AdaSRU: Adaptive Source-Free Recommendation Unlearning via Gradient-Constrained Optimization

Xiaoran Zhao[1,*], **Weiming Liu**[2,*], **Lianyong Qi**[1,3,†], **Weiyi Zhong**[4], **Yuwen Liu**[5], **Xiaolong Xu**[6], **Haolong Xiang**[6], **Qiang Ni**[7], **Xuyun Zhang**[8], **Wanchun Dou**[9], **Xiaokang Zhou**[10,11]

[1] College of Computer Science and Technology, China University of Petroleum (East China)
[2] ByteDance Inc. [3] Shandong Key Laboratory of Intelligent Oil & Gas Industrial Software
[4] School of Information Engineering and Technology,
Qingdao Vocational and Technical College of Hotel Management
[5] Shandong Computer Science Center (National Supercomputer Center in Jinan),
Qilu University of Technology (Shandong Academy of Sciences)
[6] School of Software, Nanjing University of Information Science and Technology
[7] School of Computing & Communications, Lancaster University
[8] School of Computing, Macquarie University
[9] State Key Laboratory for Novel Software Technology,
School of Computer Science, Nanjing University
[10] Faculty of Business and Data Science, Kansai University [11] RIKEN Center for Advanced Intelligence Project

xiaoranzhao@s.upc.edu.cn, lwming95@gmail.com, lianyongqi@upc.edu.cn,
weiyizhong999@gmail.com, liuyw@sdas.org, xlxu@nuist.edu.cn, hlxiang@nuist.edu.cn,
q.ni@lancaster.ac.uk, xuyun.zhang@mq.edu.au, douwc@nju.edu.cn, zhou@kansai-u.ac.jp

Abstract

Recommendation unlearning (RU) aims to remove the influence of specified user–item interactions from a trained recommender while preserving utility on the remaining data. Existing RU methods usually require access to the original training data, especially the remaining data, to recover recommendation utility after unlearning. However, access to such data is often restricted or unavailable in practice due to privacy constraints, storage costs, or third-party deployment, giving rise to source-free recommendation unlearning. To address this challenge, we propose an adaptive source-free recommendation unlearning (**AdaSRU**) method that enables effective unlearning without accessing the original remaining data. **AdaSRU** first estimates the inaccessible gradient on the remaining data. It exploits the first-order optimality condition of the trained recommender and approximates the remaining gradient using the forgetting data and squared gradient accumulators from adaptive optimizers, e.g., Adam. **AdaSRU** then formulates unlearning as a gradient-constrained update, where the estimated remaining gradient is used as a constraint to limit utility degradation while maximizing the unlearning objective. **AdaSRU** adaptively adjusts the utility-preserving correction according to the current gradient geometry. We further theoretically prove that **AdaSRU** approaches an approximate Pareto stationary solution. Extensive experiments on three real-world datasets demonstrate that **AdaSRU** achieves strong performance in both unlearning and recommendation. The code is available at https://github.com/xiaoran-zhao/AdaSRU.

1 Introduction

Recommender systems have become increasingly important in modern online services as data grow at an explosive scale [53, 39, 37, 34, 47]. They infer user preferences by leveraging large-scale user–item interaction data, e.g., *clicks, ratings, and browsing histories*. However, such behavioral data

[*] Equal contribution; [†] Corresponding author.

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

may expose sensitive preferences and private attributes, raising users' privacy concerns [55, 60, 36]. Meanwhile, privacy regulations, including the General Data Protection Regulation (GDPR) [43] and the California Consumer Privacy Act (CCPA) [24], grant users the right to request the deletion of their personal data, i.e., *the right to be forgotten*. These privacy and regulatory requirements have motivated recommendation unlearning [64, 29], which aims to remove the influence of requested interactions from a trained recommender while preserving its recommendation utility on the remaining data.

