up-projection back to the original hidden size. No dropout is applied within the adapter. As shown in Fig. 2, all base model parameters are frozen throughout training, and only adapter parameters are updated. This design allows for localized and efficient parameter updates without compromising the generalization capacity of the backbone model. Based on gradient attribution analysis (see Fig. 5), we observe that both the attention and feedforward network (FFN) modules exhibit high gradient norms on the target data, indicating that these components are most responsible for processing target knowledge. Accordingly, adapters can be inserted after both sublayers to modulate their representations. In our experiments, we find that inserting adapters after the attention layers leads to less disruption in the model's general behavior, while still enabling effective unlearning. This configuration strikes a favorable balance between intervention granularity and model stability. In addition, since adapter modules are injected independently of the base model weights, they offer a modular and nonintrusive way to apply unlearning across diverse LLM architectures.

To balance knowledge removal with recommendation utility, we implement a dual-adapter strategy. One set of adapters, referred to as *unlearning adapters*, is active during the KD phase and absorbs the edited logits to suppress target knowledge. A second set of structurally identical adapters, called *performance maintenance adapters*, is kept frozen initially and is selectively activated after unlearning. Both sets of adapters are inserted after the attention layers in each Transformer block. Upon activation, only the performance adapters are fine-tuned on a small subset of retained data. This selective activation strategy ensures that only one set of adapters is trainable at a time and avoids interference between unlearning and recovery processes.

D. Performance Maintenance Module

While the aforementioned logit modification unlearning approach effectively mitigates the performance impact on the LLMRec model, increasing the scope and intensity of unlearning demands may still lead to degradation in the model's overall performance. In particular, when the unlearning task becomes more stringent, key functionalities such as the quality of generated content and recommendation accuracy can be compromised, resulting in noticeable performance declines.

We introduce a flexible performance maintenance module to enhance the stability of the model's performance, particularly under intensive unlearning scenarios. However, the objectives of unlearning and performance restoration are fundamentally conflicting: unlearning aims to erase specific knowledge, while maintenance seeks to preserve or recover it. Updating the same adapter parameters for both tasks can lead to gradient interference, undermining the effectiveness of each.

To resolve this, the performance maintenance module is designed to activate only when the model's utility drops

below a predefined threshold. Upon activation, it selectively unfreezes a dedicated set of adapters and fine-tunes them on a subset of the retained dataset. This helps recover recommendation quality by reinforcing the alignment between user representations and generated outputs. The module operates adaptively, adjusting its intervention based on the actual performance impact of the unlearning process. Its primary goal is to preserve high-quality recommendations with minimal additional computational overhead.

Algorithm 1 Unlearning With Adaptive Performance Maintenance

Input: Forget set $\mathcal{D}_f$, Retain set $\mathcal{D}_r$, Validation set $\mathcal{D}_{val}$
Output: Unlearned model $\theta_{student}$
Set mode = `unlearn`;
for *epoch* $= 0$ **to** *max_epochs* **do**
  if *mode* == `unlearn` **then**
    Set adapter mode to `unlearn`;
    foreach *batch in* $\mathcal{D}_f$ **do**
      $\mathcal{L}_{\text{unlearning}} = \alpha \cdot \mathcal{L}_{\text{GA}} + (1-\alpha) \cdot \mathcal{L}_{\text{KL}}$;
      Update model;
  else if *mode* == `maintain` **then**
    Set adapter mode to `maintain`;
    foreach *batch in subset* $\hat{\mathcal{D}}_r \subset \mathcal{D}_r$ **do**
      $\mathcal{L}_{\text{maintain}} = \mathcal{L}_{\text{KL}}(\hat{\mathcal{D}}_r)$;
      Update model;
  Compute $\mathcal{P}(\theta)$ on $\mathcal{D}_{val}$;
  if $\mathcal{P}(\theta) < \mathcal{T}_{low}$ **then**
    mode = `maintain`;
return $\theta_{student}$

The performance of the model is measured using a performance function $\mathcal{P}(\theta)$, which evaluates the model's performance on the RD $\mathcal{D}_r$ using the ROUGE metric [36]

$$\mathcal{P}(\theta) = \text{ROUGE}\left(\hat{h}_i, h_i\right) \tag{9}$$

where $\hat{h}_i$ and h_i are the predicted and reference outputs. As shown in Algorithm 1, if $\mathcal{P}(\theta)$ falls below a threshold hyperparameter $\mathcal{T}_{\text{low}}$, the maintenance module is triggered. Its objective uses KL divergence losses

$$\mathcal{L}\text{maintain} = \mathcal{L}\text{KL}\left(\hat{\mathcal{D}}_r\right) \tag{10}$$

where $\mathcal{L}_{\text{KL}}$ guides the unlearned model to match the output distribution of a reference model, thereby mitigating unintended behavioral drift. The complete working flow of the maintenance adapter is shown in Algorithm 1.

V. Experimental Results

In this section, we present a series of experiments to perform a comprehensive evaluation of the proposed method. Let us revisit the three research questions identified above as follows.

1) Can the proposed approaches effectively unlearn the target data?
2) Can the proposed approaches maintain LLMRec system performance while unlearning the data?
3) What advantages do the proposed approaches have compared to existing methods?

We will validate the effectiveness of the proposed method in addressing them with the following experiments and results.

A. Experimental Configurations

1) Target Models: The experiments were conducted on LLaMA (7B and 13B), GPT-NEO (125M and 2.7B), and T5 (Large and 3B) models. The LLaMA series is a decoder-only Transformer architecture introduced by Meta AI, designed for efficient open-ended text generation [37]. The GPT-NEO series is a Transformer-based language model built upon GPT-3 architecture, designed and replicated by EleutherAI [38], trained on the Pile corpora dataset [39]. These models were trained as masked autoregressive language models. The T5 model is an encoder–decoder Transformer architecture introduced by Google Research, trained on the C4 corpus [36]. It was designed to unify all NLP tasks under a text-to-text framework, converting tasks such as classification, translation, and summarization into sequence-to-sequence generation tasks. These models were chosen for their architectural diversity and functional alignment with our dual unlearning objectives, allowing us to systematically assess unlearning strategies across heterogeneous recommendation system designs. All experiments were conducted on a server equipped with a single Intel Xeon Icelake processor (16 physical cores across two sockets) and two NVIDIA A100 GPUs (40GB each). For GPT-NEO, T5 models, and LLaMA-7B, training with adapter or LoRA modules was performed using standard data parallelism on a single GPU without requiring model partitioning. However, due to the larger parameter footprint of LLaMA-13B, training required model parallelism across both GPUs using model sharding techniques, e.g., via DeepSpeed. This distinction reflects the additional system-level considerations when applying unlearning to larger LLMs.

