TRACER: Token ReAssignment for Concept ERasure in Generative Recommendation

Ziheng Chen
Stony Brook University
albertchen1993pokemon@gmail.com

Jiali Cheng
University of Massachusetts Lowell
jiali_cheng@uml.edu

Zezhong Fan
Columbia University
zfan2274@gmail.com

Hadi Amiri
University of Massachusetts Lowell
hadi_amiri@uml.edu

Diyuan Wu
Institute of Science and Technology Austria
diyuan.wu@ist.ac.at

Gabriele Tolomei
Sapienza University of Rome
tolomei@di.uniroma1.it

Yang Zhang
National University of Singapore
zhangy@nus.edu.sg

Abstract

Generative recommendation formulates next-item prediction as autoregressive generation over semantic ID (SID) sequences derived from users' historical interactions, making modern recommender systems structurally similar to large language models (LLMs). As privacy and safety concerns grow, these systems increasingly require concept unlearning to remove sensitive or harmful concepts associated with items. However, existing LLM unlearning methods cannot be directly applied to generative recommendation. Unlike word tokens with explicit semantics, SIDs are abstract identifiers that are often shared by both forget and retain items, leading to severe conflicts between concept removal and recommendation utility preservation.

To address this challenge, we propose TRACER, an end-to-end concept unlearning framework based on token reassignment. Rather than directly suppressing shared SIDs, TRACER reassigns concept-related items to alternative tokens that better facilitate forgetting while minimizing side effects on retained items. We further introduce a coherence regularizer to preserve semantic consistency among retain items during unlearning. Experiments on real-world recommendation datasets demonstrate that TRACER effectively removes target concepts while substantially better preserving recommendation utility than existing unlearning baselines.

1 Introduction

Generative recommendation (GenRec) [49, 28, 40, 46] has emerged as a promising paradigm for sequential recommendation, demonstrating strong empirical performance across various domains [17, 30, 43, 39]. Unlike conventional recommendation systems, GenRec formulates recommendation as a sequence-to-sequence generation task and typically consists of two key components: item tokenization and autoregressive generation. Specifically, an item tokenizer assigns each item a semantic ID (SID), a sequence of discrete semantic tokens for indexing, while the generative model

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

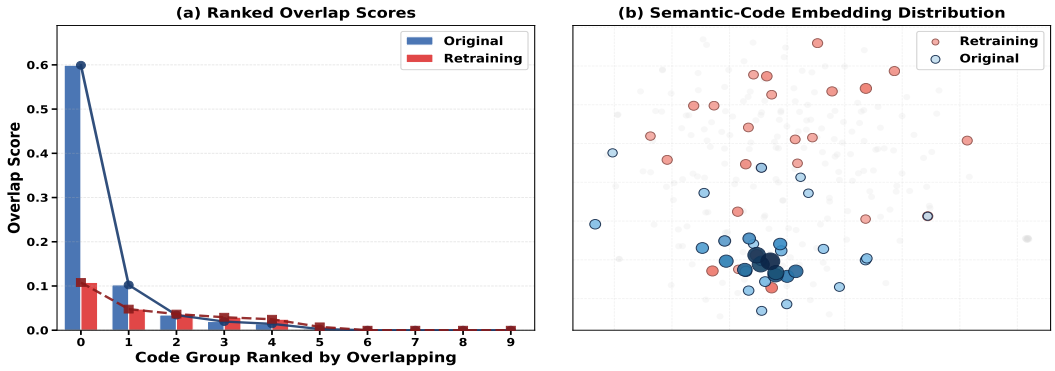


Figure 1: Retraining vs. Original GenRec model. (a): Original has highly overlapped semantic tokens, while Retraining mitigates overlap score significantly. (b): Original produces highly clustered embeddings, while Retraining produces dispersed embeddings. The overlap score is computed as the product of token frequencies in the forget and retain sets.

autoregressively predicts the next item from the SID sequence of purchased items. This design enables recommender systems to exploit the rich textual information associated with items and to inherit the representational and generative capabilities of Large Language Models (LLMs), showing strong empirical performance across diverse domains [59, 30, 49].

However, as GenRec becomes increasingly adopted, new concerns arise from sensitive information involved in model training, such as malicious descriptions [20, 23, 37], outdated information [10, 52, 16], and copyrighted text [18, 47, 13]. In this setting, concept unlearning in GenRec aims to remove the model's ability to recommend items associated with a target concept while preserving its recommendation utility on unrelated items. For example, removing a counterfeit sports brand requires the model to stop recommending items associated with that brand, while still preserving users' general sports preferences. Simply filtering concept-related items at generation time is insufficient, as it only modifies the final outputs while leaving the internal concept representation unchanged. Consequently, concept items in user histories can still influence predictions, and blocked items may be replaced with poorly matched alternatives, degrading recommendation utility.

Although concept unlearning remains underexplored in GenRec, it has been widely studied in LLMs [22, 27, 12, 14, 62]. Existing methods typically construct the forget set from data containing concept-related token sequences and suppress their likelihood during unlearning. However, this strategy cannot be directly applied to GenRec. Unlike language tokens, which are often interpretable lexical units and can help localize target concepts, SID tokens in GenRec are abstract *semantic identifiers (SIDs)* shared by items across both forget and retain sets. Therefore, a target concept cannot be reliably localized to a set of SID tokens. Besides, directly suppressing such shared tokens affects both the forget and the retain set, leading to severe gradient conflict during unlearning.

To characterize successful concept unlearning, we study the SID assignment of the *retrained* model with concept-free data in Figure 1. Specifically, we randomly select several brands as target concepts, remove their brand information from item descriptions, and filter out the corresponding brand-related interactions. Comparing the SID assignments of the original and retrained models reveals two key observations. First, SID overlap between forget and retain items drops sharply after retraining. Second, overlapped token embeddings become more dispersed after retraining than in the original model, suggesting that retraining weakens the concept-specific token structure. In contrast, unrelated tokens remain relatively stable. These observations motivate us to reshape SID assignments to reduce forget-retain overlap, rather than merely suppressing the likelihood of forget items.

To this end, we propose TRACER, a framework that achieves concept unlearning through differentiable token reassignment: rather than blindly suppressing forget items' likelihood, TRACER directly reshapes the SID structure to disentangle forget items from retain items. Specifically, TRACER relaxes the hard codebook lookup at each level into a soft distribution over candidate tokens, and introduces a lightweight learnable perturbation on top of the pretrained assignment. This design preserves the original tokenizer structure while allowing the unlearning objective to selectively redirect concept-related items toward tokens that retain items rarely use. Moreover, unlearning can disrupt recommendation coherence, especially when concept-related items appear as target items in the training data. TRACER preserves recommendation coherence by encouraging the recommender to generate semantically similar alternatives that do not contain the target concept. Although TRACER is primarily designed for concept unlearning, it can also be adapted to item-level and interaction-level unlearning.

The main contributions of this work are summarized as follows:

- We present the first study of concept unlearning in GenRec and reveal the limitations of directly applying existing LLM concept unlearning methods to this setting.
- We propose TRACER, an end-to-end GenRec unlearning framework that jointly optimizes concept-related token reassignment to satisfy the unlearning objective. We further introduce a coherence loss to preserve recommendation consistency after unlearning and provide theoretical analysis to justify the effectiveness of our algorithm.
- We conduct extensive experiments on real-world datasets and show the advantage of TRACER over existing methods.

2 Background and Preliminaries

2.1 Generative Recommendation

Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item sets. Each user u has an interaction sequence $\mathcal{H}_u = \{i_1, \ldots, i_T\}$ with $i_j \in \mathcal{I}$. The goal is to predict the next item i_{T+1} from $\mathcal{H}_u$. To leverage LLMs for generative recommendation, GenRec typically follows a two-stage paradigm.

Stage 1: Semantic Item Tokenization For each item $i \in \mathcal{I}$, a tokenizer $\mathcal{T}$ maps its textual description into a *Semantic Identifier (SID)*, denoted by

$$\boldsymbol{s}_i = \mathcal{T}(\boldsymbol{z}_i) = [s_i^1, \ldots, s_i^L], \tag{1}$$

where $\boldsymbol{z}_i$ is the item embedding produced by a frozen text encoder, and each s_i^l is a discrete semantic token selected from the l-th codebook. The codebooks are organized hierarchically: earlier levels capture coarse semantic information, while later levels encode increasingly fine-grained residual information. Concretely, residual quantization assigns one token at each level. Let $\boldsymbol{r}_i^1 = \boldsymbol{z}_i$ be the initial residual, and let $\mathcal{C}^l = \{\boldsymbol{c}_1^l, \ldots, \boldsymbol{c}_K^l\}$ denote the codebook at level l. The token assignment and residual update are given by

$$s_i^l = \arg\min_k \left\| \boldsymbol{r}_i^l - \boldsymbol{c}_k^l \right\|^2, \qquad \boldsymbol{r}_i^{l+1} = \boldsymbol{r}_i^l - \boldsymbol{c}_{s_i^l}^l. \tag{2}$$

Repeating this procedure for L levels yields the final SID $\boldsymbol{s}_i = [s_i^1, \ldots, s_i^L]$. The codebooks can be constructed in different ways depending on the tokenizer design. More generally, generative recommendation methods differ in whether SID tokenization is performed as a separate preprocessing step or learned jointly with the recommendation model.

Stage 2: Recommendation via Autoregressive Generation. For each user $u \in \mathcal{U}$, we observe a chronological interaction history $\mathcal{H}_u = (i_1, i_2, \ldots, i_{T-1}), i_t \in \mathcal{I}$. The goal of sequential recommendation is to predict the next item i_T conditioned on this history. In GenRec, each item is represented by its SID described in the prior section. Therefore the interaction history becomes a sequence of semantic token sequences $\boldsymbol{s}_{\mathcal{H}_u} = [\boldsymbol{s}_{i_1}, \boldsymbol{s}_{i_2}, \ldots, \boldsymbol{s}_{i_{T-1}}]$. The GenRec model parameterized by θ predicts the next-item SID autoregressively by maximizing its log-likelihood:

$$\log p_\theta(\boldsymbol{s}_{i_T} \mid \boldsymbol{x}_u) = \sum_{j=1}^{L} \log p_\theta(s_{i_T}^j \mid \boldsymbol{s}_{\mathcal{H}_u}, s_{i_T}^{<j}), \tag{3}$$

where $s_{i_T}^{<j} = (s_{i_T}^1, \ldots, s_{i_T}^{j-1})$ denotes the previously generated prefix of the target SID.

At inference time, the model generates the next SID, which realizes the next token prediction process, and serves as the basis for next-item recommendation. In particular, the generated SID can be one-to-one mapped back to an item via the codebook.

2.2 GenRec Concept Unlearning

Machine unlearning aims to remove the influence of a designated training subset from a trained model without retraining from scratch [29, 9, 19, 11, 21, 5]. Given a dataset $\mathcal{D}$ and a forget set $\mathcal{D}_f$, the retained data is $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$. The goal is to obtain an unlearned model $\mathcal{M}_{un}$ that eliminates

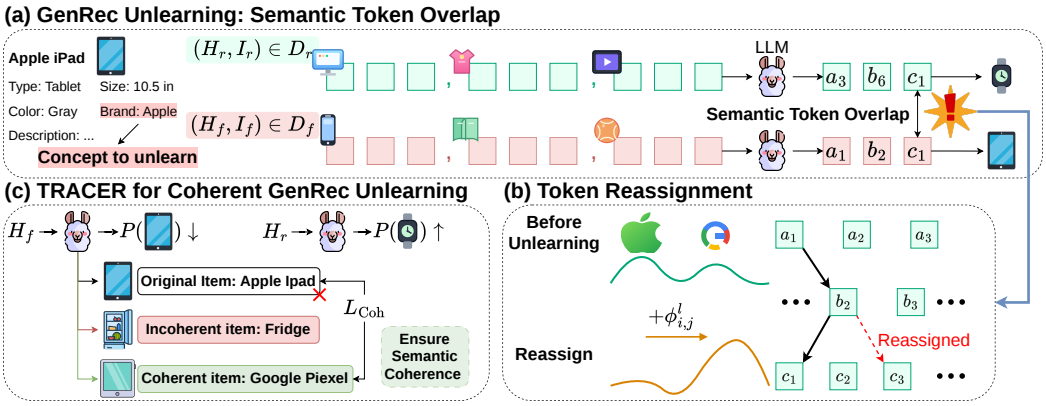


Figure 2: Overview of TRACER. **a) Semantic token overlap in GenRec unlearning:** removing conceptual knowledge from GenRec models is challenging, since many non-interpretable semantic tokens are shared by both forget and retain sets. **b) Token reassignment:** TRACER reassigns the semantic token of unlearning-affected items to the closest item of the same category, which preserves semantic coherence to the maximum extent. **c) Coherenct unlearning:** TRACER unlearn the target concept without affecting the recommendation ability and semantic coherence of the GenRec model.

the influence of $\mathcal{D}_f$ while preserving performance on $\mathcal{D}_r$ [20, 6, 8]. Since retraining on $\mathcal{D}_r$ to obtain $\mathcal{M}_{\theta^*}$ is costly, unlearning seeks to approximate this retrained model by updating the original model $\mathcal{M}_\theta$:

$$\mathcal{M}_\theta \xrightarrow{\mathcal{D}_f} \mathcal{M}_{un} \approx \mathcal{M}_{\theta^*}.$$

Definition (GenRec Concept Unlearning) Given a sensitive concept c, GenRec concept unlearning aims to: 1) prevent the model from recommending concept-related items, and 2) ensure that the presence of such items in a user's history $\mathcal{H}_u$ does not influence the prediction of future items.

Based on this definition, for a sensitive concept c, we define the forget set as $\mathcal{D}_f^c = \{(\mathcal{H}_u, i_t) \in \mathcal{D} : i_t \in \mathcal{I}_c\}$, where $\mathcal{I}_c \subseteq \mathcal{I}$ contains the items associated with c. The overall forget set is defined as the union over all target concepts $\mathcal{D}_f = \bigcup_c \mathcal{D}_f^c$, and retain set is given by $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$. For interactions in D_r, concept-related textual information should ideally be removed before *semantic item tokenization*, as if such information had never been present in the item descriptions. Following the approximate unlearning, we formulate unlearning as a trade-off between forgetting unwanted knowledge and preserving retain utility. Specifically, the unlearning objective combines a forget loss $\mathcal{L}_{\mathrm{F}}$ and a retain loss $\mathcal{L}_{\mathrm{R}}$:

$$\min_\theta \ \mathcal{L}_{\text{unlearn}}(\theta) = \mathcal{L}_{\mathrm{R}}(\theta) + \lambda \mathcal{L}_{\mathrm{F}}(\theta), \tag{4}$$

where λ controls the strength of forgetting. We introduce the details of these two losses in Section 3.2.

Challenges of GenRec Concept Unlearning Compared with traditional LLM concept unlearning [22, 2, 12, 14], GenRec concept unlearning faces two unique challenges. *First*, forget and retain items can substantially overlap in the SID space, so suppressing forget-item SIDs may also harm retained items sharing the same tokens. Since concept information cannot be cleanly removed once encoded into SIDs, we propose token reassignment to reduce forget–retain conflict while preserving retain utility. *Second*, unlearning may disrupt recommendation coherence when concept-related items appear as training targets [15, 7]. This motivates a coherence loss that encourages semantically consistent alternatives while suppressing the target concept. Given the above unique challenges, existing LLM unlearning method may fall short in GenRec concept unlearning.

3 Methodology

Our method, TRACER, consists of two modules: (1) semantic token reassignment, which alleviates overlap in semantic tokens (Figure 2 (b), § 3.1), and (2) coherence loss, which mitigates the degradation of semantic coherence (Figure 2 (c), § 3.2).

Overview and Motivation To better understand semantic token overlap–the major issue in GenRec unlearning, we investigate the oracle model retrained on concept-free data. We first filter out concept-related information in item descriptions. Then we retrain LETTER [49], a state-of-the-art GenRec model that jointly learns the tokenizer and the LLM recommender. Figure 1 shows that the resulting SID assignment exhibits lower forget-retain overlap and a weaker concept-related token structure. This observation suggests an intuitive strategy: reassign semantic tokens for concept-related items to satisfy the unlearning objective, while preserving the tokenization of the remaining items. Inspired by this finding, we first introduce a differentiable token reassignment mechanism in Section 3.1, and then present the unlearning objective in Section 3.2.