Most existing recommendation unlearning methods assume access to the original training data during unlearning [31, 54, 32, 35]. However, this assumption may fail in many practical recommendation scenarios, where user interactions are continuously updated, historical records may be overwritten, large-scale logs may be deleted after training to reduce privacy risks or storage costs, and deployed recommenders may be trained by third-party providers, leaving the model owner without access to the original training data [25, 45]. Thus, unlearning methods that rely on the original training data become infeasible to apply. This gives rise to *Source-Free Recommendation Unlearning* (SRU), which aims to remove the influence of specified interactions without accessing the original remaining data. There are two main challenges in solving SRU: (1) **CH1**: How to estimate utility-preserving signals for recommendation without access to the original remaining data? (2) **CH2**: How to jointly optimize the inherently conflicting unlearning and utility objectives to achieve a balanced solution?

However, existing methods remain limited in solving the SRU problem. First, few studies have tried unlearning without access to the original training data [1]. For example, [11] employs knowledge distillation to mitigate forgotten samples; [2] generates surrogate data to replace the original data, incurring substantial storage overhead; and [4] uses projected gradient descent to create adversarial samples for the forget classes. These methods are mostly designed for image classification, where unlearning objectives are typically defined over labels and Euclidean feature representations. They do not directly apply to recommendation unlearning, whose utility relies on sparse user–item interactions and collaborative signals [15, 22]. Meanwhile, utility preservation trades off against unlearning objectives [63]. Existing methods often employ a retaining loss during unlearning, computed over retained training data to mitigate utility degradation [20, 42, 10]. Such losses still require access to remaining data, which are unavailable in the source-free setting. Moreover, jointly optimizing the unlearning and retaining objectives requires balancing their competing gradients.

To address these challenges, we propose an ***Ada**ptive **S**ource-Free **R**ecommendation **U**nlearning* (**AdaSRU**) method, which approximates the gradient on the remaining data and adaptively balances unlearning with utility preservation. Specifically, **AdaSRU** leverages the first-order optimality of the original recommender to relate the remaining-data gradient to the forgetting-data gradient, and further uses a Taylor expansion to extend this relation along the unlearning trajectory. It then constructs a computable proxy gradient using the forgetting data and squared gradient accumulators from adaptive optimizers (e.g., Adam), implicitly preserving the collaborative signals learned during pretraining. This provides the insight that *the remaining-data gradient can be approximated without accessing the remaining data*. Subsequently, we treat unlearning as a gradient-constrained update and use the estimated remaining-data gradient as a constraint to maximize the unlearning objective while bounding utility loss, thus dynamically balancing utility and unlearning in the source-free setting. We further provide robust theoretical guarantees, thereby ensuring reliability.

We summarize our main contributions as follows: (1) We propose a novel method, namely **AdaSRU**, which estimates the inaccessible gradient on the remaining data using forgetting data and squared gradient accumulators from adaptive optimizers. (2) We design an adaptive gradient-constrained unlearning update, dynamically integrating the proxy gradient to control utility degradation while maximizing the unlearning effect, achieving a practical trade-off between utility and unlearning. (3) We conduct extensive experiments on real-world datasets to validate the performance of **AdaSRU**.

2 Related Works

Recommendation Unlearning. Recommendation unlearning aims to remove the influence of specified interactions from a trained recommender while preserving utility on the remaining data [32, 5, 59]. Existing methods fall into two paradigms: Exact Unlearning (EU) and Approximate Unlearning (AU). EU methods eliminate the influence of target data through retraining [3]. For example, SISA-based methods [8, 28, 26] partition data into shards and retrain only affected submodels upon deletion requests. In contrast, AU methods remove the influence of target data by directly updating model