One of the unlearning goals is to selectively unlearn user preferences in LLMRec. This unlearning goal applies to all types of LLMRec systems. They capture patterns and mappings between user preferences and recommended content to get the recommendation capability. When a user requests the deletion of certain preferences, the challenge is to selectively unlearn these preferences without disrupting the overall recommendation logic. Another is to unlearn the user information that is recorded verbatim by the model. This goal aims to remove specific, memorized user information from the model to prevent it from being reproduced during inference, thereby preserving users' privacy. This type of unlearning primarily applies to autoregressive models, which tend to memorize verbatim input–output mappings during training.

2) Baseline: We compared the proposed approaches with: 1) the inverting LLM training objective-related methods, including GA [9] and NPO [10]; 2) the logits-manipulate-based method, including IHL [11]; and 3) KD-based methods, including EUL [12] and SKD [13]. The comparison was performed from the perspectives of unlearning effects and the remaining performance of the unlearning model.

3) Datasets: We utilized three public datasets to evaluate unlearning in LLM recommendation scenarios: Amazon Product Reviews [40], Yelp Review Full [41], and MovieLens [42].

The Amazon dataset contains user-written product reviews along with anonymized user and product IDs. Since the IDs lack semantic information, language models struggle to generate personalized recommendations. To address this limitation, we used GPT-4 to enrich the dataset.[1] We applied the same strategy to the Yelp Review Full dataset [41], which includes full-text user reviews and 5-star ratings. We selected only 3-star or above rated reviews, then used GPT-4 to extract item descriptions from the review text. This augmented dataset with more structured and informative references for the recommendation model.

Our enhanced Amazon Review dataset consists of 1500 samples, each containing synthesized user information, review text, and recommended content aligned with the review's content. The enhanced Yelp review dataset consists of 2000 pairs of samples, where each pair represents a review of a specific product. To ensure the diversity and relevance of the synthetic dataset, we conducted both quantitative and qualitative evaluations. First, we analyzed the dataset's diversity across dimensions such as product categories and user preferences. We then compared the generated recommendations with the corresponding review information using the ROUGE-L metric, achieving an average score of around 50%. This score reflects a balanced overlap between recommended content and user information, supporting the dataset's effectiveness in generating coherent and contextually appropriate recommendations. Finally, a manual evaluation was conducted by randomly sampling a subset of entries, where we assessed the consistency and relevance of each recommendation with the associated user information. This evaluation confirmed that the recommended content logically aligns with the user information provided. These datasets are used to fine-tune LLMs based on the method in [5], enabling them to better associate user behavior with relevant product suggestions, thereby enhancing their recommendation capabilities.

Finally, we followed the OpenP5 [43] framework to process the MovieLens dataset. The original dataset contains anonymized user and item IDs along with rating scores. Following OpenP5, we enhanced each item with side information, such as genres, tags, and entity-based knowledge. User interaction histories were then transformed into dialog-style contexts describing past preferences, e.g., "User liked the matrix, which is an action and sci-fi movie." The model was trained to generate recommendation responses in natural language, using the same sequence-to-sequence objective and data formatting strategy as OpenP5.

To better evaluate the effects of unlearning under different architectures and data modalities, we fine-tuned each model on the dataset most aligned with its inductive bias and architectural strengths. Specifically, we fine-tuned GPT-NEO on the Amazon Review dataset, as its autoregressive, decoder-only structure is well-suited for modeling long-form textual patterns and verbatim memorization, matching the unlearning objective of removing sensitive user-recorded information. LLaMA was fine-tuned on the Yelp Review dataset, which contains concise and user-generated product feedback. This setting allows us to assess the model's ability to generate context-aware recommendations that align with diverse user-item descriptions. For T5, we utilized the structured MovieLens data. The encoder–decoder architecture of T5 is naturally compatible with structured input–output mappings, making it suitable for evaluating the unlearning goal: selectively removing preference-based associations while preserving general recommendation logic. This design ensures each model operates in a scenario that highlights its strengths, while also providing a meaningful basis for evaluating different unlearning strategies under realistic and heterogeneous recommendation settings.

4) Evaluation Metrics: To evaluate the performance of the unlearned model, we utilize the metrics of accuracy, BLEU score [44], ROUGE [36], and forget quality (FQ) [45]. Accuracy measures how well the model predicts correct recommendations, ensuring it retains its ability to generate relevant outputs. The BLEU score assesses the precision of machine-generated text by comparing it with reference outputs. ROUGE, focusing on recall, captures the overlap between generated text and reference text, ensuring the unlearned model can still produce coherent results. We further adopt the FQ metric proposed in the TOFU benchmark [45], which evaluates the likelihood of a model preferring the correct answer on the unlearning dataset. A well-unlearned model should no longer strongly prefer the correct answer, aligning with the behavior of a retain-only model. This is quantified using the truth ratio and statistically tested via the Kolmogorov–Smirnov test. A high p-value indicates the forget model is statistically indistinguishable from a retain model on the forget set, signaling successful unlearning [45]. In addition to the traditional performance metrics, we also use membership inference attack (MIA) [8] to assess whether the unlearned model has successfully unlearned target data. MIA attempts to determine whether a specific data point was part of the model's training set by observing the model's output. A successful unlearning process would result in an MIA success rate of approximately 50% on the target unlearning data. It indicates that the model can no longer distinguish whether the data was part of its training set or the data had never been seen.

B. Main Results

Table I summarizes the comparison results of six unlearning methods across different models in terms of unlearning effectiveness, utility maintenance, and efficiency. The unlearning effectiveness is evaluated by ACC, BLEU score, and FQ, where lower (↓) ACC and BLEU scores indicate stronger unlearning of target data, and higher (↑) FQ indicates that the model no longer confidently prefers the correct answer on

[1]Prompt for user profile generating: "Given the following product review, infer the user's interests and possible demographics in two sentences." Prompt for item extraction: "Summarize the product being reviewed in one sentence based on the review content."

TABLE I

COMPARISON OF UNLEARNING METHODS ACROSS MODELS FOR UNLEARNING 32 SAMPLES. METRICS INCLUDE UNLEARNING EFFECTIVENESS (ACC, BLEU, FQ), UTILITY MAINTENANCE (ACC, ROUGE-L), AND EFFICIENCY (TIME). NUMBERS IN PARENTHESES INDICATE STANDARD DEVIATION ACROSS DIFFERENT RUNS WITH DIFFERENT UNLEARNING SAMPLES. THE GRAY ROW IS THE AVERAGE PERFORMANCE OF THE PROPOSED UNLEARNING METHOD. THE ADDITIONAL "SIG." ROWS REPORT 95% BOOTSTRAP CONFIDENCE INTERVALS OF THE DIFFERENCE ("OURS" VALUE MINUS THE BEST VALUE OF ALL THE BASELINES). SIGNIFICANCE LEVELS ARE INDICATED AS $^{***}p < 0.005$, $^{**}p < 0.01$, $^{*}p < 0.05$, AND NO MARK INDICATES NOT SIGNIFICANT