3.1 Flexible and Optimizable Token Reassignment

We construct a differentiable surrogate for the original tokenizer $\mathcal{T}$ to fulfill the motivation, which preserves the core assignment mechanism of $\mathcal{T}$ while making the token selection process trainable. Specifically, $\mathcal{T}$ can be viewed as a hierarchical token selector that maps each item to the nearest semantic token at each codebook level. We therefore introduce a soft tokenizer $\mathcal{T}_\phi$ to approximate $\mathcal{T}$ by replacing the discrete assignment in Equation 2 with a differentiable relaxation. Inspired by [41], the hard assignment operation that selects the nearest codeword can be relaxed into a temperature-controlled softmax over all codewords. Specifically, for item i at level l, the probability of assigning token k is defined as

$$q(s_i^l = k) = \frac{\exp\left(-\|\boldsymbol{r}_i^l - \boldsymbol{c}_k^l\|^2/\tau\right)}{\sum_{j=1}^{K} \exp\left(-\|\boldsymbol{r}_i^l - \boldsymbol{c}_j^l\|^2/\tau\right)}. \tag{5}$$

In this way, the token assignment is relaxed from a one-hot selection into a probability distribution over all codewords at each level. This approximation preserves the semantic structure of the original tokenizer by using the negative squared distance between the residual $\boldsymbol{r}_i^l$ and each codeword $\boldsymbol{c}_k^l$ as the assignment logit. More importantly, the soft tokenizer makes token assignment differentiable, allowing it to adapt to the unlearning objective through gradient-based optimization.

To enable end-to-end token reassignment while preserving the semantic hierarchy of the original tokenizer $\mathcal{T}$, we further introduce a learnable perturbation parameter ϕ over the codeword logit. Specifically, for the forget item i, the assignment probability at level l is defined by applying a softmax over the perturbed logits $-\|\boldsymbol{r}_i^l - \boldsymbol{c}_k^l\|^2 + \phi_{i,k}^l$, where $\phi_{i,k}^l$ captures the adjustment to the assignment preference for codeword $\boldsymbol{c}_k^l$. To avoid excessive deviation from the original tokenizer, we impose an L_1 regularization term on ϕ. In this way, the new tokenizer $\mathcal{T}_\phi$ inherits the semantic structure from $\mathcal{T}$, while allowing the learning objective to make small and targeted token reassignment:

$$q_\phi(s_i^l = k) = \frac{\exp\left((-\|\boldsymbol{r}_i^l - \boldsymbol{c}_k^l\|^2 + \phi_{i,k}^l)/\tau\right)}{\sum_{j=1}^{K} \exp\left((-\|\boldsymbol{r}_i^l - \boldsymbol{c}_j^l\|^2 + \phi_{i,j}^l)/\tau\right)}, \qquad \mathcal{L}_{\text{reg}} = \sum_{i,l,k} \left|\phi_{i,k}^l\right|. \tag{6}$$

The soft assignment is integrated into the LLM through a soft token embedding $\tilde{\mathbf{e}}_i^l$, defined as the expectation of token embeddings under q_ϕ: $\tilde{\mathbf{e}}_i^l = \sum_{k=1}^{K} q_\phi(s_i^l = k)\mathbf{e}_k^l$, where $\mathbf{e}_k^l$ is the LLM token embedding of codeword c_k^l. We apply reassignment only to target-concept items $i \in \mathcal{I}_c$, while preserving the original SIDs of all other items by fixing $\phi_{i,k}^l = 0$.

3.2 Concept Unlearning with Semantic Coherence

To preserve the model's behavior on data unrelated to the erased concept, we further optimize the standard generation objective on retain interactions $(\mathcal{H}_r, i_r) \in \mathcal{D}_r$. This retain loss encourages the model to maintain its original recommendation ability for retained items and mitigates unintended degradation caused by concept unlearning. Let $\mathcal{T}(i_r) = [s_{i_r}^1, \cdots, s_{i_r}^L]$ denote the SID sequence of the retain target item i_r. We define

$$\mathcal{L}_R = -\sum_{(\mathcal{H}_r, i_r) \in \mathcal{D}_r} \sum_{\ell=1}^{L} \log p_\theta\left(s_{i_r}^\ell \mid \mathcal{T}(\mathcal{H}_r),\, s_{i_r}^{<\ell}\right). \tag{7}$$

Following the forgetting objective, we erase the target concept by reducing the likelihood of recommending concept-related items. For each forget interaction $(\mathcal{H}_f, i) \in \mathcal{D}_f$, where $i \in \mathcal{I}_c$, we suppress

the generation of its original SID sequence $\mathcal{T}(i) = [s_i^1, \cdots, s_i^L]$ via a token-level objective:

$$\mathcal{L}_F = \sum_{(\mathcal{H}_f, i) \in \mathcal{D}_f} \sum_{\ell=1}^{L} \log p_\theta\big(s_i^\ell \mid \mathcal{T}(\mathcal{H}_f),\, s_i^{<\ell}\big)\,. \tag{8}$$

Minimizing this term discourages the model from assigning high probability to the original SID tokens of concept-related target items, thereby reducing their autoregressive generation likelihood.

However, directly suppressing the generation probability of a concept-related item i may also penalize semantic tokens that are not specific to the target concept. This can disrupt the learned semantic structure and produce recommendations that are poorly aligned with the user history. To preserve recommendation coherence, we further encourage the model to generate retain items that are semantically similar to i under the same history $\mathcal{H}_u$. Specifically, for each forget item i, we construct a positive set $\mathcal{P}(i)$ from the retain set based on item-description similarity. We compute the similarity using the frozen encoder representation $\boldsymbol{z}_i$ before token assignment, as defined in Section 2.1. We select the top-K most similar retain items, so $|\mathcal{P}(i)| = K$. Given item $i_p \in \mathcal{P}(i_T)$ with SIDs $\mathcal{T}(i_p) = [s_p^1, \cdots, s_p^L]$, we define the coherence loss as

$$\mathcal{L}_{\text{Coh}} = -\frac{1}{K} \sum_{(\mathcal{H}_f, i) \in \mathcal{D}_f} \sum_{i_p \in \mathcal{P}(i_T)} \sum_{\ell=1}^{L} \log p_\theta\big(s_p^\ell \mid \mathcal{T}(\mathcal{H}_f),\, s_p^{<\ell}\big)\,. \tag{9}$$

The overall loss for the TRACER objective is a weighted combination of these components:

$$\mathcal{L} = \mathcal{L}_{\text{R}} + \lambda_1 \mathcal{L}_{\text{F}} + \lambda_2 \mathcal{L}_{\text{Coh}} + \lambda_3 \mathcal{L}_{\text{reg}} \tag{10}$$

3.3 Selective Updating of Perturbation Parameters

Since the reassignment parameter ϕ controls the token distribution of forget items, updating all entries of ϕ may unnecessarily perturb concept-irrelevant token assignments and degrade recommendation utility. To make the reassignment more targeted, we selectively update ϕ. In particular, given item $i \in \mathcal{I}_c$ and token t, we define the mask

$$M_{i,t}^\ell = \mathbf{1}\left[\rho_\ell(t) > \bar{\rho}_{i,\ell}\right] \cdot \mathbf{1}\left[\nabla_{\phi_{i,b}^\ell} \mathcal{L}_F > 0\right], \quad \rho_\ell(t) := \frac{1}{|\mathcal{D}_r|} \sum_{j \in \mathcal{D}_r} \mathbf{1}\left[t_j^\ell = k\right], \quad \bar{\rho}_{i,\ell} = \mathbb{E}_{q_\phi} \rho_\ell(k) \tag{11}$$

Here, $\rho_\ell(t)$ measures how frequently token t appears in the retain set, while $\bar{\rho}_{i,\ell}$ denotes the expected retain-token frequency under the current assignment distribution q_ϕ. The key intuition is that tokens heavily shared by forget and retain items are more likely to induce gradient conflict and concept leakage. Therefore, we restrict the update of ϕ to entries associated with high-overlap tokens according to Eq. 11, while leaving irrelevant token assignments unchanged.

3.4 Theoretical Analysis

We define Ω as the average token overlap between a forget item $i \in \mathcal{F}$ and a retain item $j \in \mathcal{R}$, averaged over all codebook levels. Specifically,

$$\Omega = \mathbb{E}_{i \sim \mathcal{F},\, j \sim \mathcal{R}}\left[\omega(i,j)\right], \qquad \omega(i,j) = \frac{1}{L} \sum_{\ell=1}^{L} \sum_{k \in \mathcal{T}^\ell} \mathbf{1}[s_i^\ell = k] \mathbf{1}[s_j^\ell = k]. \tag{12}$$

Here, $\mathcal{T}^\ell$ denotes the token set at codebook level ℓ, and s_i^ℓ is the token assigned to item i at level ℓ.

Proposition 1 *Suppose Assumption 3 holds. Let*

$$\theta_+ := \theta - \eta \nabla_\theta \mathcal{L}_{\text{unlearn}}(\theta)$$

be the model parameters after one gradient descent step on the unlearning objective. Then the retain-loss change satisfies

$$\mathcal{L}_R(\theta_+) - \mathcal{L}_R(\theta) \leq \eta\alpha \frac{\xi^2}{L} \Omega + C_U. \tag{13}$$

Furthermore, if for any fixed token t and any conditioning sequences $\mathcal{T}, \mathcal{T}'$, the token-level gradients satisfy

$$\langle \nabla_\theta \ell(\theta; t \mid \mathcal{T}), \nabla_\theta \ell(\theta; t \mid \mathcal{T}') \rangle \geq \gamma > 0,$$

then

$$\mathcal{L}_R(\theta_+) - \mathcal{L}_R(\theta) \geq \eta\alpha \frac{\gamma}{L} \Omega + C_L. \tag{14}$$

Here, C_U and C_L collect residual terms and second-order optimization remainders.

We defer the proof and further discussion of to Appendix B.

Lemma 2 *Under Assumption 5(Appendix), fix the model parameter θ and let*

$$\phi_+ = \phi - \eta_\phi M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi),$$

where M is defined in Eq. 11 and $\odot$ denotes element-wise product. For sufficiently small η_ϕ, we have

$$\Omega(\phi_+) - \Omega(\phi) \leq 0, \qquad \mathcal{L}_F(\phi_+) - \mathcal{L}_F(\phi) \leq 0.$$

This suggests that token reassignment can help mitigate retain-loss degradation by suppressing overlap-mediated forget-retain interference during unlearning.

4 Experiments

Datasets and preprocessing. We evaluate our TRACER on three real-world Amazon subsets, namely *Industrial_and_Scientific*, *Sports_and_Outdoors*, and *Toys_and_Games*, which cover diverse recommendation domains. Following [30], we remove users and items with fewer than five interactions, and then split each dataset chronologically into train (80%), validate (10%), and test (10%).

Concept unlearning setup. In the main text, we use brand as the target concept to erase. Brand information is explicitly provided in the metadata and naturally shared across multiple items, making it well-suited for concept-level forgetting. For each dataset, we randomly select 5% of brands as target concepts and construct the forget set from interactions whose target items are associated with these brands, which account for approximately 10% of the full dataset. To further evaluate the versatility of our framework beyond a single metadata concept, we additionally report appendix experiments on price band, where valid item prices are discretized into several coarse-grained groups, and the same unlearning protocol is applied.

Models. We evaluate our framework on a diverse collection of generative recommendation models spanning both separate-training and joint-training paradigms. The separate-training models include *MiniOneRec* [30], *Tiger* [43], *P5-SID* [49] while the joint-training models include *LETTER* [49] and *ETEGRec* [36]. These models differ substantially in architecture and the way recommendation signals are integrated with language modeling.

Backbones. MiniOneRec is built on Qwen2.5-7B [53], whereas Tiger, P5-SID, LETTER, and ETEGRec are all based on the T5 [42]. This backbone diversity—spanning different model scales, generation paradigms, and training strategies—allows us to evaluate whether TRACER generalizes across fundamentally different architectures.

Baselines. Since GenRec models rely on LLMs to generate semantic tokens, we compare TRACER with representative LLM concept unlearning methods, including ELM [22], RMU [32], RepNoise [45], and FLAT [50]. In addition, we include recent unlearning methods designed to mitigate retain-forget conflicts, such as BLUR [44] and PISCES [27].

Evaluation metrics. We evaluate methods from three complementary dimensions: *utility*, *forgetting effectiveness*, and *recommendation coherence*. Implementation details are provided in the appendix.

1. Utility. We measure recommendation quality on the retain set using standard ranking metrics, including *NDCG*, *MRR*, and *HR*. These metrics assess whether the model preserves its ranking ability on non-forgotten data after unlearning.

Table 1: Unlearning results on **Industrial & Scientific** and **Sports & Outdoors** using MOR and LETTER as backbones. Bold = best among unlearning methods; underline = second best. Retain metrics (↑): higher is better; Forget metrics (↓): lower is better; Sim. (↑): higher is better.

Method	Forget ↓		Retain ↑		Sem. ↑	Forget ↓		Retain ↑		Sem. ↑
	HR@5	NDC@5	HR@5	NDC@5	Sim.	HR@5	NDC@5	HR@5	NDC@5	Sim.
	Industrial & Scientific, MOR					**Sports & Outdoors, MOR**				
Original	0.170±.003	0.110±.002	0.126±.003	0.100±.002	0.851±.005	0.090±.002	0.063±.002	0.078±.002	0.067±.002	0.840±.005
ELM	0.143±.004	0.091±.003	0.084±.004	0.068±.003	0.824±.006	0.074±.003	0.055±.002	0.051±.004	0.044±.002	0.812±.006
RMU	0.151±.005	0.094±.003	0.061±.005	0.051±.004	0.730±.008	0.078±.004	0.054±.003	0.037±.005	0.033±.003	0.720±.007
REPNoise	0.131±.004	0.081±.003	0.068±.004	0.056±.003	0.630±.008	0.069±.003	0.044±.002	0.058±.003	0.037±.002	0.626±.007
FLAT	0.119±.003	0.074±.002	0.119±.003	0.091±.003	0.710±.007	0.059±.002	0.039±.002	0.074±.003	0.062±.002	0.693±.006
BLUR	0.139±.004	0.087±.003	0.120±.003	0.095±.003	0.680±.006	0.073±.003	0.047±.002	0.072±.003	0.065±.002	0.666±.006
PISCES	0.101±.003	0.066±.002	0.115±.003	0.090±.002	0.790±.005	0.052±.002	0.038±.001	0.072±.002	0.063±.002	0.773±.005
TRACER	**0.080**±.002	**0.058**±.001	**0.123**±.002	**0.097**±.002	**0.828**±.004	**0.044**±.001	**0.034**±.001	**0.078**±.002	**0.066**±.002	**0.825**±.004
	Industrial & Scientific, LETTER					**Sports & Outdoors, LETTER**				
Original	0.192±.004	0.121±.003	0.128±.003	0.103±.002	0.832±.005	0.103±.002	0.071±.002	0.081±.002	0.069±.002	0.821±.005
ELM	0.167±.004	0.104±.003	0.086±.004	0.070±.003	0.821±.006	0.082±.003	0.057±.002	0.053±.004	0.057±.002	0.802±.006
RMU	0.178±.005	0.108±.004	0.059±.005	0.049±.004	0.730±.007	0.085±.004	0.057±.003	0.061±.004	0.031±.003	0.725±.007
REPNoise	0.145±.004	0.090±.003	0.070±.004	0.057±.003	0.690±.008	0.070±.003	0.055±.002	0.062±.004	0.050±.002	0.681±.007
FLAT	0.148±.004	0.091±.003	0.113±.003	0.089±.003	0.710±.007	0.078±.003	0.054±.002	0.071±.003	0.066±.002	0.686±.006
BLUR	0.164±.004	0.101±.003	0.123±.003	0.096±.003	0.720±.006	0.079±.003	0.056±.002	0.076±.003	0.065±.002	0.713±.006
PISCES	0.125±.003	0.077±.002	0.117±.003	0.092±.002	0.807±.005	0.061±.002	0.045±.002	0.077±.002	0.063±.002	0.782±.005
TRACER	**0.102**±.002	**0.068**±.002	**0.124**±.002	**0.098**±.002	**0.822**±.004	**0.054**±.001	**0.041**±.001	**0.078**±.002	**0.067**±.002	**0.821**±.003

2. Unlearning Effectiveness. We evaluate forgetting on the forget set using the same ranking metrics, *NDCG*, *MRR*. Lower scores on the forget set indicate more effective removal of the target information.

3. Recommendation Coherence. To evaluate whether the model remains semantically coherent after unlearning, we report the semantic similarity between the generated items and the original target items. The similarity is computed using item descriptions encoded by the frozen encoder described in Section 2.1. This metric captures whether the model can still produce conceptually consistent recommendations while suppressing the target concept.

5 Results

We evaluate TRACER from four perspectives: 1) unlearning effectiveness; 2) flexibility with different GenRec backbones; 3) robustness to attacks; and 4) the contributions of each component of TRACER.