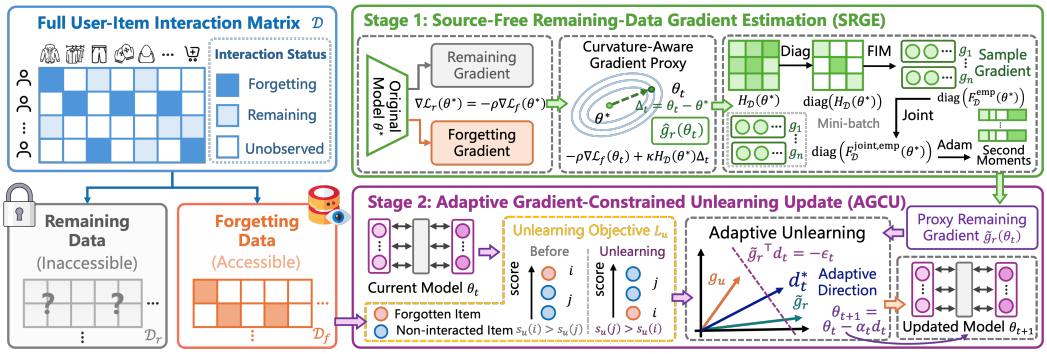


Figure 1: The overall framework of our proposed **AdaSRU**.

parameters [61, 58, 38, 21]. For instance, [9] generates unlearned parameters via an influence encoder, while [57] employs Fisher information to estimate parameter importance for adjustment. However, these methods require accessible remaining data to recover utility. Thus, ensuring sufficient unlearning while preserving remaining-data utility is challenging without the original training data.

Source-Free Machine Unlearning. Source-free unlearning aims to remove target-data influence without accessing the original training data [50, 56, 14, 12]. This setting is practical under privacy constraints yet remains challenging, and existing studies on it are still scarce. Recently, [2] uses surrogate data to approximate the source distribution and calibrates noise accordingly, yet this entails substantial data generation and storage overhead. Meanwhile, [1] approximates the unlearning update using perturbed points near local minima and only the forget set. DSDA [49] combines energy-guided data synthesis with discrimination-aware multitask optimization to mitigate utility degradation. However, these methods mainly target image tasks and are difficult to apply to recommendation scenarios, where sparse user–item interactions [13, 62] and collaborative signals [17, 41] are critical.

3 Methodology

In this section, we first formulate the problem of source-free recommendation unlearning. We then introduce the two key stages of **AdaSRU** (Fig. 1). The first stage estimates the inaccessible remaining-data gradient from the forgetting data, the trained model, and the squared gradient accumulators from adaptive optimizers. The second stage performs an adaptive gradient-constrained update that balances unlearning effectiveness and utility preservation without using fixed linear weights.

3.1 Problem Definition

Let $\mathcal{D} = \{(u, i, y_{ui}) \mid u \in \mathcal{U}, i \in \mathcal{I}\}$ denote the user–item interaction dataset, where $\mathcal{U}$ and $\mathcal{I}$ are the user and item sets, respectively. The binary variable $y_{ui} \in \{0, 1\}$ indicates whether user u has interacted with item i. A recommendation model f_θ, parameterized by θ, learns user and item embeddings $e_u, e_i \in \mathbb{R}^d$, and predicts the preference score as $\hat{y}_{ui} = e_u^\top e_i$. The pretrained model parameters θ^* are obtained by optimizing the recommendation objective over $\mathcal{D}$, i.e., $\theta^* = \arg\min_\theta \mathcal{L}_\mathcal{D}(\theta)$, where $\mathcal{L}_\mathcal{D}$ denotes the pretraining recommendation loss (e.g., BPR loss), and θ^* corresponds to a local optimum over the full training set $\mathcal{D}$.