Model		Unlearning Effectiveness			Utility Maintainence		Efficiency
		ACC ↓	BLEU ↓	FQ ↑	ACC ↑	ROUGE-L ↑	Time(m) ↓
LLAMA-13B	GA [9]	**0.01 (0.04)**	**0.01 (0.02)**	0.05 (0.07)	0.03 (0.06)	0.05 (0.04)	**2.8 (1.2)**
	NPO [10]	0.21 (0.07)	19.5 (2.41)	0.24 (0.06)	0.25 (0.05)	0.41 (0.07)	2.3 (0.9)
	IHL [11]	0.22 (0.06)	18.9 (2.94)	0.36 (0.08)	0.25 (0.06)	0.40 (0.05)	5.3 (1.5)
	SKD [13]	0.50 (0.05)	31.7 (1.21)	0.17 (0.06)	0.58 (0.02)	0.55 (0.05)	-
	EUL [12]	0.21 (0.04)	19.3 (1.83)	**0.79 (0.06)**	**0.59 (0.03)**	**0.57 (0.07)**	34.6 (3.0)
	Ours	0.19 (0.05)	18.1 (1.53)	0.65 (0.07)	0.52 (0.06)	0.53 (0.04)	15.0 (2.3)
	Sig.	[0.15, 0.20]***	[17.2, 19.0]***	[-0.17, -0.11]***	[-0.10, -0.04]***	[-0.07, -0.01]**	[11.5, 13.8]***
GPT-NEO-2.7B	GA [9]	**0.02 (0.06)**	**0.36 (0.23)**	0.01 (0.03)	0.06 (0.04)	0.09 (0.06)	**5.3 (1.0)**
	NPO [10]	0.22 (0.03)	25.5 (2.81)	0.31 (0.07)	0.28 (0.05)	0.29 (0.06)	13.8 (2.2)
	IHL [11]	0.27 (0.06)	33.1 (3.22)	0.39 (0.09)	0.38 (0.06)	0.45 (0.10)	14.2 (2.4)
	SKD [13]	0.49 (0.07)	41.3 (3.94)	0.11 (0.07)	0.44 (0.04)	0.52 (0.05)	-
	EUL [12]	0.19 (0.08)	5.82 (2.94)	**0.83 (0.05)**	**0.47 (0.05)**	**0.56 (0.06)**	137.2 (8.6)
	Ours	0.14 (0.06)	4.37 (1.97)	0.78 (0.08)	0.44 (0.07)	0.51 (0.04)	36.0 (3.1)
	Sig.	[0.09, 0.15]***	[2.91, 5.07]***	[-0.09, -0.01]***	[-0.07, 0.01]	[-0.09, -0.02]**	[29.2, 32.2]***
T5-3B	GA [9]	**0.00 (0.00)**	**0.00 (0.00)**	0.00 (0.01)	0.02 (0.05)	0.05 (0.03)	**4.7 (1.2)**
	NPO [10]	0.16 (0.05)	6.38 (2.96)	0.27 (0.08)	0.31 (0.06)	0.33 (0.06)	4.2 (0.7)
	IHL [11]	0.14 (0.06)	6.19 (1.89)	0.39 (0.09)	0.41 (0.04)	0.45 (0.03)	5.4 (1.4)
	SKD [13]	0.15 (0.07)	4.85 (1.36)	0.73 (0.03)	0.46 (0.05)	0.51 (0.07)	25.6 (2.5)
	EUL [12]	0.15 (0.04)	5.03 (2.51)	**0.85 (0.06)**	**0.47 (0.04)**	**0.53 (0.04)**	81.3 (6.3)
	Ours	0.13 (0.06)	4.18 (1.55)	0.81 (0.05)	**0.47 (0.03)**	0.53 (0.05)	10.3 (1.9)
	Sig.	[0.09, 0.16]***	[3.22, 5.14]***	[-0.09, 0.01]	[-0.03, 0.02]	[-0.03, 0.02]	[5.35, 6.88]***

the TUD. For utility maintenance, higher ACC and ROUGE-L scores on the RD represent stronger retention of general recommendation ability and generation quality. Efficiency is measured by training time, which is expected to be minimized. All the data in Table I were recorded after the unlearning performance had stabilized and showed no significant changes. For SKD, the "time" column is empty because its performance has been fluctuating.

GA drives ACC/BLEU on the TUD near zero but collapses RD utility, a hallmark of catastrophic forgetting. NPO moderates this effect yet still reduces utility noticeably. IHL achieves effective unlearning in both LLaMA-13B and GPT-NEO-2.7B, but we observe that replicating the same level of unlearning in GPT-NEO requires significantly more aggressive hyperparameter settings. Despite both models sharing a decoder-only architecture, this discrepancy likely arises from differences in model capacity and training dynamics. The smaller GPT-NEO model fine-tunes more quickly on compact datasets and tends to memorize target samples more rigidly within its core representations. As a result, it becomes more resistant to soft interventions. To match the unlearning performance observed in LLaMA-13B, we found that IHL's logit-penalty hyperparameters needed to be scaled up by approximately five times in GPT-NEO.

While GA, NPO, and IHL achieve near-zero accuracy on the unlearning set, their FQ scores remain low. This indicates that although these methods suppress the correct outputs, they are introducing strong distributional shifts that diverge from the behavior of a retain-only model. As defined in the TOFU benchmark [45], a high FQ score implies that the forget model no longer prefers the correct answer and aligns statistically with a retain-only model, indicating targeted and behaviorally consistent unlearning. The low FQ scores observed in GA, NPO, and IHL reflect overly aggressive interventions that not only remove target knowledge but also distort the model's overall output distribution.

SKD introduces fine-grained unlearning via KD, but it struggles with decoder-only LLMRec models like GPT-NEO and LLaMA-13B due to their tightly coupled sequential dependencies. The softened output alignment in SKD is insufficient to disrupt such deeply embedded representations, resulting in high BLEU and low FQ scores that indicate incomplete unlearning. In contrast, our method explicitly breaks these couplings in the first stage through GA, followed by realignment via KL divergence, enabling more effective knowledge removal. On encoder–decoder models like T5-3B, SKD performs better, as the bidirectional encoder facilitates representation reorganization and the decoder allows for flexible generation. Meanwhile, EUL achieves strong unlearning and utility retention but requires access to the full dataset. Besides, it needs more time to converge because it suffers from optimization instability due to its single-adapter design that simultaneously minimizes and maximizes the KL divergence across different subsets. Our approach avoids this gradient conflict by structurally separating unlearning and recovery, ensuring each objective is optimized within a stable and focused context.