TRACER achieves effective and targeted unlearning Across multiple GenRec models, TRACER consistently achieves the strongest unlearning effectiveness while preserving recommendation utility (see Table 1). Specifically on MOR, TRACER obtains the best HR@5, HR@10, NDCG@5, NDCG@10, MRR scores, showing that it most effectively removes the target concept. On Forget HR@5, TRACER outperforms PISCES, the best performing baseline, more than 20%. On Retain HR@5, TRACER outperforms all unlearning methods. We observe similar patterns on other metrics and other GenRec models. Among the baselines, BLUR and FLAT explicitly consider gradient conflict in their formulation. We discovery that BLUR, a method optimized for retain-forget conflict, has poor unlearning effectiveness. This is because BLUR explicitly prioritize retain performance over forget effectiveness.

TRACER preserves recommendation ability and coherence. Meanwhile, TRACER maintains nearly the same utility as the original model on the retain split. On average, it preserves 97.1% of the original model performance across HR, NDCG, and MRR. In particular, its HR@10 reaches 0.159 compared with 0.162 for the original model, retaining approximately 98.0% of the original HR@10. TRACER is also the best or tied-best unlearning method in 24 out of 25 retain-side comparisons, suggesting that it selectively suppresses forgotten behavior rather than broadly damaging the recommender. The coherence results further show that TRACER does not achieve forgetting by simply disrupting generation quality. TRACER reaches an average semantic similarity of 0.834, close to the original model's 0.852 and higher than all evaluated unlearning baselines. Its relative coherence drop is only about 2.0%, indicating that TRACER preserves the model's semantic structure while weakening the information targeted for removal. In summary, these results demonstrate that TRACER can unlearn conceptual knowledge from GenRec models in a targeted and effective manner. We show the item-level unlearning in Appendix D.4.

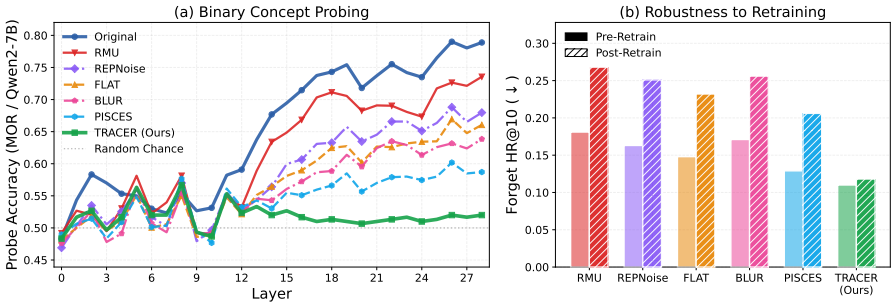


Figure 3: Probing Results on MOR.

TRACER Enables Backbone-Agnostic GenRec Unlearning We conduct experiments across different backbones, including both hard-tokenization and end-to-end GenRec models. On TIGER and MOR, TRACER demonstrates superior performance, achieving 16.5% lower targeting scores than PISCES while preserving 96.8% of the original model utility on unrelated samples across HR, NDCG, and MRR. For end-to-end backbones such as LETTER and ETEGRec, TRACER also consistently achieves the best performance in both concept erasing and utility preservation.

TRACER is Efficient Moreover, TRACER runs 33.5% and 36.7% faster than PISCES and REPNoise, respectively. TRACER also requires the least time to achieve effective unlearning compared with existing methods. Additional experimental results and efficiency analysis are provided in Appendix D.2.

TRACER is Robustness to Attacks This setting tests whether the erased concept can be relearned from concept-related interactions under the post-unlearning tokenizer. Following [32], we evaluate the robustness of TRACER against fine-tuning attacks. We construct a stronger fine-tuning set in which 20% of the target items are associated with target concept. For these items, we use their post-reassignment SIDs, corresponding to the deployed SID space after unlearning. As shown in Figure 3 (right), TRACER remains robust against fine-tuning attacks even after 2000 fine-tuning steps.

Concept Probing Analysis To estimate whether the erased concept remains in the model's internal representations, we conduct a probing analysis following [22]. This analysis examines whether the model still encodes the target concept even if it no longer explicitly generates or recommends it. Specifically, we treat interactions in $\mathcal{D}_f$ as positive samples, and randomly sample the same number of interactions from D_r^c as negative samples. We then train a linear probe on the hidden representations from each model layer and report the probing accuracy across layers. As shown in Figure 3(left), TRACER and PISCES consistently achieve low probing accuracies, only slightly above random guessing, across all layers for brand-related items, indicating that the target concept is effectively removed from the internal representations.

Ablation Study We conduct ablation studies to examine the contribution of each component of TRACER. Figure 4 shows the impact of removing the reassignment module across different GenRec backbones, demonstrating that it substantially degrades both model utility and unlearning effectiveness. Figure 6(Appendix) studies the effect of the hyperparameter λ in Eq. 10. The results show that $\mathcal{L}_{\text{Coh}}$ is crucial for maintaining recommendation coherence, while the regularizer is important for preserving model utility. More fine-grained ablations and hyperparameter analyses are provided in Appendix D.5.

Limitations. TRACER is designed for concept unlearning in generative recommendation and therefore assumes that sensitive information is associated with semantic concepts rather than directly encoded in item identifiers (SIDs). Extending the framework to settings where personal information is embedded within SIDs or other latent representations remains an important direction for future work. In addition, while TRACER reassigns target concept tokens to tokens that are infrequently used by retain items, the potential interaction between this reassignment strategy and the popularity distribution of items has not been systematically investigated. In particular, low-popularity retain items whose tokens overlap with reassignment targets may be disproportionately affected. Although our experiments demonstrate strong overall recommendation performance, a more detailed analysis

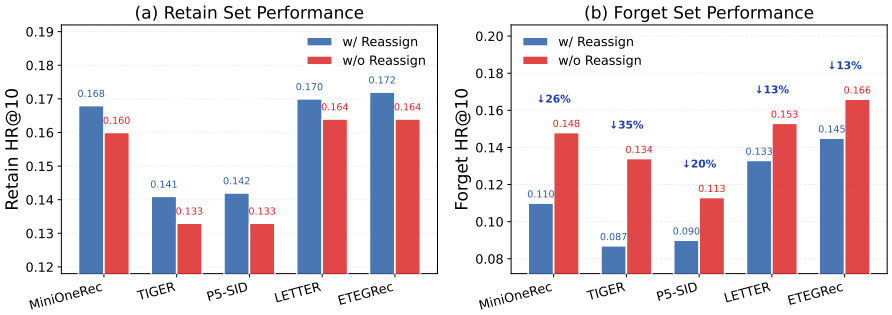


Figure 4: Effect of Reassignment.

of fairness and popularity-related effects would provide a more comprehensive understanding of the proposed approach.

6 Conclusion

We introduce the first concept unlearning task for generative recommendation (GenRec), where the goal is to remove target concept information while preserving recommendation utility and semantic coherence. To address this task, we propose TRACER, a token reassignment framework that learns to reassign GenRec semantic tokens for effective unlearning, while preserving recommendation coherence. Extensive experiments demonstrate the effectiveness [61], flexibility, and robustness of TRACER, which outperforms existing unlearning baselines.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for their insightful comments and constructive suggestions, which helped improve the quality and clarity of this work. This material is in part based upon work supported by the National Science Foundation under Grant No. 2541273. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This work was partially supported by the following project: GHOST – Protecting User Privacy from Community Detection in Social Networks (B83C24007070005), funded by Sapienza University of Rome through the "Progetti di Ricerca Grandi" program.

References

[1] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *IEEE Symposium on Security and Privacy (SP)*.

[2] Pengfei Cao, Chenhao Wang, Zhitao He, Hongbang Yuan, Jiachun Li, Yubo Chen, Kang Liu, Jun Zhao, et al. 2024. Rwku: Benchmarking real-world knowledge unlearning for large language models. *Advances in Neural Information Processing Systems* 37 (2024), 98213–98263.

[3] Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[4] Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. 2022. Recommendation unlearning. In *Proceedings of the ACM web conference 2022*. 2768–2777.

[5] Junkai Chen, Yuhao He, Qianshan Wei, Junxiang You, Jingwen Shao, Junkai Lin, Zhongkai Yue, Xiaotian Ye, Zhengbo Jiao, Jiali Cheng, et al. 2026. Open-MMUnlearning: Unifying Methods and Evaluation for MLLM Unlearning. *arXiv preprint arXiv:2610.10358* (2026).

[6] Ziheng Chen, Jiali Cheng, Hadi Amiri, Kaushiki Nag, Lu Lin, Sijia Liu, Gabriele Tolomei, and Xiangguo Sun. 2025. FROG: Fair Removal on Graph. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 415–424.

[7] Ziheng Chen, Jiali Cheng, Zezhong Fan, Hadi Amiri, Yunzhi Yao, Xiangguo Sun, and Yang Zhang. 2026. CURE: Circuit-Aware Unlearning for LLM-based Recommendation. *arXiv preprint arXiv:2604.04982* (2026).

[8] Ziheng Chen, Jin Huang, Jiali Cheng, Yuchan Guo, Mengjie Wang, Lalitesh Morishetti, Kaushiki Nag, and Hadi Amiri. 2025. FUTURE: Flexible Unlearning for Tree Ensemble. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 4680–4684.

[9] Jiali Cheng and Hadi Amiri. 2024. MU-bench: A multitask multimodal benchmark for machine unlearning. *arXiv preprint arXiv:2406.14796* (2024).

[10] Jiali Cheng and Hadi Amiri. 2025. MultiDelete for Multimodal Machine Unlearning. In *Computer Vision – ECCV 2024*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Springer Nature Switzerland, Cham, 165–184.

[11] Jiali Cheng and Hadi Amiri. 2025. Speech Unlearning. In *Interspeech 2025*. 3209–3213. doi:10.21437/Interspeech.2025-2412

[12] Jiali Cheng and Hadi Amiri. 2025. Tool Unlearning for Tool-Augmented LLMs. In *Forty-second International Conference on Machine Learning*.

[13] Jiali Cheng and Hadi Amiri. 2026. *Machine Unlearning Across Tasks and Modalities*. Springer Nature Switzerland, Cham, 227–240. doi:10.1007/978-3-032-17282-2_15

[14] Jiali Cheng, Ziheng Chen, Chirag Agarwal, and Hadi Amiri. 2026. Toward Understanding Unlearning Difficulty: A Mechanistic Perspective and Circuit-Guided Difficulty Metric. *arXiv preprint arXiv:2601.09624* (2026).

[15] Jiali Cheng, George Dasoulas, Huan He, Chirag Agarwal, and Marinka Zitnik. 2023. GNNDelete: A General Strategy for Unlearning in Graph Neural Networks. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=X9yCkmT5Qrl

[16] Jiali Cheng, Rui Pan, and Hadi Amiri. 2026. Investigating Tool-Memory Conflicts in Tool-Augmented LLMs. *arXiv preprint arXiv:2601.09760* (2026).

[17] Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965* (2025).

[18] Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238* (2023).

[19] Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. 2025. Towards llm unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. *arXiv preprint arXiv:2502.05374* (2025).

[20] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. 2024. SalUn: Empowering Machine Unlearning via Gradient-based Weight Saliency in Both Image Classification and Generation. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=gn0mIhQGNM

[21] Xingyu Fu, Siyi Liu, Yinuo Xu, Pan Lu, Guangqiuse Hu, Tianbo Yang, Taran Anantasagar, Christopher Shen, Yikai Mao, Yuanzhe Liu, et al. 2025. Learning human-perceived fakeness in ai-generated videos via multimodal llms. *arXiv preprint arXiv:2509.22646* (2025).

[22] Rohit Gandikota, Sheridan Feucht, Samuel Marks, and David Bau. 2024. Erasing conceptual knowledge from language models. *arXiv preprint arXiv:2410.02760* (2024).

[23] Rohit Gandikota, Joanna Materzyńska, Jaden Fiotto-Kaufman, and David Bau. 2023. Erasing Concepts from Diffusion Models. In *Proceedings of the 2023 IEEE International Conference on Computer Vision*.

[24] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9304–9312.

[25] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. 2019. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030* (2019).

[26] Neha R. Gupta, Vittorio Orlandi, Chia-Rui Chang, Tianyu Wang, Marco Morucci, Pritam Dey, Thomas J. Howell, Xian Sun, Angikar Ghosal, Sudeepa Roy, Cynthia Rudin, and Alexander Volfovsky. 2025. dame-flame: A Python Package Providing Fast Interpretable Matching for Causal Inference. *Journal of Statistical Software* 113, 2 (2025), 1–26. doi:10.18637/jss.v113.i02

[27] Yoav Gur-Arieh, Clara Haya Suslik, Yihuai Hong, Fazl Barez, and Mor Geva. 2025. Precise in-parameter concept erasure in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 18997–19017.

[28] Ruidong Han, Bin Yin, Shangyu Chen, He Jiang, Fei Jiang, Xiang Li, Chi Ma, Mincong Huang, Xiaoguang Li, Chunzhen Jing, et al. 2025. Mtgr: Industrial-scale generative recommendation framework in meituan. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 5731–5738.

[29] Jinghan Jia, Yihua Zhang, Yimeng Zhang, Jiancheng Liu, Bharat Runwal, James Diffenderfer, Bhavya Kailkhura, and Sijia Liu. 2024. SOUL: Unlocking the Power of Second-Order Optimization for LLM Unlearning. *arXiv preprint arXiv:2404.18239* (2024).

[30] Xiaoyu Kong, Leheng Sheng, Junfei Tan, Yuxin Chen, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. 2025. Minionerec: An open-source framework for scaling generative recommendation. *arXiv preprint arXiv:2510.24431* (2025).

[31] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11523–11532.

[32] Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218* (2024).

[33] Luyun Lin, Lixing Lin, Zhen Zhang, Moxuan Zheng, and Yiqing Wang. 2026. A Volume-Price-Adjusted MACD Trading Strategy with Sensitivity Calibration for US Equity Indices. *arXiv preprint arXiv:2604.26063* (2026).

[34] Luyun Lin and Yiqing Wang. 2025. Shap stability in credit risk management: A case study in credit card default model. *Risks* 13, 12 (2025), 238.

[35] Lixing Lin, Juli You, Yue Li, Luyun Lin, Yiqing Wang, Zhen Zhang, and Moxuan Zheng. 2026. Reflect-Guard: Enhancing LLM Safeguards against Adversarial Prompts via Logical Self-Reflection. *arXiv preprint arXiv:2605.24834* (2026).

[36] Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. 2025. Generative recommender with end-to-end learnable item tokenization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 729–739.

[37] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. 2024. Rethinking machine unlearning for large language models. *arXiv preprint arXiv:2402.08787* (2024).

[38] Yue Liu, Zhiyuan Cheng, and Longying Lai. 2026. Improving the Completeness and Comparability of Segment Disclosures: A Large Language Model Approach. arXiv:2605.23924 [cs.CL] https://arxiv.org/abs/2605.23924

[39] Yuanzhe Liu, Jingyuan Zhu, Yuchen Mo, Gen Li, Xu Cao, Jin Jin, Yifan Shen, Zhengyuan Li, Tianjiao Yu, Wenzhen Yuan, et al. 2026. Palm: Progress-aware policy learning via affordance reasoning for long-horizon robotic manipulation. *arXiv preprint arXiv:2601.07060* (2026).

[40] Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, Honghui Bao, Jinghao Zhang, Wuchao Li, Pengfei Zheng, Xiangyu Wu, et al. 2025. Onerec-think: In-text reasoning for generative recommendation. *arXiv preprint arXiv:2510.11639* (2025).

[41] Ana Lucic, Harrie Oosterhuis, Hinda Haned, and Maarten De Rijke. 2022. FOCUS: Flexible optimizable counterfactual explanations for tree ensembles. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. 5313–5322.

[42] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.

[43] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems* 36 (2023), 10299–10315.

[44] Hadi Reisizadeh, Jinghan Jia, Zhiqi Bu, Bhanukiran Vinzamuri, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, Sijia Liu, and Mingyi Hong. 2026. Blur: A bi-level optimization approach for llm unlearning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. 7043–7058.

[45] Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. 2024. Representation noising: A defence mechanism against harmful finetuning. *Advances in Neural Information Processing Systems* 37 (2024), 12636–12676.

[46] Yifan Shen, Yuanzhe Liu, Jingyuan Zhu, Xu Cao, Xiaofeng Zhang, Yixiao He, Wenming Ye, James Rehg, and Ismini Lourentzou. 2026. Fine-grained preference optimization improves spatial reasoning in vlms. *Advances in Neural Information Processing Systems* 38 (2026), 17929–17960.

[47] Weijia Shi, Jaechan Lee, Yangsibo Huang, Sadhika Malladi, Jieyu Zhao, Ari Holtzman, Daogao Liu, Luke Zettlemoyer, Noah A. Smith, and Chiyuan Zhang. 2025. MUSE: Machine Unlearning Six-Way Evaluation for Language Models. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=TArmA033BU

[48] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. 2022. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 303–319.

[49] Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2400–2409.

[50] Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Shah, Yujia Bao, Yang Liu, and Wei Wei. 2024. Llm unlearning via loss adjustment with only forget data. In *The Thirteenth International Conference on Learning Representations*.