Given a forgetting set $\mathcal{D}_f \subset \mathcal{D}$, *recommendation unlearning* aims to obtain an unlearned model θ_u that removes the influence of $\mathcal{D}_f$ from the pretrained recommender while preserving utility on the remaining data $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$. To this end, recommendation unlearning typically involves two objectives: (1) an *unlearning objective* $\mathcal{L}_u(\mathcal{D}_f, \theta)$, which reverses the recommendation loss on the forgotten data and reduces the model's preference for $\mathcal{D}_f$ samples, and (2) a *retaining objective* $\mathcal{L}_r(\mathcal{D}_r, \theta)$, which preserves recommendation utility on $\mathcal{D}_r$. This can be formulated as:

$$\min_\theta \ \mathcal{L}_u(\mathcal{D}_f, \theta) + \lambda \mathcal{L}_r(\mathcal{D}_r, \theta), \tag{1}$$

where λ balances the unlearning and utility objectives. However, $\mathcal{L}_u(\mathcal{D}_f, \theta)$ and $\mathcal{L}_r(\mathcal{D}_r, \theta)$ are inherently conflicting, and their trade-off evolves during optimization. Thus, existing fixed-weight methods often struggle to balance unlearning effectiveness and model utility.

In this paper, we focus on *source-free recommendation unlearning*. The original training data $\mathcal{D}$ are unavailable during unlearning, and consequently $\mathcal{D}_r$ is also inaccessible. This setting is particularly challenging because the retaining objective $\mathcal{L}_r(\mathcal{D}_r, \theta)$ and its gradient $\nabla_\theta \mathcal{L}_r(\mathcal{D}_r, \theta)$ cannot be directly computed, making existing recommendation unlearning methods infeasible. Therefore, we propose an adaptive source-free recommendation unlearning method, i.e., **AdaSRU**. **AdaSRU** estimates the inaccessible gradient on $\mathcal{D}_r$ using $\mathcal{D}_f$ and the pretrained model θ^*, along with squared gradient accumulators from adaptive optimizers. It then constructs an adaptive gradient-constrained update, thereby enabling efficient recommendation unlearning without accessing $\mathcal{D}_r$.

3.2 Source-Free Remaining-Data Gradient Estimation

Although existing recommendation unlearning methods have achieved promising results, most rely on access to $\mathcal{D}_r$ to compute $\nabla\mathcal{L}_r$ and thus preserve recommendation utility during unlearning, which is infeasible in the source-free setting. To address this limitation, we propose Source-Free Remaining-Data Gradient Estimation (SRGE), which estimates the unavailable remaining-data gradient. Specifically, SRGE exploits the optimality condition of θ^* on $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f$ to establish the relationship between the remaining-data gradient $\nabla\mathcal{L}_r(\theta)$ and the forgetting-data gradient $\nabla\mathcal{L}_f(\theta)$. It then uses squared gradient accumulators from adaptive optimizers (e.g., Adam) as a lightweight curvature proxy, thereby avoiding explicit access to $\mathcal{D}_r$ and costly second-order computation.

Retaining–Forgetting Gradient Relation. To begin with, we establish the relationship between the inaccessible gradient $\nabla\mathcal{L}_r(\theta)$ on $\mathcal{D}_r$ and the accessible gradient $\nabla\mathcal{L}_f(\theta)$ on $\mathcal{D}_f$, where $\mathcal{L}_f(\mathcal{D}_f, \theta)$ is the original recommendation loss on $\mathcal{D}_f$, distinct from the unlearning objective $\mathcal{L}_u(\mathcal{D}_f, \theta)$. The latter aims to reduce the model's preference for $\mathcal{D}_f$, i.e., reversing the effect of $\mathcal{L}_f(\mathcal{D}_f, \theta)$. Let $n = |\mathcal{D}|$ and $m = |\mathcal{D}_f|$, so that $\mathcal{D}_r$ contains $n - m$ interactions. The objective over $\mathcal{D}$ is then:

$$\mathcal{L}_\mathcal{D}(\theta) = \frac{n-m}{n}\mathcal{L}_r(\theta) + \frac{m}{n}\mathcal{L}_f(\theta). \tag{2}$$

Since θ^* is optimized over $\mathcal{D}$, we assume that θ^* is stationary and follow the standard first-order optimality condition, i.e., $\nabla\mathcal{L}_\mathcal{D}(\theta^*) = 0$. Therefore, we derive the following proposition.