As shown in Table I, our method targets a balanced trade-off among unlearning effectiveness, utility preservation, and stability. The additional "Sig." rows report 95% confidence

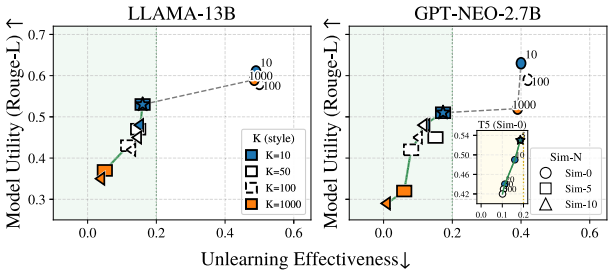

Fig. 6. Comparison of unlearning performance under parameter sensitivity across different models. We replace direct logit subtraction with a probability rank mapping: the target token is reassigned to the Kth position in the softmax order, and we optionally suppress the top N label-similar tokens (Sim-N). We evaluate configurations under two objectives, an unlearning score (mean over ten runs; lower is better) on the x-axis and model utility (Rouge-L; higher is better) on the y-axis, plotting the Pareto set (min-x/max-y) and shading the acceptable region $x \leq 0.2$. The operating point ($\star$) is selected as the Pareto optimal configuration with $x \leq 0.2$ that maximizes utility.

intervals (CIs) of ("Ours" minus the best value in baselines) per metric. Consistent with the trade-off nature of unlearning, some baselines achieve the best scores on a given metric, thus yielding intervals unfavorable to "Ours" for that column, but typically at the cost of the complementary objectives or stability. Across models, our results remain competitive on both unlearning- and retention-side metrics without catastrophic utility collapse, reflecting a balanced operating point rather than an extreme bias toward any single objective. This behavior arises from a two-stage procedure, an initial GA "kick" followed by KL-based realignment, that removes the necessary representations while limiting collateral drift. For fairness, all baselines use the same LoRA-based parameter-efficient tuning (EUL and ours are adapter-native), so differences primarily reflect algorithmic behavior rather than update budget.

C. Parameter Sensitivity of Logits Modification

For T5 (inset), modest rank shifts already suffice: moving the target back by only $K \approx 5$–10 achieves acceptable unlearning with minimal utility loss, and improvements quickly saturate once $K > 20$. This matches the observation that T5's top-20 tokens already account for most of the probability mass, so pushing the target further into the tail alters little of the local distribution while still risking quality degradation.

By contrast, LLaMA-13B and GPT-NEO-2.7B require finer-grained control of both K and Sim-N (see Fig. 6). A very large K, e.g., $K = 1000$, pushes the target into a low-probability region and strengthens the unlearning signal, but without Sim-N the model can still route information through top-ranked similar tokens, yielding incomplete unlearning. Conversely, overly large K combined with aggressive Sim-N induces strong early gradients and distributional distortion, hurting convergence and generation quality. Across both decoder-only models, a moderate rank shift ($K \approx 10$–100) with light similar-token suppression (Sim-$N \approx 5$–10) consistently lies on, or near, the Pareto front within the $x \leq 0.2$ budget, providing a clear unlearning signal while preserving utility, and serves as our default operating point.

D. Performance Across Different Numbers of Unlearning Samples and Various Model Architectures

Fig. 7 illustrates the trade-off between unlearning effectiveness and model utility across six unlearning methods, evaluated on various model architectures and under two different sample size settings for the target unlearning data. The x-axis represents the accuracy on the TUD, where lower values indicate more effective unlearning. The y-axis denotes model utility on the RD, with higher values reflecting stronger retention of general recommendation capability. In all configurations, similar to EUL, ours is positioned in the upper left region of the plots, indicating that it achieves low target accuracy while maintaining high utility on nontarget data.

The results also reveal clear trends for model size and unlearning difficulty. As the number of unlearning samples increases, all methods experience some decrease in utility, reflecting the inherent challenge of removing more knowledge without degrading general performance. However, larger models such as LLaMA-13B demonstrate greater resilience to this degradation. This can be attributed to their stronger generalization ability and greater resistance to overfitting, which makes them less reliant on tightly original bound contextual patterns when generating outputs. As a result, disrupting specific memory traces, such as associations between user profiles and product mentions, becomes more feasible in larger models without unintentionally harming unrelated behaviors. This observation is consistent with our earlier analysis of IHL, where LLaMA-13B was able to achieve strong unlearning with moderate hyperparameter settings, whereas GPT-NEO-2.7B required significantly more aggressive configurations to reach similar levels of unlearning. These results indicate that larger models are not only better at preserving utility during unlearning but also more amenable to selectively disrupting memorized associations due to their more flexible internal organization.

For the T5-3B model, we observe that most advanced methods, including EUL, SKD, and our proposed approach, converge to similar performance levels in both unlearning accuracy and utility retention. The encoder–decoder architecture of T5, with its bidirectional encoding and decoupled decoding process, appears to provide stronger representational flexibility and robustness. As a result, once the unlearning objective introduces a directional shift in knowledge, T5 can more easily reorganize internal representations to recover stable outputs, even if the specific objective differs across methods. However, it is important to note that not all methods benefit equally from this architectural advantage. GA suffers from aggressive gradient reversal that distorts general representations, while NPO's scalar preference loss lacks sufficient granularity to guide fine-grained unlearning in the presence of T5's complex encoding-decoding dynamics. These results indicate that while T5 provides a more forgiving substrate for knowledge reconfiguration, it still requires methodologically well-aligned objectives to achieve optimal outcomes.

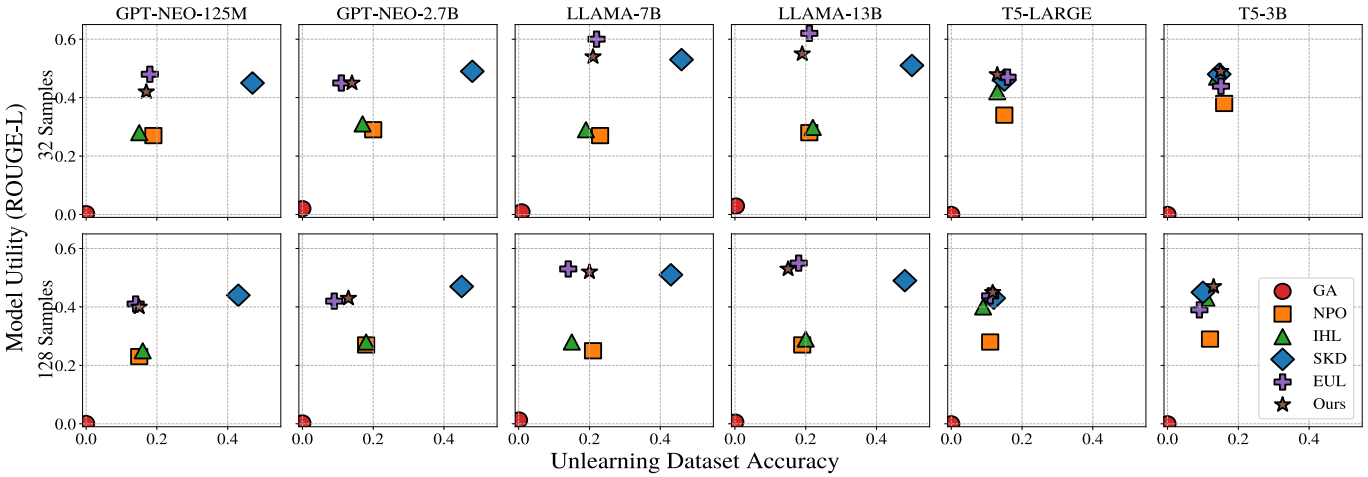

Fig. 7. Comparison of six unlearning methods on three language models, each with two different model sizes, under two unlearning sample settings (32 and 128 examples). Each subplot illustrates the trade-off between unlearning accuracy (x-axis) and model utility (y-axis). Lower values on the x-axis indicate more effective unlearning, while higher values on the y-axis represent stronger retention of general capabilities. The proposed method ("Ours") consistently achieves a favorable balance between unlearning and retention across all model configurations.