[51] Yaxuan Wang, Jiaheng Wei, Chris Yuhao Liu, Jinlong Pang, Quan Liu, Ankit Parag Shah, Yujia Bao, Yang Liu, and Wei Wei. [n. d.]. Llm unlearning via loss adjustment with only forget data, 2024. *URL https://arxiv. org/abs/2410.11143* ([n. d.]).

[52] Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for llms: A survey. *arXiv preprint arXiv:2403.08319* (2024).

[53] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025).

[54] Shuai Yuan, Xian Sun, Hannah Kim, Shuzhi Yu, and Carlo Tomasi. 2022. Optical Flow Training under Limited Label Budget via Active Learning. In *European Conference on Computer Vision (ECCV)*. Springer, 410–427.

[55] Johnny R. Zhang, Gaoyuan Du, Qianyi Sun, Shiqi Wang, Jiaxuan Li, and Xian Sun. 2026. Carbon-Aware Compute–Power Scheduling for AI Data Centers with Microgrid Prosumer Operations. arXiv:2605.03751 [cs.CE] https://arxiv.org/abs/2605.03751

[56] Yunqi Zhang and Shaileshh Bojja Venkatakrishnan. 2023. Kadabra: Adapting kademlia for the decentralized web. In *International Conference on Financial Cryptography and Data Security*. Springer, 327–345.

[57] Yunqi Zhang and Shaileshh Bojja Venkatakrishnan. 2025. Honeybee: Byzantine Tolerant Decentralized Peer Sampling with Verifiable Random Walks. In *Proceedings of the Twenty-sixth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 321–330.

[58] Yunqi Zhang and Shaileshh Bojja Venkatakrishnan. 2023. Rethinking incentive in payment channel networks. In *2023 IEEE 43rd International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE, 61–66.

[59] Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, et al. 2025. OneRec Technical Report. *arXiv preprint arXiv:2506.13695* (2025).

[60] Chunzheng Zhu, Jianxin Lin, Guanghua Tan, Ningbo Zhu, Kenli Li, Chunlian Wang, and Shengli Li. 2024. Advancing Ultrasound Medical Continuous Learning with Task-Specific Generalization and Adaptability. In *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 3019–3025.

[61] Chunzheng Zhu, Yangfang Lin, Shen Chen, Yijun Wang, and Jianxin Lin. 2026. Medeyes: Learning dynamic visual focus for medical progressive diagnosis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 13916–13924.

[62] Chunzheng Zhu, Yangfang Lin, Jialin Shao, Jianxin Lin, and Yijun Wang. 2025. Pathology-Aware Prototype Evolution via LLM-Driven Semantic Disambiguation for Multicenter Diabetic Retinopathy Diagnosis. In *Proceedings of the 33rd ACM International Conference on Multimedia*. 9196–9205.

A Broader Impact

This work addresses the growing need for *machine unlearning* in recommender systems, enabling platforms to comply with privacy regulations such as GDPR's right to be forgotten and CCPA by selectively removing the influence of specific items or brands from a trained model without full retraining. Our framework empowers content providers and users to exercise meaningful control over how their data is represented in recommendation models, reducing the risk of unwanted brand association or continued promotion of recalled, delisted, or sensitive products. We emphasize that unlearning should be viewed as one component of a broader responsible-AI strategy rather than a standalone guarantee. Our experiments are conducted on English-language Amazon product review datasets; the effectiveness of SID-based unlearning in multilingual or cross-domain settings remains an interesting direction for future work [54, 26, 55].

B Theoretical Analysis

B.1 Theoretical results

We first define $\mathcal{F}$ be the forget set, $\mathcal{R}$ be the retain set and Ω as the average token overlap between a forget item $i \in \mathcal{F}$ and a retain item $j \in \mathcal{R}$, averaged over all codebook levels. Specifically,

$$\Omega = \mathbb{E}_{i\sim\mathcal{F},\, j\sim\mathcal{R}}\left[\omega(i,j)\right], \qquad \omega(i,j) = \frac{1}{L}\sum_{\ell=1}^{L}\sum_{k\in\mathcal{T}^\ell} \mathbf{1}[s_i^\ell = k]\mathbf{1}[s_j^\ell = k]. \tag{15}$$

Here, $\mathcal{T}^\ell$ denotes the token set at codebook level ℓ, and t_i^ℓ is the token assigned to item i at level ℓ. We define the loss of generating a token t conditioned on a sequence $\mathcal{T}$ as

$$\ell(\theta; t \mid \mathcal{T}) = -\log p_\theta(t \mid \mathcal{T}).$$

Assumption 3 *We assume that the token-level loss satisfies the following properties.*

(a) Lipschitzness. *For any token t and conditioning sequence $\mathcal{T}$ in the codebook,*

$$\|\nabla_\theta \ell(\theta; t \mid \mathcal{T})\|_2 \le \xi, \qquad \forall\theta.$$

(b) Smoothness. *For any token t and conditioning sequence $\mathcal{T}$ in the codebook,*

$$\|\nabla_\theta^2 \ell(\theta; t \mid \mathcal{T})\|_{\mathrm{op}} \le M, \qquad \forall\theta.$$

The above assumptions are standard in the optimization literature; see, for example, [44, Assumption 1].

Proposition 4 *Suppose Assumption 3 holds. Let*

$$\theta_+ := \theta - \eta\nabla_\theta \mathcal{L}_{\mathrm{unlearn}}(\theta)$$

be the model parameters after one gradient descent step on the unlearning objective. Then the retain-loss change satisfies

$$\mathcal{L}_R(\theta_+) - \mathcal{L}_R(\theta) \le \eta\alpha\frac{\xi^2}{L}\Omega + C_U. \tag{16}$$

Furthermore, if for any fixed token t and any conditioning sequences $\mathcal{T}, \mathcal{T}'$, the token-level gradients satisfy

$$\langle \nabla_\theta \ell(\theta; t \mid \mathcal{T}), \nabla_\theta \ell(\theta; t \mid \mathcal{T}')\rangle \ge \gamma > 0,$$

then

$$\mathcal{L}_R(\theta_+) - \mathcal{L}_R(\theta) \ge \eta\alpha\frac{\gamma}{L}\Omega + C_L. \tag{17}$$

Here, C_U and C_L collect residual gradient-alignment terms and second-order optimization remainders; they are not assumed to be invariant under token reassignment.

Proposition 4 identifies average token overlap Ω as a directly controllable source of forget-retain interference: the shared-token component of the forget-retain gradient alignment scales linearly with Ω and enters the retain-loss change with positive coefficient $\eta\alpha$ during unlearning. Thus, reducing Ω suppresses an explicit overlap-mediated source of retain-loss increase. We do not claim that Ω fully determines retain degradation, since the residual term C_U, C_L may also vary under reassignment; rather, the point is to isolate this shared-token mechanism and use it to motivate token reassignment, which we next show reduces the overlap Ω. In the lemma below, we consider fixed the model θ, and run one step of gradient descent on ϕ reduce the overlap.

We then made the following Assumption on the structure of the history and the .

Assumption 5 *We assume one of the following holds on the history of the items and the retain set:*

(A1) All the items in the history is not reassigned.

(A2) (1) The tokenizations of the items in the retain set and forget set are disjoint.
(2) For any $\mathcal{H}_u$ in the retain set, there exist a small enough $\varepsilon_{\text{hist}} > 0$ such that:

$$\frac{1}{|\mathcal{H}_u|} \sum_{i \in \mathcal{H}_u} \mathbf{1}[i \in \mathcal{F}] \leq \varepsilon_{\text{hist}}.$$

Further, we assume that $\Omega(\phi)$ is $\beta_{\phi,\Omega}$-smooth in ϕ.

Assumption 5 formalizes the idea that retain examples are only weakly coupled with the reassignment parameters of forget items. Condition (A1) corresponds to the simplified setting where reassigned tokens do not appear in histories, so the history component does not contribute to the ϕ-gradient. Condition (A2) covers a more general case: retain and forget items are token-separated, and forget items occur only sparsely in retain histories. Under this condition, the gradient of the retain loss with respect to forget-item reassignment parameters is small, of order $O(\varepsilon_{\text{hist}})$. The smoothness assumption is used only to convert first-order descent directions into one-step decreases for sufficiently small step size.

Lemma 6 *Under Assumption 5, for any fixed model θ, let $\phi_+ = \phi - \eta_\phi M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi)$, where M is define in 11 and the $\odot$ denote element-wise product. For small enough η_ϕ, we have:*

$$\Omega(\phi_+) - \Omega(\phi) \leq 0, \quad \mathcal{L}_F(\phi_+) - \mathcal{L}_F(\phi) \leq 0.$$

Lemma 6 shows that the masked reassignment update simultaneously reduces the forget log-likelihood and the average forget-retain token overlap, provided that the retain examples are only weakly coupled with forget-item reassignment. The mask M selects coordinates for which increasing the reassignment score would both increase the forget objective gradient and correspond to above-average overlap with retain tokens. Consequently, moving in the negative masked unlearning-gradient direction decreases the forget objective while also moving against the overlap gradient.

Combined with Proposition 4, this lemma provides a mechanistic justification for token reassignment. Proposition 4 identifies Ω as an explicit overlap-mediated component of the retain-loss increase during unlearning, while Lemma 6 shows that optimizing ϕ can directly reduce this component. Therefore, token reassignment can mitigate retain degradation by suppressing shared-token interference between forget and retain items. We emphasize that this does not imply that Ω fully determines the retain loss: the residual terms C_U and C_L in Proposition 4 may also change under reassignment. Rather, the result isolates one controllable mechanism through which reassignment can improve the retain-loss behavior.

B.2 Proof of Proposition 4: Token Overlap Affects Unlearning

We first write the forget and retain losses as averages of item-level generation losses:

$$\mathcal{L}_F(\theta) = \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} \ell_i(\theta), \qquad \mathcal{L}_R(\theta) = \frac{1}{|\mathcal{R}|} \sum_{j \in \mathcal{R}} \ell_j(\theta). \tag{18}$$

For an item i, we define the item-level loss as the average token-level generation loss across codebook levels:

$$\ell_i(\theta) = \frac{1}{L}\sum_{\ell=1}^{L} \ell(\theta; t_i^\ell \mid \mathcal{T}(\mathcal{H}_i), t_i^{<\ell}), \tag{19}$$

where

$$\ell(\theta; t_i^\ell \mid \mathcal{T}(\mathcal{H}_i), t_i^{<\ell}) = -\log p_\theta\left(t_i^\ell \mid \mathcal{T}(\mathcal{H}_i), t_i^{<\ell}\right). \tag{20}$$

Thus,

$$\nabla_\theta \mathcal{L}_F = \frac{1}{|\mathcal{F}|}\sum_{i\in\mathcal{F}} \nabla_\theta \ell_i, \qquad \nabla_\theta \mathcal{L}_R = \frac{1}{|\mathcal{R}|}\sum_{j\in\mathcal{R}} \nabla_\theta \ell_j. \tag{21}$$

The forget-retain gradient alignment can be expanded as

$$\begin{aligned}
\langle \nabla_\theta \mathcal{L}_F, \nabla_\theta \mathcal{L}_R \rangle &= \left\langle \frac{1}{|\mathcal{F}|}\sum_{i\in\mathcal{F}} \nabla_\theta \ell_i, \frac{1}{|\mathcal{R}|}\sum_{j\in\mathcal{R}} \nabla_\theta \ell_j \right\rangle \\
&= \frac{1}{|\mathcal{F}||\mathcal{R}|}\sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{R}} \langle \nabla_\theta \ell_i, \nabla_\theta \ell_j \rangle \\
&= \mathbb{E}_{i\sim\mathcal{F},\, j\sim\mathcal{R}}\left[\langle \nabla_\theta \ell_i, \nabla_\theta \ell_j \rangle\right].
\end{aligned} \tag{22}$$

Therefore, the global forget-retain gradient alignment is the expected pairwise gradient alignment between forget and retain items.

For each item i and level ℓ, define the token-level gradient component

$$g_{i,\ell} := \nabla_\theta \ell(\theta; t_i^\ell \mid \mathcal{T}(\mathcal{H}_i), t_i^{<\ell}). \tag{23}$$

For notational convenience, we also write $g_{i,\ell,k}$ for this gradient when $t_i^\ell = k$. Since

$$\ell_i(\theta) = \frac{1}{L}\sum_{\ell=1}^{L} \ell(\theta; t_i^\ell \mid \mathcal{T}(\mathcal{H}_i), t_i^{<\ell}),$$

we have

$$\nabla_\theta \ell_i(\theta) = \frac{1}{L}\sum_{\ell=1}^{L} g_{i,\ell}.$$

Therefore, the pairwise gradient alignment can be decomposed as

$$\begin{aligned}
\langle \nabla_\theta \ell_i, \nabla_\theta \ell_j \rangle &= \frac{1}{L^2}\sum_{\ell=1}^{L}\sum_{m=1}^{L} \langle g_{i,\ell}, g_{j,m} \rangle \\
&= \frac{1}{L^2}\sum_{\ell=1}^{L}\sum_{k\in\mathcal{T}^\ell} \mathbf{1}[t_i^\ell = k]\mathbf{1}[t_j^\ell = k]\, \langle g_{i,\ell,k}, g_{j,\ell,k} \rangle + C_{ij},
\end{aligned} \tag{24}$$

where the residual term C_{ij} is given explicitly by

$$\begin{aligned}
C_{ij} :=& \frac{1}{L^2}\sum_{\ell=1}^{L}\sum_{\substack{k,k'\in\mathcal{T}^\ell \\ k\neq k'}} \mathbf{1}[t_i^\ell = k]\mathbf{1}[t_j^\ell = k']\, \langle g_{i,\ell,k}, g_{j,\ell,k'} \rangle \\
&+ \frac{1}{L^2}\sum_{\substack{\ell,m=1 \\ \ell\neq m}}^{L}\sum_{k\in\mathcal{T}^\ell}\sum_{k'\in\mathcal{T}^m} \mathbf{1}[t_i^\ell = k]\mathbf{1}[t_j^m = k']\, \langle g_{i,\ell,k}, g_{j,m,k'} \rangle .
\end{aligned} \tag{25}$$

The first term in C_{ij} collects same-level but non-shared-token contributions, while the second term collects cross-level interactions.

Using the positive alignment assumption for shared tokens and the uniform gradient bound in Assumption 3, we have

$$\gamma \le \langle g_{i,\ell,k}, g_{j,\ell,k} \rangle \le \|g_{i,\ell,k}\|_2 \|g_{j,\ell,k}\|_2 \le \xi^2. \tag{26}$$

Therefore, from the above decomposition,

$$\begin{aligned}\frac{\gamma}{L^2}\sum_{\ell=1}^{L}\sum_{k\in\mathcal{T}^\ell}\mathbf{1}[t_i^\ell=k]\mathbf{1}[t_j^\ell=k]+C_{ij}\\ \le\langle\nabla_\theta\ell_i,\nabla_\theta\ell_j\rangle\\ \le\frac{\xi^2}{L^2}\sum_{\ell=1}^{L}\sum_{k\in\mathcal{T}^\ell}\mathbf{1}[t_i^\ell=k]\mathbf{1}[t_j^\ell=k]+C_{ij}.\end{aligned}\tag{27}$$

By the definition of token overlap,

$$\omega(i,j)=\frac{1}{L}\sum_{\ell=1}^{L}\sum_{k\in\mathcal{T}^\ell}\mathbf{1}[t_i^\ell=k]\mathbf{1}[t_j^\ell=k],$$

we have

$$\sum_{\ell=1}^{L}\sum_{k\in\mathcal{T}^\ell}\mathbf{1}[t_i^\ell=k]\mathbf{1}[t_j^\ell=k]=L\omega(i,j).$$

Hence,

$$\frac{\gamma}{L}\omega(i,j)+C_{ij}\le\langle\nabla_\theta\ell_i,\nabla_\theta\ell_j\rangle\le\frac{\xi^2}{L}\omega(i,j)+C_{ij}.\tag{28}$$

Taking expectation over $i\sim\mathcal{F}$ and $j\sim\mathcal{R}$ gives

$$\begin{aligned}\langle\nabla_\theta\mathcal{L}_F,\nabla_\theta\mathcal{L}_R\rangle&=\mathbb{E}_{i\sim\mathcal{F},\,j\sim\mathcal{R}}\left[\langle\nabla_\theta\ell_i,\nabla_\theta\ell_j\rangle\right]\\ &\ge\mathbb{E}_{i\sim\mathcal{F},\,j\sim\mathcal{R}}\left[\frac{\gamma}{L}\omega(i,j)+C_{ij}\right]\\ &=\frac{\gamma}{L}\mathbb{E}_{i\sim\mathcal{F},\,j\sim\mathcal{R}}\left[\omega(i,j)\right]+\mathbb{E}_{i\sim\mathcal{F},\,j\sim\mathcal{R}}\left[C_{ij}\right].\end{aligned}\tag{29}$$

Similarly, the upper bound in equation 28 yields

$$\langle\nabla_\theta\mathcal{L}_F,\nabla_\theta\mathcal{L}_R\rangle\le\frac{\xi^2}{L}\mathbb{E}_{i\sim\mathcal{F},\,j\sim\mathcal{R}}\left[\omega(i,j)\right]+\mathbb{E}_{i\sim\mathcal{F},\,j\sim\mathcal{R}}\left[C_{ij}\right].\tag{30}$$

Since

$$\Omega=\mathbb{E}_{i\sim\mathcal{F},\,j\sim\mathcal{R}}\left[\omega(i,j)\right],$$

and defining

$$C_0:=\mathbb{E}_{i\sim\mathcal{F},\,j\sim\mathcal{R}}\left[C_{ij}\right],$$

we obtain the two-sided gradient-alignment bound

$$\frac{\gamma}{L}\Omega+C_0\le\langle\nabla_\theta\mathcal{L}_F,\nabla_\theta\mathcal{L}_R\rangle\le\frac{\xi^2}{L}\Omega+C_0.\tag{31}$$

We now translate this gradient-alignment bound into a bound on the retain-loss change after one unlearning step. Recall that

$$\mathcal{L}_{\text{unlearn}}(\theta)=\mathcal{L}_R(\theta)-\alpha\mathcal{L}_F(\theta),\qquad\alpha>0.$$

One gradient descent step gives

$$\theta_+=\theta-\eta\nabla_\theta\mathcal{L}_{\text{unlearn}}(\theta)=\theta-\eta\nabla_\theta\mathcal{L}_R(\theta)+\eta\alpha\nabla_\theta\mathcal{L}_F(\theta).$$

For brevity, write

$$G_R:=\nabla_\theta\mathcal{L}_R(\theta),\qquad G_F:=\nabla_\theta\mathcal{L}_F(\theta),$$

and

$$\Delta:=\theta_+-\theta=-\eta G_R+\eta\alpha G_F=-\eta(G_R-\alpha G_F).$$

By Assumption 3, the token-level loss is M-smooth. Since the item-level and retain losses are averages of token-level losses, $\mathcal{L}_R$ is also M-smooth. Hence, Taylor's theorem with remainder gives

$$\langle G_R,\Delta\rangle-\frac{M}{2}\|\Delta\|^2\le\mathcal{L}_R(\theta_+)-\mathcal{L}_R(\theta)\le\langle G_R,\Delta\rangle+\frac{M}{2}\|\Delta\|^2.\tag{32}$$

Moreover,

$$\langle G_R, \Delta \rangle = -\eta \|G_R\|^2 + \eta\alpha \langle G_R, G_F \rangle ,$$

and

$$\|\Delta\|^2 = \eta^2 \|G_R - \alpha G_F\|^2.$$

Substituting these identities into equation 32, we obtain

$$\begin{aligned} &-\eta\|G_R\|^2 + \eta\alpha \langle G_R, G_F \rangle - \frac{M\eta^2}{2}\|G_R - \alpha G_F\|^2 \\ &\quad \le \mathcal{L}_R(\theta_+) - \mathcal{L}_R(\theta) \\ &\quad \le -\eta\|G_R\|^2 + \eta\alpha \langle G_R, G_F \rangle + \frac{M\eta^2}{2}\|G_R - \alpha G_F\|^2. \end{aligned} \tag{33}$$

Combining equation 33 with the lower bound in equation 31, we get

$$\mathcal{L}_R(\theta_+) - \mathcal{L}_R(\theta) \ge -\eta\|G_R\|^2 + \eta\alpha \left(\frac{\gamma}{L}\Omega + C_0\right) - \frac{M\eta^2}{2}\|G_R - \alpha G_F\|^2. \tag{34}$$

Similarly, combining equation 33 with the upper bound in equation 31, we get

$$\mathcal{L}_R(\theta_+) - \mathcal{L}_R(\theta) \le -\eta\|G_R\|^2 + \eta\alpha \left(\frac{\xi^2}{L}\Omega + C_0\right) + \frac{M\eta^2}{2}\|G_R - \alpha G_F\|^2. \tag{35}$$

Finally, all terms in equation 34 and equation 35 that do not explicitly depend on Ω can be absorbed into constants. Thus, there exist constants C_L and C_U, independent of the explicit shared-token overlap term, such that

$$\eta\alpha\frac{\gamma}{L}\Omega + C_L \le \mathcal{L}_R(\theta_+) - \mathcal{L}_R(\theta) \le \eta\alpha\frac{\xi^2}{L}\Omega + C_U. \tag{36}$$

This proves the desired two-sided control of the retain-loss change in terms of the average token overlap Ω.

B.3 Proof of Lemma 6: Optimizing the Unlearning loss Reduces Token Overlap

In this section, we show that optimizing the token reassignment parameters can reduce the overlap measure Ω, thereby tightening the overlap-dependent component in the bound on $\mathcal{L}_R(\theta_+) - \mathcal{L}_R(\theta)$ from Proposition 4. We first introduce the token reassignment mechanism and formulate item generation probabilistically. We then prove the overlap-reduction result under the two cases in Assumption 5. Section B.3.2 treats Assumption 5(A1), where histories do not contain reassigned items and hence the history tokenization does not depend on ϕ. Section B.3.3 treats Assumption 5(A2), where histories may contain reassigned items, but retain and forget items are token-separated and forget items appear only sparsely in retain histories.

B.3.1 Unlearning loss as a function of the reassignment parameter ϕ

Given the reassignment parameter ϕ, we recall that the probability of tokenize user i into token k at level ℓ is:

$$q_\phi(s_i^\ell = k|i) = \frac{\exp\left((-\|\boldsymbol{r}_i^l - \boldsymbol{c}_k^l\|^2 + \phi_{i,k}^l)/\tau\right)}{\sum_{j=1}^K \exp\left((-\|\boldsymbol{r}_i^l - \boldsymbol{c}_j^l\|^2 + \phi_{i,j}^l)/\tau\right)}$$

Thus, given the history $\mathcal{H}_u$, and the target item i, the probability of generating the target item i given the history $\mathcal{H}_u$ is

$$\begin{aligned} \Pr_{\phi,\theta}[i|\mathcal{H}_u] &= \sum_{\mathcal{T}(\mathcal{H}_u)} \sum_{\mathcal{T}} \Pr[i|\mathcal{T}] p_\theta(\mathcal{T} \mid \mathcal{T}(\mathcal{H}_u)) \Pr[\mathcal{T}(\mathcal{H}_u)|\mathcal{H}_u] \\ &= \sum_{\mathcal{T}(\mathcal{H}_u)} \sum_{\mathcal{T}} \frac{q_\phi(\mathcal{T}|i)\Pr(i)}{\sum_j q_\phi(\mathcal{T}|j)\Pr(j)} p_\theta(\mathcal{T} \mid \mathcal{T}(\mathcal{H}_u)) \Pr[\mathcal{T}(\mathcal{H}_u)|\mathcal{H}_u] \\ &= \sum_{\mathcal{T}} \frac{q_\phi(\mathcal{T}|i)\Pr(i)}{\sum_j q_\phi(\mathcal{T}|j)\Pr(j)} \sum_{\mathcal{T}(\mathcal{H}_u)} p_\theta(\mathcal{T} \mid \mathcal{T}(\mathcal{H}_u)) \prod_{j \in \mathcal{H}_u} \frac{q_\phi(\mathcal{T}(j)|j)\Pr(j)}{\sum_k q_\phi(\mathcal{T}(j)|k)\Pr(k)} \end{aligned}$$

where $\mathcal{T}$ denote a sequence of tokens of length L, which is a possible tokenization of the target item i, $\mathcal{T}(\mathcal{H}_u) = \{\mathcal{T}(j)\}_{j\in\mathcal{H}_u}$ which is the tokenization of the history of user u. $\Pr[i]$ denote the frequency of item i in the dataset. The first step is by the law of total probability, and the second and third step is by the Bayes rule, as well as the fact that each item in the history is tokenized independently.

Now for fixed model θ the forget and retain loss can be equivalently written as:

$$\mathcal{L}_F(\phi;\theta) = \sum_{(\mathcal{H}_u,i)\in F} \log \Pr_{\phi,\theta}[i|\mathcal{H}_u]$$
$$\mathcal{L}_R(\phi;\theta) = -\sum_{(\mathcal{H}_r,i_r)\in \mathcal{R}} \log \Pr_{\phi,\theta}[i_r|\mathcal{H}_r]$$
$$\mathcal{L}_{\text{unlearn}}(\phi;\theta) = \mathcal{L}_F((\phi;\theta) - \alpha\mathcal{L}_R((\phi;\theta)$$

and we drop the dependency on θ when there's no confusion. Next, we study how one step of gradient descent on ϕ effect the forget loss and retain loss. In particular, recall that the token overlap between a forget item i and a retain item j is defined as

$$\omega(i,j) = \frac{1}{L}\sum_{\ell=1}^{L}\sum_{k\in\mathcal{T}^\ell} \mathbf{1}[s_i^\ell = k]\mathbf{1}[s_j^\ell = k]. \tag{37}$$

Since the token of a forget item is reassigned according to $q_\phi(\cdot \mid i)$, while the retain item tokens are fixed, the expected overlap can be written as

$$\mathbb{E}[\omega(i,j)] = \frac{1}{L}\sum_{\ell=1}^{L}\sum_{k\in\mathcal{T}^\ell} q_\phi(s_i^\ell = k \mid i) q_\phi(s_j^\ell = k \mid j) \tag{38}$$

Therefore, the average expected sensitive-token overlap is

$$\Omega(\phi) = \frac{1}{|\mathcal{F}||\mathcal{R}|L}\sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{R}}\sum_{\ell=1}^{L}\sum_{k\in\mathcal{T}^\ell} q_\phi(s_i^\ell = k \mid i) q_\phi(s_j^\ell = k \mid j). \tag{39}$$

The update of the reassignment parameters on the unlearning loss $\mathcal{L}_{\text{unlearn}}$ is:

$$(\phi_{i,b}^\ell)_+ = \phi_{i,b}^\ell - \eta_\phi M_{i,b}^\ell \cdot \nabla_{\phi_{i,b}^\ell}\mathcal{L}_{\text{unlearn}}(\phi)$$

with the selective mask M define as

$$M_{i,b}^\ell = \mathbf{1}\left[\rho_\ell(b) > \bar{\rho}_{i,\ell}\right] \cdot \mathbf{1}\left[\nabla_{\phi_{i,b}^\ell}\mathcal{L}_F > 0\right],$$

where

$$\bar{\rho}_{i,\ell} = \sum_{k\in\mathcal{V}^\ell} q_\phi(k \mid i)\rho_\ell(k), \qquad \rho_\ell(k) = \frac{1}{|\mathcal{R}|}\sum_{j\in\mathcal{R}} \mathbf{1}\left[t_j^\ell = k\right]. \tag{40}$$

We aim to show that:

$$\Omega(\phi_+) - \Omega(\phi) \le 0,$$
$$\mathcal{L}_F(\phi_+) - \mathcal{L}_F(\phi_+) \le 0.$$

B.3.2 Proof under Assumption 5(A1)

We first consider a simple case with the following assumptions:

(a) The history of all items does not contain items in the forget set.

Note that the token reassignment is only for items in the forget set, which implies that $\mathcal{L}_R$ does not depend on ϕ in this case and thus $\nabla_\phi\mathcal{L}_{\text{unlearn}} = \nabla_\phi\mathcal{L}_F$. Thus, we immediately have:

$$\mathcal{L}_F(\phi_+) \le \mathcal{L}_F(\phi)$$

for small enough η_ϕ.

Next, recall the definition of Ω,

$$\Omega(\phi) = \frac{1}{|\mathcal{F}||\mathcal{R}|L} \sum_{i\in\mathcal{F}} \sum_{j\in\mathcal{R}} \sum_{\ell=1}^{L} \sum_{k\in\mathcal{T}^\ell} q_\phi(s_i^\ell = k \mid i) q_\phi(s_j^\ell = k \mid j).$$

and directly compute the gradient, we obtained that:

$$\begin{aligned}
\frac{\partial \Omega}{\partial \phi_{i,b}^\ell} &= \frac{1}{|\mathcal{F}|L} \sum_{k\in\mathcal{V}^\ell} \rho_\ell(k) \frac{\partial q_\phi(k \mid i)}{\partial \phi_{i,b}^\ell} \\
&= \frac{1}{\tau|\mathcal{F}|L} \sum_{k\in\mathcal{V}^\ell} \rho_\ell(k) q_\phi(k \mid i) \left(\mathbf{1}[k=b] - q_\phi(b \mid i)\right) \\
&= \frac{1}{\tau|\mathcal{F}|L} q_\phi(b \mid i) \left(\rho_\ell(b) - \bar{\rho}_{i,\ell}\right),
\end{aligned} \tag{41}$$

where

$$\bar{\rho}_{i,\ell} = \sum_{k\in\mathcal{V}^\ell} q_\phi(k \mid i) \rho_\ell(k), \qquad \rho_\ell(k) = \frac{1}{|\mathcal{R}|} \sum_{j\in\mathcal{R}} \mathbf{1}\left[t_j^\ell = k\right]. \tag{42}$$

It is sufficient to show

$$\langle \nabla_\phi \Omega, M \cdot \nabla_\phi \mathcal{L}_{\text{unlearn}} \rangle \geq 0.$$

Indeed we have:

$$\begin{aligned}
&\langle \nabla_\phi \Omega(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \rangle \\
&= \sum_{i\in\mathcal{F}} \sum_{\ell=1}^{L} \sum_{b\in\mathcal{V}^\ell} \frac{\partial \Omega(\phi)}{\partial \phi_{i,b}^\ell} M_{i,b}^\ell \frac{\partial \mathcal{L}_{\text{unlearn}}(\phi)}{\partial \phi_{i,b}^\ell} \\
&= \frac{1}{\tau|\mathcal{F}|L} \sum_{i\in\mathcal{F}} \sum_{\ell=1}^{L} \sum_{b\in\mathcal{V}^\ell} q_\phi(b \mid i) \left(\rho_\ell(b) - \bar{\rho}_{i,\ell}\right) M_{i,b}^\ell \frac{\partial \mathcal{L}_F(\phi)}{\partial \phi_{i,b}^\ell} \\
&\geq 0.
\end{aligned} \tag{43}$$

B.3.3 Proof under Assumption 5(A2)

We consider the general case where the histories of examples in both the forget and retain sets may contain forget items. In this case, both $\mathcal{L}_F$ and $\mathcal{L}_R$ depend on ϕ. Recall that

$$\mathcal{L}_{\text{unlearn}}(\phi) = \mathcal{L}_F(\phi) - \alpha \mathcal{L}_R(\phi).$$

Hence,

$$\nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) = \nabla_\phi \mathcal{L}_F(\phi) - \alpha \nabla_\phi \mathcal{L}_R(\phi). \tag{44}$$

We first control the retain-gradient term. Since the reassignment parameter ϕ only acts on forget items, the retain loss can depend on ϕ only through two sources: a retain target token sequence may be confused with the tokenization of a forget item, or forget items may appear in the histories of retain examples.

Under Assumption (A2), retain items and forget items are token-separated. Thus, for any retain item $y \in \mathcal{R}$, forget item $i \in \mathcal{F}$, and token sequence $\mathcal{T}$ generated by a retain item, we have

$$q_\phi(i \mid \mathcal{T}) = 0.$$

Therefore, the target-token contribution to

$$\frac{\partial}{\partial \phi_{i,b}^\ell} \log \Pr_{\phi,\theta}[y \mid \mathcal{H}_u]$$

vanishes for retain examples. Indeed, this contribution contains the factor

$$\mathbf{1}[y=i] - q_\phi(i \mid \mathcal{T}).$$

Since $y \in \mathcal{R}$ and $i \in \mathcal{F}$, we have $\mathbf{1}[y = i] = 0$. Moreover, token separation gives $q_\phi(i \mid \mathcal{T}) = 0$. Hence the target-token contribution is zero.