TABLE II
COMPARISON OF PREDICTIVE RESULTS OF EACH METHOD OF THREE MODELS USING THE SAME PREFIX

Status	Text-Llama	Text-GPT-NEO	Text-T5
Prefix	Based on the user review recommend the business. Review: "...The show was good. The headliner, while a bit long, was good..."; The recommended is:	Based on the user review and user profile recommend a product. Product review: "The tomatoes were... I miss the fruits..."; Profile: "Interests: Beauty treatments, Fashion...", The recommended product is:	What would ML100K user_76 be likely to purchase next after buying ML100K items, item_CI3CI46CI10, item_CI9CI43, item_CI8CI99CI0, item_CI8CI96CI3, item_CI5CI60CI1?
Original recommendation	"Premium Front Row Theater Tickets"	"Organic Fruit Box Subscription"	"ML100K item_CI3CI47CI6"
GA [9]	" "	",,,,,"	" "
NPO [10]	"Premium"	"andowder orchard and tomato salad sandwich"	"item_CCCCCCC"
IHL [11]	"Adv Ticket "Preferred Seating T"	"Tomorrow's Produce Box"	"ML C11N11"
SKD [46]	"The Premium Front Row Theater Tickets"	"Tomorrow's Special Fruit Box"	"item_C050505"
EUL [12]	"Headliner Stage Ticket"	"Fresh Tomato"	"item_C0000011"
Proposed	"**Ticket at Dave & Buster's Gift Card Store**"	"**The Fruit and Vegetable Basket**"	"**item_C1789009**"

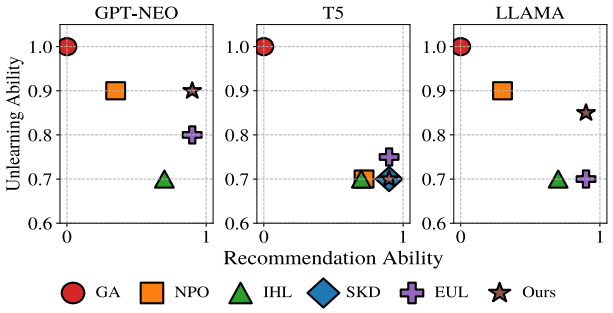

Fig. 8. Average unlearning ability and the recommendation ability after unlearning of three models.

E. Logical Analysis of Unlearned LLMRec Outputs

Our proposed unlearning method demonstrates superior performance in both recommendation quality and unlearning effectiveness, as illustrated in Table II and Fig. 8. Table II provides the qualitative examples from three model architectures using the same input prefix. The results show that our method generates semantically coherent and contextually relevant recommendations, such as "The Fruit and Vegetable Basket" or "Ticket at Dave and Buster's Gift Card Store," that differ from the original targets yet remain aligned with the input logic. This indicates that the model not only unlearns the intended user preference but also reconstructs substitutes based on its remaining knowledge. This is achieved by not only reducing the likelihood of the target data during the unlearning phase but also guiding the model to shift probability mass toward logically consistent alternatives through KL-based realignment. In contrast, GA-based methods such as GA suffer from over-aggressive optimization. As shown in the table, the outputs degenerate into incoherent or empty strings across all models, reflecting a significant loss of generation capability. NPO and IHL adopt limited granularity updating lead to insufficient control over the output distribution. As a result, they often produce fragmented or linguistically unnatural text, revealing a lack of semantic coherence after unlearning. SKD and EUL show more fluent outputs, yet SKD fails to remove target

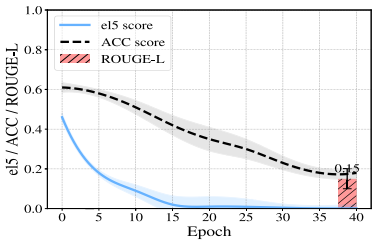

Fig. 9. Performance of the proposed unlearning method on the user information unlearning goal.

knowledge effectively in GPT-NEO and LLaMA, reaffirming its limitations in decoder-only architectures.

Fig. 8 further validates these observations with quantitative comparison across models.[2] Our method is positioned closest to the upper-right corner, confirming its ability to achieve strong unlearning while preserving recommendation utility. By contrast, GA achieves high unlearning accuracy but at the cost of nearly complete utility collapse. NPO and IHL fail to strike a balance, as they compromise output fluency while still falling short in target removal. SKD and EUL maintain better generation ability but exhibit insufficient unlearning in key scenarios.

F. Effectiveness of Unlearning User Information

In this unlearning goal, we center our analysis on the GPT-NEO model, as it is the only one in our setup trained on the Amazon Review dataset, which contains synthetic user-identifiable content. This makes it the most relevant architecture for assessing the effectiveness of user information unlearning. As shown in Fig. 9, the proposed method achieves a near-zero el_5 score after unlearning, indicating that GPT-NEO no longer reproduces memorized 5-g sequences from the original user data. The el_n metric, specifically el_5, measures the likelihood that a pretrained LLM can extract n-grams (sequences of n tokens) from a given sequence of tokens. It estimates the probability of such extractions by calculating the average success rate of extracting n-grams [9]. Although the ACC and ROUGE-L remain above zero, this does not imply a failure to unlearn, as ACC and ROUGE-L may capture partial overlaps or common language patterns unrelated to personal information.

We further evaluate the effectiveness of user information unlearning using the MIA metric. Table III presents the MIA success rates for GPT-NEO, T5, and LLaMA models. A success rate close to 0.5 indicates that the model's behavior is indistinguishable from random guessing, indicating that membership information has been successfully erased. Since GPT-NEO and LLaMA are decoder-only models, they generate output token by token, making the MIA calculation based on the success rate for each token in the target information. In contrast, T5's MIA is computed based on the prediction features of the entire target sequence, reflecting the model's prediction of the entire context. While GPT-NEO is the only model directly exposed to structured user information during training, the other models could also strongly unlearn correlations between user reviews and recommended items. The proposed method achieves MIA scores close to the ideal threshold of 0.5 across all models, further confirming its robustness in removing sensitive information.

[2]We used GPT-4 to evaluate each model's output with the following prompt: "Given the input prompt, reference output, and the model-generated recommendation, evaluate the generated recommendation on two dimensions: 1) recommendation ability (0–1): is the recommendation fluent, semantically coherent, and appropriate for the input context? 2) unlearning ability (0–1): does the generated recommendation avoid reproducing or referencing the reference output? Provide a brief justification for each score."