Consequently, the only remaining dependence of the retain loss on ϕ comes from forget items appearing in retain histories. By the history sparsity condition in Assumption (A2), and using the boundedness of the softmax score terms, there exists a constant $C_R > 0$ such that

$$\|\nabla_\phi \mathcal{L}_R(\phi)\| \le C_R \varepsilon_{\text{hist}}. \tag{45}$$

We now show that the reassignment update decreases the forget loss. Recall the masked update

$$\phi_+ = \phi - \eta_\phi M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi). \tag{46}$$

Assume that, for fixed θ, $\mathcal{L}_F(\phi; \theta)$ is $\beta_{F,\phi}$-smooth as a function of ϕ. Then

$$\begin{aligned} \mathcal{L}_F(\phi_+) - \mathcal{L}_F(\phi) &\le \langle \nabla_\phi \mathcal{L}_F(\phi), \phi_+ - \phi \rangle + \frac{\beta_{F,\phi}}{2} \|\phi_+ - \phi\|^2 \\ &= -\eta_\phi \left\langle \nabla_\phi \mathcal{L}_F(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \right\rangle \\ &\quad + \frac{\beta_{F,\phi} \eta_\phi^2}{2} \left\| M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \right\|^2. \end{aligned} \tag{47}$$

Using equation 44, we have

$$\begin{aligned} &\langle \nabla_\phi \mathcal{L}_F(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \rangle \\ &= \langle \nabla_\phi \mathcal{L}_F(\phi), M \odot \nabla_\phi \mathcal{L}_F(\phi) \rangle - \alpha \left\langle \nabla_\phi \mathcal{L}_F(\phi), M \odot \nabla_\phi \mathcal{L}_R(\phi) \right\rangle \\ &= \|M \odot \nabla_\phi \mathcal{L}_F(\phi)\|^2 - \alpha \left\langle M \odot \nabla_\phi \mathcal{L}_F(\phi), \nabla_\phi \mathcal{L}_R(\phi) \right\rangle. \end{aligned} \tag{48}$$

By Cauchy–Schwarz and equation 45,

$$\begin{aligned} &\langle \nabla_\phi \mathcal{L}_F(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \rangle \\ &\ge \|M \odot \nabla_\phi \mathcal{L}_F(\phi)\|^2 - \alpha \|M \odot \nabla_\phi \mathcal{L}_F(\phi)\| \, \|\nabla_\phi \mathcal{L}_R(\phi)\| \\ &\ge \|M \odot \nabla_\phi \mathcal{L}_F(\phi)\|^2 - \alpha C_R \varepsilon_{\text{hist}} \|M \odot \nabla_\phi \mathcal{L}_F(\phi)\| \\ &= \|M \odot \nabla_\phi \mathcal{L}_F(\phi)\| \left(\|M \odot \nabla_\phi \mathcal{L}_F(\phi)\| - \alpha C_R \varepsilon_{\text{hist}} \right). \end{aligned} \tag{49}$$

Therefore, if the retain-history coupling is sufficiently small so that

$$\alpha C_R \varepsilon_{\text{hist}} < \|M \odot \nabla_\phi \mathcal{L}_F(\phi)\|, \tag{50}$$

then

$$\langle \nabla_\phi \mathcal{L}_F(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \rangle > 0 \tag{51}$$

whenever $M \odot \nabla_\phi \mathcal{L}_F(\phi) \neq 0$. Hence, for a sufficiently small step size $\eta_\phi > 0$, the first-order decrease dominates the second-order smoothness term. In particular, if

$$0 < \eta_\phi \le \frac{2 \left\langle \nabla_\phi \mathcal{L}_F(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \right\rangle}{\beta_{F,\phi} \left\| M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \right\|^2}, \tag{52}$$

then

$$\mathcal{L}_F(\phi_+) \le \mathcal{L}_F(\phi). \tag{53}$$

It remains to show that the same update decreases the overlap $\Omega(\phi)$. Although the losses depend on ϕ through both target and history tokenizations in the dependent-history setting, $\Omega(\phi)$ itself does not involve histories. Since retain item tokens are fixed, we can write

$$\Omega(\phi) = \frac{1}{|\mathcal{F}| L} \sum_{i \in \mathcal{F}} \sum_{\ell=1}^{L} \sum_{b \in \mathcal{V}^\ell} q_\phi(b \mid i) \rho_\ell(b), \tag{54}$$

where

$$\rho_\ell(b) = \frac{1}{|\mathcal{R}|} \sum_{j \in \mathcal{R}} \mathbf{1}[t_j^\ell = b].$$

Therefore,

$$\frac{\partial \Omega(\phi)}{\partial \phi_{i,b}^\ell} = \frac{1}{\tau |\mathcal{F}| L} q_\phi(b \mid i) \left(\rho_\ell(b) - \bar{\rho}_{i,\ell} \right), \tag{55}$$

where

$$\bar{\rho}_{i,\ell} = \sum_{k \in \mathcal{V}^\ell} q_\phi(k \mid i) \rho_\ell(k).$$

Assume that $\Omega(\phi)$ is $\beta_{\Omega,\phi}$-smooth in ϕ. Then

$$\begin{aligned} \Omega(\phi_+) - \Omega(\phi) &\leq \langle \nabla_\phi \Omega(\phi), \phi_+ - \phi \rangle + \frac{\beta_{\Omega,\phi}}{2} \|\phi_+ - \phi\|^2 \\ &= -\eta_\phi \left\langle \nabla_\phi \Omega(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \right\rangle \\ &\quad + \frac{\beta_{\Omega,\phi} \eta_\phi^2}{2} \left\| M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \right\|^2. \end{aligned} \tag{56}$$

Thus, it remains to show that

$$\langle \nabla_\phi \Omega(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \rangle > 0.$$

Using equation 44, we decompose

$$\begin{aligned} &\langle \nabla_\phi \Omega(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \rangle \\ &= \langle \nabla_\phi \Omega(\phi), M \odot \nabla_\phi \mathcal{L}_F(\phi) \rangle - \alpha \left\langle \nabla_\phi \Omega(\phi), M \odot \nabla_\phi \mathcal{L}_R(\phi) \right\rangle. \end{aligned} \tag{57}$$

Define

$$\Delta_\Omega(\phi) := \langle \nabla_\phi \Omega(\phi), M \odot \nabla_\phi \mathcal{L}_F(\phi) \rangle. \tag{58}$$

Using equation 55, we obtain

$$\begin{aligned} \Delta_\Omega(\phi) &= \frac{1}{\tau |\mathcal{F}| L} \sum_{i \in \mathcal{F}} \sum_{\ell=1}^{L} \sum_{b \in \mathcal{V}^\ell} M_{i,b}^\ell q_\phi(b \mid i) \left(\rho_\ell(b) - \bar{\rho}_{i,\ell}\right) \frac{\partial \mathcal{L}_F(\phi)}{\partial \phi_{i,b}^\ell} \\ &\geq 0. \end{aligned} \tag{59}$$

The last inequality follows from the definition of the mask. Indeed, whenever $M_{i,b}^\ell = 1$, we have both

$$\rho_\ell(b) - \bar{\rho}_{i,\ell} > 0$$

and

$$\frac{\partial \mathcal{L}_F(\phi)}{\partial \phi_{i,b}^\ell} > 0.$$

Here $\partial_{\phi_{i,b}^\ell} \mathcal{L}_F(\phi)$ denotes the full gradient of the forget loss with respect to ϕ, including both target-token and history-token contributions.

Next, by Cauchy–Schwarz and equation 45,

$$\begin{aligned} &|\langle \nabla_\phi \Omega(\phi), M \odot \nabla_\phi \mathcal{L}_R(\phi) \rangle| \\ &\leq \|M \odot \nabla_\phi \Omega(\phi)\| \, \|\nabla_\phi \mathcal{L}_R(\phi)\| \\ &\leq C_R \varepsilon_{\text{hist}} \|M \odot \nabla_\phi \Omega(\phi)\|. \end{aligned} \tag{60}$$

Therefore,

$$\begin{aligned} &\langle \nabla_\phi \Omega(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \rangle \\ &\geq \Delta_\Omega(\phi) - \alpha C_R \varepsilon_{\text{hist}} \|M \odot \nabla_\phi \Omega(\phi)\|. \end{aligned} \tag{61}$$

Hence, if the second retain-history coupling condition holds:

$$\Delta_\Omega(\phi) > \alpha C_R \varepsilon_{\text{hist}} \|M \odot \nabla_\phi \Omega(\phi)\|, \tag{62}$$

then

$$\langle \nabla_\phi \Omega(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \rangle > 0. \tag{63}$$

Consequently, for sufficiently small $\eta_\phi > 0$, the first-order decrease dominates the second-order smoothness term. In particular, if

$$0 < \eta_\phi \leq \frac{2 \langle \nabla_\phi \Omega(\phi), M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \rangle}{\beta_{\Omega,\phi} \left\| M \odot \nabla_\phi \mathcal{L}_{\text{unlearn}}(\phi) \right\|^2}, \tag{64}$$

then

$$\Omega(\phi_+) \leq \Omega(\phi). \tag{65}$$

Combining the two parts proves that, under Assumption (A2), if equation 50 and equation 62 hold, then for sufficiently small $\eta_\phi > 0$,

$$\mathcal{L}_F(\phi_+) \leq \mathcal{L}_F(\phi), \qquad \Omega(\phi_+) \leq \Omega(\phi).$$

C Related Work

Machine Unlearning. Machine unlearning aims to remove the influence of specific training data from a learned model without retraining from scratch [3, 1]. Exact unlearning methods such as SISA [1] partition training data and retrain affected shards, offering provable guarantees but incurring substantial computational cost. Approximate approaches trade exactness for efficiency: gradient ascent on forget samples [48, 35], Fisher-information-based parameter updates [24], and influence-function-based corrections [25]. More recently, unlearning has been studied in large language models, where methods such as ELM [22], RMU [32, 60], REPNoise [45, 34], and FLAT [51] modify model weights or representations to suppress specific knowledge. These methods operate directly on model parameters and do not account for the structured semantic identifiers used in generative recommender systems.

Recommendation Unlearning. Unlearning in recommender systems poses unique challenges due to collaborative signals: removing one user's or item's influence may propagate through learned interaction patterns [4]. RecEraser [4] extends SISA-style partitioning to collaborative filtering but requires maintaining multiple sub-models. BLUR [44] proposes bi-level optimization to balance forgetting and retention in sequential recommendation. PISCES [27] introduces preference-aware unlearning that preserves user preference structures during item removal. However, existing recommendation unlearning methods are designed for traditional ID-based or embedding-based recommenders and do not address the discrete token structure of generative recommendation models, where item identity is encoded as a sequence of hierarchical codes rather than a single embedding vector.

Generative Recommendation with Semantic IDs. A recent paradigm casts recommendation as sequence-to-sequence generation, where each item is represented by a semantic ID (SID)—a short sequence of discrete tokens derived from a hierarchical quantizer such as RQ-VAE [31]. TIGER [43] first introduced this framework with a T5 backbone, followed by P5-SID [49], LETTER [49], and ETEGRec [36] which improve tokenization and training strategies. MiniOneRec [30] scales this approach to billion-parameter LLMs (e.g., Qwen2.5-7B) with multi-task learning. In these models, an item's identity is fully determined by its SID token sequence; simply removing an item's training interactions does not erase its code from the model's vocabulary or prevent the model from generating it. TRACER is, to our knowledge, the first method to address unlearning in this generative SID-based recommendation setting by directly reassigning the discrete tokens of forget items.

D Additional Results

D.1 Implementation details.

All backbones use a three-layer RQ-VAE tokenizer (codebook size 256 per layer, embedding dimension 64). TRACER optimizes the LLM with AdamW (lr 3×10^{-5}) and the soft-reassignment parameters with Adam (lr 1×10^{-2}), batch size 128, for up to 10 epochs. The Softmax temperature is annealed from 2.0 to 0.1. Loss weights are λ_1=0.4, λ_2=0.08, λ_3=0.08; positive set size K=5; reassignment temperature τ=0.005. All baselines start from the same SFT checkpoint with tuned learning rates in $\{10^{-3}, 10^{-4}, 10^{-5}\}$. Evaluation uses beam search (width 10). All results are averaged over 5 runs on 4× A100 GPUs.

D.2 More results on diverse datasets

We report the complete set of unlearning results across all five backbones (MOR, TIGER, P5SID, LETTER, ETEGRec) and three datasets. The main paper presents results for two backbone–dataset combinations (MOR and LETTER on Industrial & Scientific and Sports & Outdoors); here we provide the full results including all five backbones [33]. Table 2 reports results on **Industrial & Scientific**, Table 3 on **Sports & Outdoors**, and Table 4 on **Toys & Games**. Each table includes Retain metrics (HR@5, HR@10, NDCG@5, NDCG@10, MRR; higher is better), Forget metrics (the same five; lower is better), and Semantic Similarity (higher is better).

Across all 15 backbone–dataset combinations, TRACER consistently achieves the best performance on every metric group. On the Forget side, TRACER attains the lowest leakage in all settings, often by a substantial margin over the second-best method (PISCES). On the Retain side, TRACER

Table 2: Complete unlearning results on **Industrial & Scientific**. Bold = best among unlearning methods; underline = second best. Retain metrics (↑): higher is better; Forget metrics (↓): lower is better; Sim. (↑): higher is better.

Backbone	Method	Retain (↑)					Forget (↓)					Sim.↑
		HR@5	HR@10	NDCG@5	NDCG@10	MRR	HR@5	HR@10	NDCG@5	NDCG@10	MRR	
MOR	Original	0.126	0.171	0.100	0.110	0.100	0.170	0.230	0.110	0.150	0.110	0.851
	RMU	0.061	0.121	0.051	0.083	0.089	0.151	0.181	0.094	0.121	0.098	0.730
	ELM	0.084	0.139	0.068	0.093	0.091	0.143	0.176	0.091	0.116	0.096	0.824
	BLUR	0.120	0.166	0.095	0.109	0.095	0.139	0.171	0.087	0.111	0.093	0.680
	REPNoise	0.068	0.126	0.056	0.086	0.090	0.131	0.163	0.081	0.105	0.089	0.630
	FLAT	0.119	0.162	0.091	0.107	0.095	0.119	0.148	0.074	0.096	0.082	0.710
	PISCES	0.115	0.161	0.090	0.106	0.095	0.101	0.129	0.066	0.086	0.076	0.790
	TRACER	**0.123**	**0.168**	**0.097**	**0.110**	**0.096**	**0.080**	**0.110**	**0.058**	**0.079**	**0.071**	**0.828**
TIGER	Original	0.102	0.144	0.089	0.101	0.088	0.144	0.203	0.099	0.139	0.099	0.872
	RMU	0.056	0.109	0.053	0.079	0.080	0.128	0.153	0.083	0.109	0.086	0.750
	ELM	0.061	0.111	0.058	0.080	0.079	0.119	0.149	0.079	0.103	0.084	0.836
	BLUR	0.095	0.140	**0.086**	0.097	0.086	0.116	0.145	0.076	0.100	0.081	0.630
	REPNoise	0.059	0.112	0.054	0.080	0.081	0.107	0.137	0.070	0.094	0.077	0.650
	FLAT	0.091	0.134	0.080	0.095	0.083	0.096	0.124	0.063	0.085	0.071	0.770
	PISCES	0.088	0.136	0.079	0.094	0.084	0.083	0.114	0.055	0.075	0.066	0.862
	TRACER	**0.097**	**0.141**	**0.086**	**0.098**	**0.087**	**0.061**	**0.087**	**0.047**	**0.066**	**0.061**	**0.863**
P5SID	Original	0.103	0.145	0.090	0.101	0.089	0.148	0.198	0.100	0.136	0.097	0.843
	RMU	0.055	0.108	0.052	0.078	0.078	0.131	0.151	0.084	0.106	0.085	0.718
	ELM	0.064	0.113	0.059	0.081	0.082	0.123	0.145	0.080	0.102	0.083	0.815
	BLUR	0.096	0.140	**0.085**	0.097	0.086	0.121	0.142	0.077	0.099	0.080	0.672
	REPNoise	0.058	0.110	0.055	0.079	0.080	0.111	0.134	0.071	0.092	0.076	0.638
	FLAT	0.087	0.133	0.078	0.093	0.083	0.095	0.122	0.060	0.082	0.070	0.703
	PISCES	0.091	0.136	0.081	0.095	0.084	0.085	0.111	0.054	0.074	0.065	0.784
	TRACER	**0.097**	**0.142**	**0.085**	**0.099**	**0.087**	**0.063**	**0.090**	**0.045**	**0.066**	**0.060**	**0.836**
LETTER	Original	0.128	0.174	0.103	0.112	0.101	0.192	0.258	0.121	0.162	0.121	0.832
	RMU	0.059	0.118	0.049	0.082	0.087	0.178	0.207	0.108	0.137	0.110	0.730
	ELM	0.086	0.142	0.070	0.094	0.092	0.167	0.202	0.104	0.133	0.107	0.821
	BLUR	0.123	0.168	0.096	**0.110**	**0.098**	0.164	0.198	0.101	0.129	0.104	0.720
	REPNoise	0.070	0.129	0.057	0.086	0.089	0.145	0.174	0.090	0.114	0.096	0.690
	FLAT	0.113	0.158	0.089	0.105	0.094	0.148	0.177	0.091	0.116	0.097	0.710
	PISCES	0.117	0.164	0.092	0.108	0.095	0.125	0.153	0.077	0.099	0.092	0.807
	TRACER	**0.124**	**0.170**	**0.098**	**0.110**	**0.098**	**0.102**	**0.133**	**0.068**	**0.090**	**0.081**	**0.822**
ETEGRec	Original	0.127	0.175	0.102	0.114	0.100	0.209	0.267	0.130	0.162	0.129	0.856
	RMU	0.065	0.124	0.056	0.086	0.089	0.178	0.215	0.114	0.145	0.116	0.738
	ELM	0.088	0.144	0.072	0.096	0.091	0.174	0.211	0.110	0.141	0.112	0.829
	BLUR	0.123	0.170	0.097	0.110	0.097	0.169	0.207	0.106	0.136	0.108	0.691
	REPNoise	0.064	0.123	0.050	0.082	0.085	0.154	0.185	0.095	0.121	0.099	0.657
	FLAT	0.117	0.162	0.092	0.107	0.095	0.158	0.189	0.097	0.124	0.101	0.726
	PISCES	0.114	0.161	0.088	0.104	0.093	0.137	0.168	0.084	0.108	0.095	0.812
	TRACER	**0.126**	**0.172**	**0.101**	**0.112**	**0.099**	**0.111**	**0.145**	**0.075**	**0.098**	**0.087**	**0.847**

matches or exceeds the Original model's utility in most configurations, confirming that targeted token reassignment effectively preserves recommendation quality for non-forget users. For Semantic Similarity, TRACER ranks first in the majority of settings, demonstrating that the coherence-aware design maintains meaningful item representations after unlearning [38, 14].