TABLE III
COMPARISON RESULT OF DIFFERENT METHODS ON USER INFORMATION UNLEARNING GOAL: THE el_5 SCORE OF MODEL MEMORIZING TEXT AFTER UNLEARNING AND MIA SUCCESS RATIO AFTER UNLEARNING

	$el_5\downarrow$	GPT-NEO-MIA(0.5)	T5-MIA(0.5)	LLAMA-MIA(0.5)
Original	0.586	0.793	0.762	0.770
GA	0.001	0.805	0.783	0.792
NPO	0.006	0.608	0.523	0.541
IHL	0.008	0.621	0.519	0.535
SKD	0.563	0.715	0.593	0.682
EUL	0.005	0.508	0.511	0.515
Proposed	**0.001**	**0.529**	**0.522**	**0.583**

GA's aggressive optimization introduces substantial instability. The model output becomes inverted and incoherent, which not only leads to the loss of meaningful generation but also makes the model more vulnerable to MIA. As demonstrated in Table III, the elevated MIA success rate under GA indicates that abnormal output patterns may expose training membership, thus defeating the intended privacy objective. Both IHL and NPO achieve favorable MIA performance, reflecting their ability to reduce the model's verbatim memory of user-specific content. The success of IHL stems from its use of an IHL that penalizes high-confidence predictions on target tokens. This effectively forces the model to reduce probability mass around memorized spans, thereby suppressing the regeneration of exact n-gram sequences. As a result, the model exhibits lower extractability scores and becomes less vulnerable to membership inference, since it no longer assigns distinctive confidence patterns to memorized tokens. Similarly, NPO leverages a preference-based ranking objective to reduce the relative likelihood of target completions while promoting alternative outputs. EUL demonstrates strong performance on the MIA metric, achieving a success rate close to that of the unseen dataset baseline. This indicates that, from an information-theoretic standpoint, the TUD is indistinguishable from samples not seen during training. SKD, consistent with its behavior in user preference unlearning, shows limited effectiveness on decoder-based architectures. Its soft logit editing fails to eliminate verbatim memorization, as seen from its relatively high MIA scores. However, on the T5 model, SKD performs better. Our method performs strongly across all evaluation metrics, demonstrating effective unlearning of user information and thereby upholding the user's "right to be forgotten."

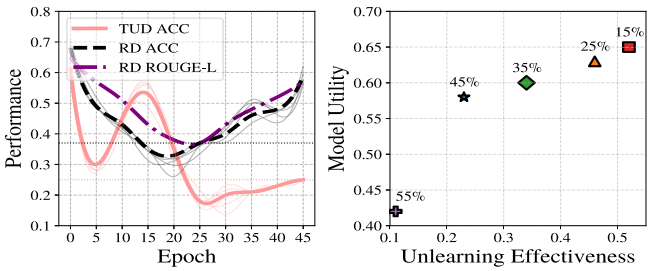

Fig. 10. Performance maintenance module effect.

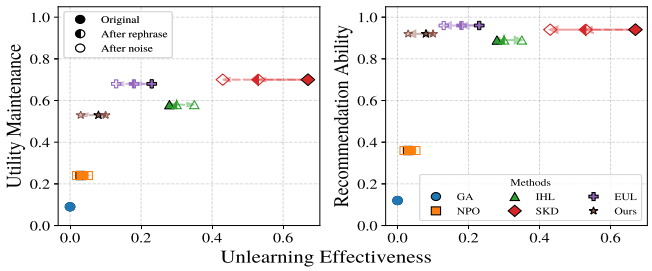

Fig. 11. Case study on the Steam dataset with 1000 unlearn samples. Comparison of unlearning effectiveness and model utility under semantic restatement and noisy perturbation adversarial settings. The left and up positions indicate better unlearning effectiveness and utility maintenance.

G. Analysis on Performance Maintenance Module

The left of Fig. 10 illustrates the alternating activation of the performance maintenance module and the unlearning module of LLaMA-13B. Initially, as TUD accuracy declines and RD accuracy drops below a predefined utility threshold, the maintenance adapter is activated to recover general performance. However, as the RD accuracy improves, the unlearning effect weakens. Once the recovery reaches an upper threshold, the training step is ended. This mechanism enables the system to dynamically balance model utility and unlearning effectiveness.

The right of Fig. 10 compares different activation thresholds for the maintenance module. Each point corresponds to a distinct utility drop condition. We observe that under higher decrease thresholds, e.g., 55%, unlearning is strong, but the recovered utility is low. Since the maintaining adapter does not use the entire retained dataset for training, but only a small portion of it, there is a certain performance bottleneck. On the other hand, lower thresholds trigger maintenance prematurely, before the unlearning has converged, resulting in weakened unlearning. Furthermore, due to the joint optimization of GA and KL losses, the transition between modules may introduce temporary optimization instability, manifested as oscillations in unlearning accuracy. Nevertheless, as the KL loss progresses, the student model gradually aligns with the edited teacher distribution and achieves stable unlearning. We observed consistent patterns across different architectures, where triggering the performance maintenance module too early weakens unlearning, while overly delayed activation leads to unrecoverable degradation in utility. Fig. 10 (right) illustrates this trend on LLaMA as a representative example. These findings indicate that the activation threshold must strike a balance. A moderate threshold is thus essential to achieve an effective trade-off between unlearning and utility preservation.

H. Large-Scale Case Study: Steam Dataset

To further evaluate the scalability and robustness of our proposed framework, we conduct a case study on the Steam dataset [47], which is one of the largest real-world benchmarks for game recommendation. The dataset comprises more than U.S. $7 million user-item interactions, spanning approximately 2 50 000 users, which provides a realistic environment for production-level evaluation.

In this experiment, we first trained the original LLMRec followed by [48], which merges the LLM and recommendation system characters to make the LLaMA-13B learn the users' interaction history mapping with the recommendation candidates. We use it to simulate large-scale unlearning requests by removing 1000 samples from the training data. To assess robustness, we additionally introduce two adversarial variants of the unlearning dataset to test unlearning performance: 1) semantic restatements, where user interactions are rephrased with equivalent meaning but different surface form wording and 2) noisy perturbations, where random characters are injected into target sequences to emulate obfuscated deletion requests. These settings approximate real-world scenarios where the prompt may be adversarial or ambiguous.

Fig. 11 reports the trade-off between unlearning effectiveness (ACC on the unlearning dataset) and model utility (ACC on the RD) under these conditions. The results demonstrate that our method maintains a favorable balance between unlearning and retention, even under large-scale and adversarial unlearning conditions. In particular, while competing methods often exhibit degradation in either unlearning effectiveness or recommendation quality, our approach remains stable during unlearning, indicating that it can scale effectively to production-level deployments.

VI. Discussion

While our adapter-based approach significantly reduces the cost of individual unlearning events, the cumulative burden under high-frequency deletion requests may still be substantial. To partially mitigate this, our framework supports batch unlearning, as shown in the experimental results for unlearning 32 and 128 samples. It can amortize computational cost across multiple deletion requests. However, batching may introduce latency, and the unlearning interface could be exposed to adversarial misuse, such as frequent or targeted deletion requests intended to disrupt model behavior. To safeguard against potential misuse, future deployments may incorporate lightweight protective mechanisms, such as rate-limiting deletion requests per user identity, request authentication via access tokens, or logging and auditing interfaces to detect abnormal request patterns. These mechanisms can be integrated at the system level to enhance robustness without interfering with unlearning behavior.

Another limitation lies in the lack of dedicated benchmarks and datasets specifically designed for evaluating unlearning in LLMRec systems. Although all datasets used in our experiments are open-source, real-world corpora collected from commercial platforms, they were not originally constructed for the purpose of LLMRec unlearning. Consequently, they

may not fully capture the challenges unique to unlearning in LLMRec and thus require additional augmentation using AI-generated content to enrich user and item descriptions. In our experiments, we refer to such augmented data as synthetic data, which serves to complement real-world datasets and provide more comprehensive coverage of user expression diversity. However, there is a lack of open-source, LLMRec systems that could serve as standardized benchmarks for evaluating unlearning under realistic deployment conditions. To partially mitigate this limitation, we adopt widely used public datasets and follow prior work [5], [43] in fine-tuning LLMs for recommendation tasks, thereby grounding our setup in established LLMRec practices.

VII. Conclusion

In this article, we introduce an efficient unlearning approach for LLMRec that ensures privacy and user preference alignment and minimizes compromising system performance. The proposed method integrates GA-driven knowledge reversal with a logit-modification-based KD mechanism. To improve scalability, we incorporate lightweight adapter modules into key Transformer modules and employ a dual-adapter strategy to decouple unlearning from utility recovery. A performance maintenance module is designed to stabilize the utility during intensive unlearning scenarios through adaptive activation on the retained subdataset. The experimental results confirm that our system achieves effective unlearning while preserving essential language and recommendation functions, making it a scalable solution for LLMRec applications.

The efficient and verifiable unlearning is more than a technical enhancement. It is also foundational for user trust and regulatory compliance. Because our adapters isolate updates to a small and auditable subset of the model, the LLMRec deployer can satisfy the "right to be forgotten" requests without full retraining. It is an ability that is especially critical in high-stakes domains, such as healthcare and finance, where interactions expose sensitive demographic or transactional patterns. By demonstrating both rapid deletion and minimal impact on recommendation accuracy, our results show that targeted unlearning can be integrated into large-scale recommendation systems, enabling target information deletion while preserving overall system quality. In future work, we will focus on certifiable deletion guarantees, adaptive scheduling, and access control for high-frequency deletion scenarios, and the development of open benchmarks that better reflect real-world LLMRec deployments.

References

[1] B. Meskó and E. J. Topol, "The imperative for regulatory oversight of large language models (or generative AI) in healthcare," *npj Digit. Med.*, vol. 6, no. 1, p. 120, Jul. 2023.

[2] F. Xing, "Designing heterogeneous LLM agents for financial sentiment analysis," *ACM Trans. Manage. Inf. Syst.*, vol. 16, no. 1, pp. 1–24, Mar. 2025.

[3] K. I. Roumeliotis, N. D. Tselikas, and D. K. Nasiopoulos, "LLMs in e-commerce: A comparative analysis of GPT and LLaMA models in product review evaluation," *Natural Lang. Process. J.*, vol. 6, Mar. 2024, Art. no. 100056.

[4] W. Hua, L. Li, S. Xu, L. Chen, and Y. Zhang, "Tutorial on large language models for recommendation," in *Proc. 17th ACM Conf. Recommender Syst.*, Sep. 2023, pp. 1281–1283.

[5] J. Ji et al., "GenRec: Large language model for generative recommendation," in *Proc. Eur. Conf. Inf. Retr.*, 2023, pp. 494–502.

[6] N. Carlini et al., "Extracting training data from large language models," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 2633–2650.

[7] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does LLM safety training fail?," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 80079–80079.

[8] M. Kurmanji, P. Triantafillou, E. Triantafillou, and T. Eleni, "Towards unbounded machine unlearning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 1957–1987.

[9] J. Jang et al., "Knowledge unlearning for mitigating privacy risks in language models," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2023, pp. 14389–14408.

[10] R. Zhang, L. Lin, Y. Bai, and M. Song, "Negative preference optimization: From catastrophic collapse to effective unlearning," in *Proc. 1st Conf. Lang. Model.*, 2024, pp. 1–10. [Online]. Available: https://openreview.net/forum?id=MXLBXjQkmb

[11] S. Cha, S. Cho, D. Hwang, and M. Lee, "Towards robust and parameter-efficient knowledge unlearning for LLMs," in *Proc. 13th Int. Conf. Learn. Represent.*, 2025, pp. 1–23. [Online]. Available: https://openreview.net/forum?id=1ExfUpmIW4

[12] J. Chen and D. Yang, "Unlearn what you want to forget: Efficient unlearning for LLMs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 12041–12052.

[13] 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 *Proc. Conf. Nations Americas Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, Apr. 2025, pp. 8827–8840.

[14] T. B. Brown et al., "Language models are few-shot learners," in *Proc. NIPS*, 2020, pp. 1877–1901.

[15] G. Team et al., "Gemini: A family of highly capable multimodal models," 2023, *arXiv:2312.11805*.

[16] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Comput. Surv.*, vol. 54, no. 11s, pp. 1–37, Jan. 2022.

[17] L. Bourtoule et al., "Machine unlearning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 141–159.

[18] T. Shaik, X. Tao, H. Xie, L. Li, X. Zhu, and Q. Li, "Exploring the landscape of machine unlearning: A comprehensive survey and taxonomy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 7, pp. 11676–11696, Jul. 2025.

[19] V. S. Chundawat, A. K. Tarun, M. Mandal, and M. Kankanhalli, "Zero-shot machine unlearning," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2345–2354, 2023.

[20] A. K. Tarun, V. S. Chundawat, M. Mandal, and M. Kankanhalli, "Fast yet effective machine unlearning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 13046–13055, Sep. 2024.

[21] W. Wang, C. Zhang, Z. Tian, and S. Yu, "Machine unlearning via representation forgetting with parameter self-sharing," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 1099–1111, 2024.

[22] Y. YangLiu, X. Xu, and Y. Yao, "Large language model unlearning," in *Proc. Adv. Neural Inf. Process. Syst. 37*, 2024, pp. 105425–105475.

[23] Y. Wang et al., "LLM unlearning via loss adjustment with only forget data," in *Proc. 13th Int. Conf. Learn. Represent.*, 2025, pp. 1–29. [Online]. Available: https://openreview.net/forum?id=6ESRicalFE

[24] S. Chang et al., "Reversing the forget-retain objectives: An efficient LLM unlearning framework from logit difference," in *Proc. Adv. Neural Inf. Process. Syst. 37*, 2024, pp. 12581–12611.