D.3 Efficiency Comparison

Table 5 reports the wall-clock unlearning time (in minutes) for each method across all four backbone–dataset combinations used in the main experiments. All timings are averaged over 5 runs on 4× NVIDIA A100 GPUs. TRACER is consistently the fastest method, requiring only 19–27 minutes depending on the setting—roughly 2× faster than the next-fastest baseline (ELM) and up to 3.5× faster than the slowest (RMU) [56, 58]. Notably, LETTER-based configurations are faster than MOR across all methods, as the joint-training backbone involves fewer sequential decoding steps during unlearning.

D.4 Generalization to Item-Level Unlearning

While our main experiments focus on brand-level unlearning (removing all items associated with a target brand), we further evaluate whether TRACER generalizes to *item-level* forgetting, where individual items are designated for removal. Table 6 reports results on the Industrial & Scientific dataset. TRACER consistently achieves the lowest forget leakage across all five backbones while maintaining competitive retain performance and high semantic coherence, confirming that our SID-code reassignment strategy is not limited to brand-level forgetting but extends naturally to arbitrary item-level unlearning requests [57].

Table 3: Complete unlearning results on **Sports & Outdoors**. Bold = best among unlearning methods; underline = second best. Retain metrics (↑): higher is better; Forget metrics (↓): lower is better; Sim. (↑): higher is better.

Backbone	Method	Retain (↑) HR@5	HR@10	NDCG@5	NDCG@10	MRR	Forget (↓) HR@5	HR@10	NDCG@5	NDCG@10	MRR	Sim.↑
MOR	Original	0.078	0.096	0.067	0.073	0.068	0.090	0.096	0.063	0.065	0.056	0.840
	RMU	0.037	0.068	0.033	0.057	0.059	0.078	0.075	0.054	0.054	0.051	0.720
	ELM	0.051	0.079	0.044	0.061	0.062	0.074	0.074	0.055	0.048	0.047	0.812
	BLUR	0.072	0.093	**0.065**	0.070	0.066	0.073	0.071	0.047	0.052	0.048	0.666
	REPNoise	0.058	0.052	0.037	0.057	0.061	0.069	0.067	0.044	0.049	0.044	0.626
	FLAT	0.074	0.090	0.062	0.065	0.065	0.059	0.068	0.039	0.042	0.044	0.693
	PISCES	0.072	0.088	0.063	0.071	0.063	0.052	0.056	0.038	0.039	0.040	0.773
	TRACER	**0.078**	**0.094**	**0.065**	**0.072**	**0.067**	**0.044**	**0.046**	**0.034**	**0.033**	**0.036**	**0.825**
TIGER	Original	0.065	0.082	0.059	0.068	0.060	0.078	0.084	0.055	0.059	0.052	0.867
	RMU	0.036	0.062	0.037	0.054	0.055	0.070	0.065	0.042	0.049	0.043	0.734
	ELM	0.040	0.062	0.048	0.044	0.056	0.057	0.068	0.045	0.044	0.044	0.819
	BLUR	0.059	**0.078**	0.058	0.065	0.058	0.058	0.061	0.043	0.043	0.044	0.611
	REPNoise	0.039	0.064	0.048	0.042	0.054	0.057	0.058	0.042	0.039	0.039	0.644
	FLAT	0.058	**0.078**	0.051	0.064	0.058	0.049	0.056	0.034	0.038	0.036	0.759
	PISCES	0.061	0.071	0.057	0.059	0.060	0.042	0.049	0.032	0.034	0.037	0.847
	TRACER	**0.063**	**0.078**	**0.059**	**0.068**	**0.061**	**0.031**	**0.036**	**0.028**	**0.030**	**0.032**	**0.849**
P5SID	Original	0.065	0.080	0.062	0.067	0.060	0.079	0.084	0.056	0.058	0.051	0.831
	RMU	0.033	0.058	0.035	0.050	0.051	0.063	0.073	0.041	0.047	0.046	0.708
	ELM	0.039	0.061	0.052	0.045	0.056	0.065	0.060	0.043	0.045	0.043	0.798
	BLUR	0.061	0.075	**0.060**	0.065	0.058	0.056	0.059	0.046	0.044	0.042	0.649
	REPNoise	0.038	0.059	0.039	0.054	0.054	0.054	0.063	0.037	0.039	0.040	0.618
	FLAT	0.055	0.073	0.055	0.057	0.054	0.049	0.055	0.035	0.034	0.035	0.681
	PISCES	0.057	0.075	0.055	0.064	0.057	0.042	0.049	0.032	0.035	0.034	0.762
	TRACER	**0.063**	**0.079**	**0.060**	**0.066**	**0.059**	**0.033**	**0.037**	**0.026**	**0.030**	**0.030**	**0.825**
LETTER	Original	0.081	0.097	0.069	0.076	0.068	0.103	0.108	0.071	0.069	0.063	0.821
	RMU	0.061	0.064	0.031	0.035	0.059	0.085	0.099	0.057	0.063	0.059	0.725
	ELM	0.053	0.079	0.057	0.053	0.060	0.082	0.093	0.057	0.059	0.055	0.802
	BLUR	0.076	0.092	0.065	0.073	0.065	0.079	0.094	0.056	0.056	0.052	0.713
	REPNoise	0.062	0.055	0.050	0.042	0.061	0.070	0.073	0.055	0.053	0.051	0.681
	FLAT	0.071	0.087	0.066	0.066	0.064	0.078	0.075	0.054	0.048	0.051	0.686
	PISCES	0.077	0.088	0.063	0.073	0.065	0.061	0.071	0.045	0.044	0.049	0.782
	TRACER	**0.078**	**0.093**	**0.067**	**0.074**	**0.066**	**0.054**	**0.057**	**0.041**	**0.038**	**0.042**	**0.821**
ETEGRec	Original	0.080	0.100	0.069	0.075	0.067	0.109	0.113	0.074	0.068	0.064	0.848
	RMU	0.040	0.071	0.050	0.043	0.059	0.089	0.096	0.064	0.061	0.060	0.727
	ELM	0.065	0.069	0.057	0.053	0.060	0.095	0.093	0.060	0.059	0.056	0.814
	BLUR	0.078	0.099	0.067	0.070	0.064	0.091	0.087	0.061	0.057	0.054	0.637
	REPNoise	0.057	0.052	0.050	0.036	0.056	0.077	0.085	0.055	0.052	0.050	0.631
	FLAT	0.072	0.092	0.065	0.066	0.064	0.082	0.081	0.058	0.053	0.051	0.716
	PISCES	0.069	0.093	0.060	0.069	0.060	0.073	0.073	0.047	0.044	0.047	0.798
	TRACER	**0.079**	**0.100**	**0.068**	**0.074**	**0.066**	**0.059**	**0.061**	**0.044**	**0.040**	**0.043**	**0.841**

D.5 Effect of Positive Set Size K and Temperature τ

Effect of K. The hyperparameter K controls the number of semantically similar retain items used to construct the positive set $\mathcal{P}(i)$ in the coherence loss $\mathcal{L}_{\text{Coh}}$ (Eq. 9). A smaller K limits the diversity of the coherence signal, resulting in lower semantic similarity after reassignment. Conversely, a larger K introduces less relevant items into the positive set, diluting the coherence objective and slightly degrading both unlearning effectiveness and similarity. As shown in Figure 5 (left), K=5 achieves the best trade-off, yielding the highest semantic similarity (0.828) while maintaining strong forget performance (Forget HR@5 = 0.058) and retain quality (Retain HR@5 = 0.096).

Effect of τ. The temperature τ in Eq. 6 governs the sharpness of the soft codeword-assignment distribution during token reassignment. A very small τ ($e.g.$, 0.003) makes the distribution excessively peaked, reducing the flexibility of the reassignment and limiting the model's ability to find alternative semantic codes for the forget item. A very large τ ($e.g.$, 0.009) makes the distribution nearly uniform, causing the model to scatter probability mass across irrelevant codewords and degrading semantic coherence. As shown in Figure 5 (right), τ=0.005 achieves the best balance, yielding the lowest forget leakage (Forget HR@5 = 0.056), the highest retain quality (Retain HR@5 = 0.097), and the strongest semantic coherence (Sim = 0.828). τ=0.007 is a close second, while τ=0.009 is the worst-performing setting across all metrics, confirming that over-softening the assignment distribution is more harmful than over-sharpening it.

Table 4: Complete unlearning results on **Toys & Games**. Bold = best among unlearning methods; underline = second best. Retain metrics (↑): higher is better; Forget metrics (↓): lower is better; Sim. (↑): higher is better.

Backbone	Method	Retain (↑) HR@5	HR@10	NDCG@5	NDCG@10	MRR	Forget (↓) HR@5	HR@10	NDCG@5	NDCG@10	MRR	Sim.↑
MOR	Original	0.049	0.065	0.037	0.042	0.036	0.055	0.077	0.037	0.044	0.036	0.928
	RMU	0.034	0.033	0.027	0.019	0.033	0.044	0.067	0.029	0.036	0.031	0.794
	ELM	0.032	0.052	0.025	0.034	0.035	0.043	0.063	0.030	0.036	0.033	0.907
	BLUR	0.045	0.062	0.035	0.040	0.034	0.041	0.062	0.029	0.034	0.032	0.749
	REPNoise	0.037	0.046	0.018	0.024	0.033	0.040	0.058	0.027	0.031	0.030	0.696
	FLAT	0.048	0.063	0.036	0.038	0.035	0.035	0.055	0.026	0.028	0.028	0.775
	PISCES	0.044	0.060	**0.037**	0.038	**0.036**	0.032	0.047	0.026	0.028	0.028	0.866
	TRACER	**0.049**	**0.064**	**0.037**	**0.041**	**0.036**	**0.026**	**0.036**	**0.021**	**0.024**	**0.024**	**0.928**
TIGER	Original	0.039	0.054	0.032	0.038	0.032	0.048	0.070	0.032	0.042	0.031	0.951
	RMU	0.030	0.040	0.017	0.023	0.028	0.043	0.052	0.025	0.033	0.025	0.817
	ELM	0.031	0.033	0.021	0.032	0.028	0.035	0.057	0.022	0.033	0.026	0.922
	BLUR	0.038	0.052	0.028	0.037	0.031	0.035	0.057	0.026	0.030	0.026	0.688
	REPNoise	0.029	0.030	0.018	0.029	0.028	0.037	0.045	0.024	0.029	0.024	0.716
	FLAT	0.037	0.049	0.030	0.034	0.032	0.028	0.048	0.019	0.026	0.023	0.845
	PISCES	0.033	0.052	0.030	0.036	0.030	0.026	0.040	0.019	0.025	0.023	0.951
	TRACER	**0.039**	**0.054**	**0.031**	**0.038**	**0.033**	**0.019**	**0.031**	**0.016**	**0.020**	**0.019**	**0.963**
P5SID	Original	0.039	0.055	0.031	0.040	0.034	0.046	0.065	0.032	0.039	0.034	0.918
	RMU	0.032	0.042	0.016	0.023	0.030	0.040	0.058	0.025	0.032	0.029	0.782
	ELM	0.032	0.032	0.025	0.025	0.032	0.033	0.054	0.025	0.031	0.028	0.893
	BLUR	0.038	0.051	**0.031**	0.040	0.034	0.034	0.054	0.024	0.029	0.027	0.737
	REPNoise	0.024	0.040	0.018	0.030	0.029	0.030	0.048	0.023	0.029	0.028	0.688
	FLAT	0.034	0.051	0.028	0.038	0.032	0.029	0.041	0.020	0.025	0.027	0.762
	PISCES	0.038	0.047	0.028	0.038	0.033	0.025	0.036	0.018	0.024	0.027	0.856
	TRACER	**0.039**	**0.056**	**0.031**	**0.041**	**0.035**	**0.020**	**0.030**	**0.015**	**0.020**	**0.023**	**0.918**
LETTER	Original	0.050	0.067	0.039	0.042	0.039	0.062	0.086	0.039	0.046	0.040	0.911
	RMU	0.024	0.046	0.028	0.021	0.033	0.059	0.067	0.033	0.041	0.037	0.787
	ELM	0.035	0.054	0.026	0.034	0.034	0.051	0.074	0.029	0.039	0.034	0.908
	BLUR	0.047	**0.065**	0.040	0.042	0.038	0.053	0.066	0.033	0.038	0.034	0.801
	REPNoise	0.029	0.049	0.028	0.023	0.035	0.047	0.058	0.026	0.034	0.034	0.762
	FLAT	0.046	0.061	0.036	0.035	0.034	0.041	0.067	0.029	0.033	0.032	0.776
	PISCES	0.044	0.062	0.036	0.039	0.038	0.038	0.058	0.026	0.029	0.033	0.890
	TRACER	**0.050**	**0.065**	**0.041**	**0.043**	**0.039**	**0.034**	**0.042**	**0.022**	**0.025**	**0.029**	**0.917**
ETEGRec	Original	0.050	0.068	0.037	0.044	0.038	0.066	0.090	0.045	0.046	0.042	0.938
	RMU	0.027	0.049	0.018	0.035	0.032	0.056	0.073	0.040	0.041	0.037	0.803
	ELM	0.040	0.046	0.032	0.033	0.034	0.051	0.075	0.040	0.042	0.038	0.914
	BLUR	0.047	0.066	0.034	0.044	0.037	0.053	0.068	0.036	0.038	0.035	0.758
	REPNoise	0.025	0.048	0.027	0.022	0.031	0.049	0.067	0.032	0.036	0.033	0.712
	FLAT	0.046	0.065	0.033	0.041	0.035	0.052	0.063	0.034	0.033	0.035	0.788
	PISCES	0.044	0.063	0.031	0.038	0.034	0.043	0.056	0.030	0.032	0.033	0.879
	TRACER	**0.051**	**0.067**	**0.038**	**0.045**	**0.039**	**0.034**	**0.049**	**0.027**	**0.027**	**0.029**	**0.938**

Table 5: Unlearning time (minutes, mean ± std over 5 runs) on 4× A100 GPUs. **Bold**: fastest; underline: second fastest.

Method	Ind. & Sci.		Spo. & Out.	
	MOR	LETTER	MOR	LETTER
RMU	67.5±1.8	51.3±1.4	63.8±1.7	48.6±1.3
ELM	35.2±0.9	26.8±0.7	33.1±0.8	25.4±0.6
REPNoise	43.5±1.1	33.1±0.8	41.2±1.0	31.5±0.7
FLAT	51.2±1.3	39.4±1.0	48.6±1.2	37.2±0.9
BLUR	47.5±1.2	36.2±0.9	44.7±1.1	34.1±0.8
PISCES	41.6±1.0	31.7±0.8	39.8±0.9	30.2±0.7
TRACER	**26.7**±0.5	**20.3**±0.4	**24.9**±0.5	**19.1**±0.4

D.6 Concrete Examples

To provide qualitative insight into how TRACER modifies recommendation behavior, we present top-10 recommendation lists produced by the base model and by TRACER for three representative forget-brand queries, one from each dataset. Items belonging to the forget set are highlighted with red shading; the "Forget Leaked" row counts how many of the 10 retrieved items originate from the target brand's forget set. As shown in Table 7, the base model frequently surfaces memorized forget-set items (up to 9 out of 10), whereas TRACER consistently achieves 0/10 leakage while returning semantically relevant alternatives from retain brands.

Table 6: Item-level unlearning results on **Industrial & Scientific**. Bold = best among unlearning methods; underline = second best. Retain (↑); Forget (↓); Sim. (↑).

Backbone	Method	Retain (↑) HR@5	HR@10	NDCG@5	NDCG@10	MRR	Forget (↓) HR@5	HR@10	NDCG@5	NDCG@10	MRR	Sim.↑
MOR	Original	0.115	0.162	0.083	0.098	0.085	0.206	0.256	0.167	0.187	0.167	0.866
	RMU	0.080	0.080	0.061	0.049	0.075	0.164	0.226	0.142	0.155	0.148	0.735
	ELM	0.077	0.131	0.056	0.081	0.078	0.159	0.215	0.130	0.151	0.144	0.817
	BLUR	0.110	0.156	**0.081**	0.096	**0.081**	0.156	0.211	0.125	0.146	0.141	0.663
	REPNoise	0.083	0.086	0.046	0.075	0.076	0.152	0.183	0.124	0.131	0.134	0.622
	FLAT	0.111	0.155	**0.081**	0.087	0.079	0.145	0.166	0.109	0.124	0.124	0.703
	PISCES	0.108	0.150	0.074	0.093	0.079	0.121	0.143	0.100	0.108	0.117	0.799
	TRACER	**0.114**	**0.157**	**0.081**	**0.098**	**0.081**	**0.097**	**0.122**	**0.089**	**0.097**	**0.106**	**0.834**
TIGER	Original	0.095	0.134	0.076	0.092	0.074	0.173	0.227	0.152	0.172	0.149	0.833
	RMU	0.066	0.063	0.057	0.049	0.066	0.134	0.202	0.130	0.144	0.134	0.734
	ELM	0.062	0.108	0.054	0.078	0.069	0.133	0.189	0.119	0.138	0.129	0.831
	BLUR	**0.092**	0.132	0.072	0.090	0.072	0.130	0.187	0.111	0.136	0.128	0.661
	REPNoise	0.068	0.072	0.041	0.074	0.067	0.128	0.160	0.111	0.120	0.122	0.630
	FLAT	0.090	0.129	**0.074**	0.083	0.071	0.123	0.145	0.096	0.114	0.111	0.699
	PISCES	0.088	0.124	0.067	0.091	0.071	0.102	0.126	0.091	0.097	0.104	0.796
	TRACER	**0.092**	**0.133**	**0.074**	**0.093**	**0.073**	**0.083**	**0.107**	**0.080**	**0.092**	**0.095**	**0.833**
P5SID	Original	0.095	0.139	0.075	0.093	0.074	0.177	0.219	0.152	0.170	0.147	0.868
	RMU	0.069	0.067	0.055	0.049	0.065	0.140	0.193	0.131	0.142	0.131	0.738
	ELM	0.064	0.114	0.052	0.077	0.068	0.138	0.185	0.116	0.137	0.127	0.827
	BLUR	0.089	0.133	0.072	0.093	0.071	0.137	0.178	0.113	0.133	0.123	0.664
	REPNoise	0.070	0.075	0.043	0.071	0.066	0.132	0.156	0.112	0.118	0.118	0.635
	FLAT	0.092	0.132	0.075	0.085	0.069	0.124	0.141	0.097	0.114	0.111	0.730
	PISCES	0.089	0.127	0.070	0.091	0.068	0.103	0.121	0.093	0.097	0.103	0.776
	TRACER	**0.093**	**0.138**	**0.076**	**0.094**	**0.072**	**0.087**	**0.104**	**0.079**	**0.089**	**0.095**	**0.829**
LETTER	Original	0.119	0.165	0.086	0.102	0.087	0.231	0.288	0.185	0.201	0.184	0.847
	RMU	0.084	0.081	0.066	0.054	0.077	0.182	0.256	0.159	0.165	0.163	0.735
	ELM	0.080	0.134	0.058	0.085	0.080	0.180	0.242	0.143	0.162	0.160	0.829
	BLUR	0.112	0.161	0.083	0.100	0.083	0.175	0.234	0.135	0.159	0.156	0.662
	REPNoise	0.089	0.089	0.050	0.080	0.076	0.172	0.203	0.135	0.139	0.151	0.625
	FLAT	**0.114**	0.156	0.083	0.093	0.083	0.163	0.184	0.117	0.136	0.137	0.715
	PISCES	**0.114**	0.151	0.077	0.098	0.080	0.138	0.162	0.110	0.116	0.126	0.790
	TRACER	**0.114**	**0.164**	**0.084**	**0.101**	**0.084**	**0.109**	**0.136**	**0.096**	**0.106**	**0.119**	**0.842**
ETEGRec	Original	0.117	0.163	0.084	0.105	0.084	0.249	0.297	0.200	0.200	0.195	0.856
	RMU	0.083	0.081	0.065	0.052	0.073	0.196	0.264	0.172	0.165	0.172	0.714
	ELM	0.077	0.131	0.059	0.090	0.075	0.193	0.249	0.152	0.160	0.171	0.807
	BLUR	0.112	0.160	0.079	0.105	0.082	0.191	0.245	0.147	0.160	0.165	0.688
	REPNoise	0.086	0.088	0.049	0.083	0.074	0.187	0.213	0.146	0.141	0.160	0.613
	FLAT	0.111	0.154	0.082	0.097	0.079	0.176	0.193	0.126	0.133	0.147	0.723
	PISCES	0.112	0.150	0.075	0.102	0.081	0.148	0.167	0.119	0.116	0.136	0.798
	TRACER	**0.113**	**0.161**	**0.083**	**0.106**	**0.083**	**0.119**	**0.140**	**0.107**	**0.104**	**0.127**	**0.811**

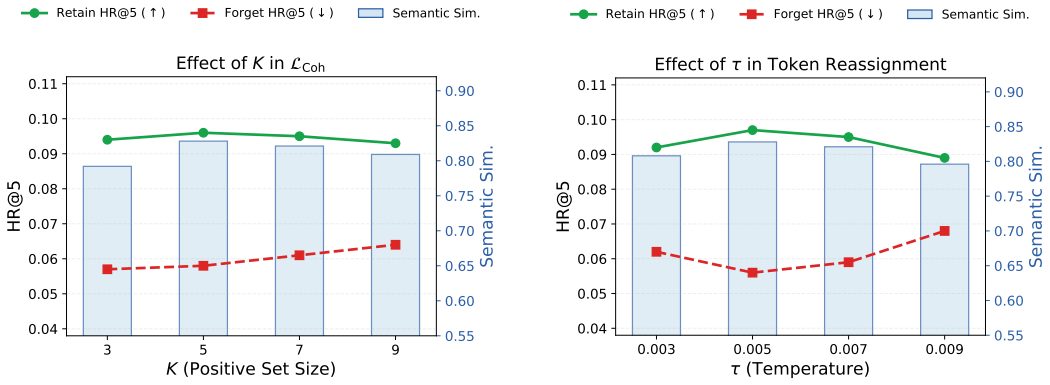


Figure 5: Ablation studies on K (left) and τ (right). **Left:** K=5 achieves the best trade-off between semantic coherence and unlearning effectiveness. **Right:** τ=0.005 yields the strongest performance; larger τ gradually degrades all metrics as the token-reassignment distribution becomes too diffuse.

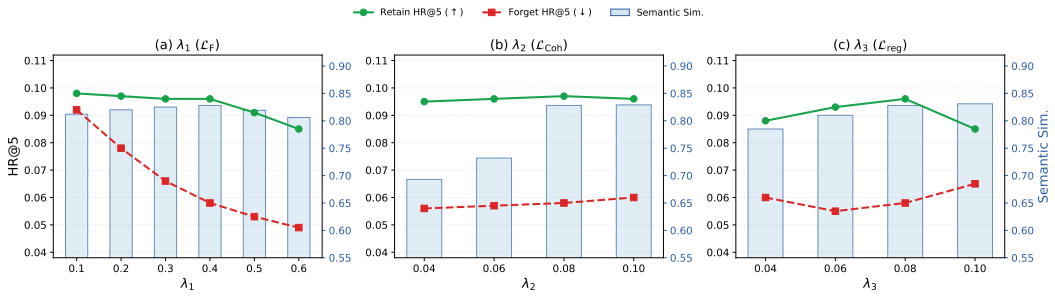


Figure 6: Ablation study on loss weights. (a) λ_1 controls the forget loss $\mathcal{L}_{\mathrm{F}}$: increasing λ_1 improves forgetting but degrades retain performance when too large. (b) λ_2 controls the coherence loss $\mathcal{L}_{\mathrm{Coh}}$: higher values improve semantic similarity with diminishing returns. (c) λ_3 controls the regularization loss $\mathcal{L}_{\mathrm{reg}}$: stronger regularization preserves retain quality but weakens forgetting.

Table 7: Qualitative Example 1 (Industrial & Scientific). Target: **Anderson Metals** — "Anderson Metals Brass Hose Fitting, Tee, 5/16" x 5/16" x 5/16" Barb". Original model leaks 5/10 forget items (rank = 1); TRACER: 0/10.

User Interaction History (most recent 5 items)		
1	J-B Weld 8276 KwikWeld Quick Setting Steel Reinforced Epoxy - 2 oz.	
2	J-B Weld 8265S Original Cold-Weld Steel Reinforced Epoxy - 2 oz.	
3	Accessbuy 180-Piece Rubber Grommet Shop Assortment	
4	Plymor 6" x 9", 4 Mil (Pack of 100) Heavy Duty Plastic Reclosable Zipper Bags	
5	Loctite Liquid Professional Super Glue 20-Gram Bottle (1365882)	
Rank	**Original (Base Model)**	**TRACER (Ours)**
1	**Anderson Metals**: Anderson Metals Brass Hose Fitting, Tee, 5/16"	Dremel: Dremel 3D Printing BT20-01 Build Sheets (Pack of 3)
2	Neiko: Neiko 01412A Electronic Digital Caliper	Pro Tapes: ProTapes Pro Duct 110 PE-Coated Cloth General Purpose
3	Merit Brass: Stainless Steel 304 Cast Pipe Fitting, Hex Locknut	Anycubic: Anycubic All Metal Right Hand MK8 Extruder
4	Pomona: Pomona 1296 BNC (M) Binding Posts	Scotch: Scotch 1110-C Multi Use Duct Tape, 10-Yards
5	Dremel: Dremel 3D Printing BT20-01 Build Sheets (Pack of 3)	AcuRite: AcuRite 00613 Humidity Monitor with Indoor Thermometer
6	**Loctite**: Loctite 40140 Clear 401 Prism Surface Insensitive	Camco: Camco 4ft TastePURE Drinking Water Hose
7	**Gardner Bender**: Gardner Bender GHG-1538 5PK 3/8" Grommet, Black	Crest: Pro-Health Clean Mint Toothpaste 6 oz. (Pack of 4)
8	Pro Tapes: ProTapes Pro Duct 110 PE-Coated Cloth General Purpose	Extech: Extech TG20 Wire Tracer and Tone Generator
9	**DEWALT**: DEWALT DCT418S1 12V Li-Ion Hand Held Radar Scanner	Forney: Forney 72391 Replacement Cutters for Bench Grinding Wheel
10	**DAP**: Dap 18172 10.1 Oz Alex Plus Acrylic Latex Caulk	Cambridge: Cambridge Hose Clamps SAE Size 16, 10 Pcs.
Forget	**5/10**	**0/10**

Table 8: Qualitative Example 2 (Sports & Outdoors). Target: **Speedo** — "Speedo Vanquisher 2.0 Mirrored Swim Goggle". Original model leaks 9/10 forget items (rank = 3); TRACER: 0/10.

User Interaction History (most recent 5 items)		
1	WOTOW 16 in 1 Multi-Function Bike Bicycle Cycling Mechanic Repair Tool	
2	AmazonBasics 1/4-Inch Yoga and Exercise Mat with Carrying Strap	
3	Speedo Vanquisher 2.0 Mirrored Swim Goggle	
4	Speedo Vanquisher 2.0 Mirrored Swim Goggle	
5	Speedo Vanquisher 2.0 Mirrored Swim Goggle	
Rank	**Original (Base Model)**	**TRACER (Ours)**
1	**Speedo**: Speedo Short Blade Swim Training Fins	Duduma: Duduma Polarized Sports Sunglasses for Men
2	**Speedo**: Speedo Silicone Long Hair Swim Cap	HODGSON: HODGSON Polarized Sports Sunglasses with 5 Lenses
3	**Speedo**: Speedo Vanquisher 2.0 Mirrored Swim Goggle	Duduma: Duduma Polarized Designer Fashion Sports Sunglasses
4	**Speedo**: Speedo Deluxe Ventilator Mesh Bag	Zionor: Zionor X4 Ski Snowboard Snow Goggles Magnet Dual Layers
5	**Speedo**: Speedo Ventilator Mesh Equipment Bag	DRSKIN: DRSKIN Compression Cool Dry Sports Tights Shirt
6	**Speedo**: Speedo Silicone Solid Swim Cap	COSVER: COSVER Men's Polarized Sunglasses for Men Sports
7	**Speedo**: Speedo Kids' Skoogles Swim Goggle	Baleaf: Baleaf Men's Running Fitness Workout Compression Pants
8	**Speedo**: Speedo Men's Endurance+ Launch Splice Jammer	Tesla: Tesla Men's Cool Dry Compression Muscle Tank
9	**Speedo**: Speedo Women's Pro LT Super Pro Swimsuit	LUENX: LUENX Men Aviator Sunglasses Polarized Women
10	adidas: adidas Men's Tiro 15 Training Pants	DRSKIN: DRSKIN Men's 3/4 Compression Tight Pants
Forget	**9/10**	**0/10**

Table 9: Qualitative Example 3 (Toys & Games). Target: **Learning Resources** — "Learning Resources New Sprouts Reel it". Original model leaks 8/10 forget items (not in top-20); TRACER: 0/10.

User Interaction History (most recent 5 items)	
1	Green Toys Airplane Vehicle Toy, Purple, 8.5" x 9" x 4.5"
2	Fisher-Price Go Baby Go Poppity-Pop Musical Dino
3	Learning Resources New Sprouts Stir Fry Set
4	VTech Lil' Critters Roll and Discover Ball
5	Learning Resources Pretend & Play Doctor Kit for Kids, 19 Piece Set

Rank	**Original (Base Model)**	**TRACER (Ours)**
1	Peppa Pig: Peppa Pig 92602 Fancy Dress Party Toy Figure	Peppa Pig: Peppa Pig 92602 Fancy Dress Party Toy Figure
2	Peppa Pig: Peppa Pig Character Options Classroom Playset	Peppa Pig: Peppa Pig Character Options Classroom Playset
3	**Paw Patrol**: Paw Patrol, Lights and Sounds Air Patroller Plane	Fisher-Price: Fisher-Price Think & Learn Code-a-Pillar Toy
4	**Paw Patrol**: Paw Patrol Super Pup Rubble's Crane, Vehicle and Figure	Fisher-Price: Fisher-Price DoodlePro, Trip, (Red)
5	**Paw Patrol**: Paw Patrol - Paw Terrain Vehicle (Amazon Exclusive)	Fisher-Price: Fisher-Price Train Inflatable Ball Pit
6	**Paw Patrol**: Paw Patrol Action Pack Pups 3pk Figure Set	Fisher-Price: Fisher-Price Bright Beats Dance & Move BeatBo
7	**Paw Patrol**: Paw Patrol, Adventure Bay Railway Track Set	Fisher-Price: Fisher-Price Bright Beats Smart Touch Play Space
8	**Paw Patrol**: Paw Patrol Chase's Cruiser, Vehicle and Figure	Fisher-Price: Fisher-Price Laugh & Learn Smart Stages Piggy Bank
9	**Paw Patrol**: Paw Patrol Chase's Spy Cruiser, Vehicle and Figure	VTech: VTech Swim & Spray Musical Dolphin
10	**Paw Patrol**: Paw Patrol Racers 3-Pack Vehicle Set, Rubble/Rocky/Skye	Fisher-Price: Fisher-Price Laugh & Learn Let's Get Ready Sink
Forget	**8/10**	**0/10**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We state the scope of the results in the abstract and introduction part.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We state the assumptions, and for each Theorem and Lemma, we state the assumptions under which it holds.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We state all the training hyperparameters in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release the source code and trained checkpoints upon acceptance. The evaluation datasets (Amazon Reviews) are publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all details and necassary settings for the experiments in section 6 and appendix C

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All main results report mean $\pm$ standard deviation over 5 independent runs (Tables 1–2). Ablation studies follow the same protocol.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the hardware setup and wall-clock unlearning time for all methods in the appendix (Table 5).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and conform to the NeurIPS Code of Ethics

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impacts in Appendix A.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper does not contain safety issues.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original papers for all datasets (Amazon Reviews), backbone models (Qwen2.5, T5), and baseline methods. All assets are publicly available for research use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not introduce new datasets or pretrained model assets. The contribution is a method (TRACER) evaluated on existing public benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core results in this paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.