[25] X. Yuan, T. Pang, C. Du, K. Chen, W. Zhang, and M. Lin, "A closer look at machine unlearning for large language models," in *Proc. 13th Int. Conf. Learn. Represent.*, 2025, pp. 1–26. [Online]. Available: https://openreview.net/forum?id=Q1MHvGmhyT

[26] Z. Shi et al., "Safety alignment via constrained knowledge unlearning," in *Proc. 63rd Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2025, pp. 25515–25529.

[27] Z. Liu et al., "Disentangling biased knowledge from reasoning in large language models via machine unlearning," in *Proc. 63rd Annu. Meeting Assoc. Comput. Linguistics*, 2025, pp. 6105–6123.

[28] H. Wang et al., "Towards efficient and effective unlearning of large language models for recommendation," *Frontiers Comput. Sci.*, vol. 19, no. 3, Nov. 2024, Art. no. 193327.

[29] Z. Hu, Y. Zhang, M. Xiao, W. Wang, F. Feng, and X. He, "Exact and efficient unlearning for large language model-based recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 10, pp. 5866–5877, Oct. 2025.

[30] C. Fan, J. Jia, Y. Zhang, A. Ramakrishna, M. Hong, and S. Liu, "Towards LLM unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond," in *Proc. 42nd Int. Conf. Mach. Learn.*, vol. 267, 2025, pp. 19762–19778.

[31] S. Hu, Y. Fu, S. Wu, and V. Smith, "Jogging the memory of unlearned llms through targeted relearning attacks," in *Proc. Neurips Safe Generative AI Workshop*, 2024, pp. 1–29. [Online]. Available: https://openreview.net/forum?id=YulEbrG99x
[32] F. Cheng, H. Li, F. Liu, R. van Rooij, K. Zhang, and Z. Lin, "Empowering LLMs with logical reasoning: A comprehensive survey," 2025, *arXiv:2502.15652*.
[33] M. Phuong and C. Lampert, "Towards understanding knowledge distillation," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5142–5151.
[34] S. Ghadimi and G. Lan, "Stochastic First{-} and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2341–2368, Jan. 2013.
[35] N. Houlsby et al., "Parameter-efficient transfer learning for NLP," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2790–2799.
[36] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2019.
[37] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.
[38] S. Black, L. Gao, P. Wang, C. Leahy, and S. Biderman, "GPT-Neo: Large scale autoregressive language modeling with mesh-tensorflow," Mar. 2021, vol. 5, no. 3, doi: 10.5281/zenodo.5297715.
[39] L. Gao et al., "The pile: An 800GB dataset of diverse text for language modeling," 2021, *arXiv:2101.00027*.
[40] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 188–197.
[41] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 649–657.
[42] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, Jan. 2016.
[43] S. Xu, W. Hua, and Y. Zhang, "OpenP5: An open-source platform for developing, training, and evaluating LLM-based recommender systems," in *Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2024, pp. 386–394.
[44] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.
[45] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, and J. Z. Kolter, "TOFU: A task of fictitious unlearning for LLMs," in *Proc. 1st Conf. Lang. Model.*, 2024, pp. 1–26. [Online]. Available: https://openreview.net/forum?id=B41hNBoWLo
[46] Y. R. Dong, H. Lin, M. Belkin, R. Huerta, and I. Vulić, "Unmemorization in large language models via self-distillation and deliberate imagination," 2024, *arXiv:2402.10052*.
[47] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 197–206.
[48] J. Liao et al., "LLaRA: Large language-recommendation assistant," in *Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2024, pp. 1785–1795.

Chenchen Tan received the bachelor's and bachelor's (Hons.) degrees in information technology from Deakin University, Burwood, VIC, Australia, in 2020 and 2021, respectively, and the B.E. degree from Southwest University, Chongqing, China, in 2021. She is currently pursuing the Ph.D. degree with the Faculty of Information Technology, Monash University, Clayton, VIC, Australia.

Her research interests include LLM unlearning and machine learning.

Xinghao Li received the bachelor's and bachelor's (Hons.) degrees in information technology from Deakin University, Burwood, VIC, Australia, in 2020 and 2021, respectively, and the B.E. degree from Southwest University, Chongqing, China, in 2021. He is currently pursuing the Ph.D. degree with the Faculty of Information Technology, Monash University, Clayton, VIC, Australia.

He has a profound interest in MLLM unlearning and machine learning.

Youyang Qu (Member, IEEE) received the B.S. degree in mechanical automation and the M.S. degree in software engineering from Beijing Institute of Technology, Beijing, China, in 2012 and 2015, respectively, and the Ph.D. degree from the School of Information Technology, Deakin University, Australia, in 2019.

He is currently a Research Scientist at Data61, CSIRO, Australia. Before joining CSIRO, he was a Research Fellow at Deakin University. He has over 50 publications, including high-quality journal articles and conference papers, such as IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, and *ACM Computing Surveys*. His research interests focus on machine learning, big data, IoT, blockchain, and corresponding security issues.

Cunjian Chen (Senior Member, IEEE) received the Ph.D. degree in computer science from West Virginia University, Morgantown, WV, USA, in 2014.

He is currently an Adjunct Lecturer with Monash University, Clayton, VIC, Australia, and a Research Fellow with the Monash Suzhou Research Institute, Suzhou, Jiangsu, China.

Dr. Chen received the Outstanding Area Chairs Award from ICME 2021 for recognition of his contributions. He has contributed to the academic community in various roles, including the Tutorial Chair of IJCB, the Area Chair of ICME, ICIP, ICASSP, and FG, and the Session Chair of ICASSP, ICME, and FG. He is currently an Associate Editor of *Neural Processing Letters* and has also served as an Associate Editor for *IET Image Processing*.

Shujie Cui received the B.Sc. degree from Shandong University, Jinan, China, in 2011, the joint M.Sc. degree in computer science from Shandong University and the University of Luxembourg, Esch-sur-Alzette, Luxembourg, in 2013, and the Ph.D. degree in computer science from the University of Auckland, Auckland, New Zealand, in 2019.

After obtaining her M.Sc. degree, she worked as a Research Associate at the Department of Computer Science, City University of Hong Kong, Hong Kong. She is currently a Lecturer with the Faculty of Information Technology, Monash University, Clayton, VIC, Australia. Her research interests include cloud computing, applied cryptography, and security and privacy.

Longxiang Gao (Senior Member, IEEE) received the Ph.D. degree in computer science from Deakin University, Australia, in 2014.

He is currently a Professor at Qilu University of Technology (Shandong Academy of Sciences), Jinan, China, and Shandong Computer Science Center (National Supercomputer Center), Jinan. He was a Senior Lecturer at the School of Information Technology, Deakin University, and a Post-Doctoral Research Fellow at the IBM Research and Development, Australia. He has over 100 publications, including patents, monographs, book chapters, and journal and conference papers. Some of his publications have been published in the top venue, such as IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING. He has been the Chief Investigator (CI) of more than 20 research projects (the total awarded amount is over U.S. 5 million), from pure research projects to contracted industry research. His research interests include fog/edge computing, blockchain, data analysis, and privacy protection.