Proposition 1. (Gradient Relation at the Original Optimum) *Given the full-data objective $\mathcal{L}_\mathcal{D}(\theta)$, the retaining objective $\mathcal{L}_r(\theta)$, and the forget-set training objective $\mathcal{L}_f(\theta)$, suppose that the original model θ^* satisfies the first-order optimality condition for the full-data objective, i.e., $\nabla\mathcal{L}_\mathcal{D}(\theta^*) = 0$. Then the remaining-data gradient at θ^* is exactly related to the forget-set training gradient as:*

$$\nabla\mathcal{L}_r(\theta^*) = -\rho\nabla\mathcal{L}_f(\theta^*), \qquad \rho = \frac{m}{n-m}. \tag{3}$$

The proof of Proposition 1 is provided in Appendix D.1. We can observe that this proposition establishes a source-free gradient identity: the inaccessible $\nabla\mathcal{L}_r(\theta^*)$ can be exactly inferred from $\nabla\mathcal{L}_f(\theta^*)$ via the scale factor ρ, without accessing $\mathcal{D}_r$. Proposition 1 characterizes $\nabla\mathcal{L}_r$ only at θ^*, whereas unlearning requires an estimate at the current model θ_t. To transfer this relation, we therefore introduce a curvature-aware approximation. Specifically, we expand the remaining-data gradient around θ^* via a first-order Taylor approximation: $\nabla\mathcal{L}_r(\theta_t) \approx \nabla\mathcal{L}_r(\theta^*) + H_r(\theta^*)(\theta_t - \theta^*)$. This yields a curvature-aware estimate of the remaining-data gradient at θ_t.

Proposition 2. (Curvature-Aware Remaining-Data Gradient Approximation) *Given $\mathcal{L}_r(\theta)$, $\mathcal{L}_f(\theta)$, and $\mathcal{L}_\mathcal{D}(\theta)$, let θ_t be the current model during unlearning. Define the parameter displacement as $\Delta_t = \theta_t - \theta^*$. Under the first-order Taylor approximations of both $\nabla\mathcal{L}_r(\theta_t)$ and $\nabla\mathcal{L}_f(\theta_t)$ around θ^*, the remaining-data gradient at θ_t can be approximated by the proxy:*

$$\nabla\mathcal{L}_r(\theta_t) \approx \widehat{g}_r(\theta_t) = -\rho\nabla\mathcal{L}_f(\theta_t) + \kappa H_\mathcal{D}(\theta^*)\Delta_t, \qquad \rho = \frac{m}{n-m}, \quad \kappa = \frac{n}{n-m}, \tag{4}$$

where $H_\mathcal{D}(\theta^*) = \nabla^2\mathcal{L}_\mathcal{D}(\theta^*)$ denotes the full-data Hessian at θ^*.

The proof of Proposition 2 is provided in Appendix D.2. We can observe that Proposition 2 extends the source-free gradient identity from θ^* to the current model θ_t. Specifically, by applying a first-order Taylor expansion to both $\nabla\mathcal{L}_r(\theta_t)$ and $\nabla\mathcal{L}_f(\theta_t)$ around θ^*, we have:

$$\nabla\mathcal{L}_r(\theta_t) \approx \nabla\mathcal{L}_r(\theta^*) + H_r(\theta^*)\Delta_t, \quad \nabla\mathcal{L}_f(\theta_t) \approx \nabla\mathcal{L}_f(\theta^*) + H_f(\theta^*)\Delta_t. \tag{5}$$

After that, we obtain $\nabla\mathcal{L}_r(\theta_t) \approx -\rho\nabla\mathcal{L}_f(\theta_t) + (H_r(\theta^*) + \rho H_f(\theta^*))\,\Delta_t$. Since $\mathcal{D}_r$ is unavailable, $H_r(\theta^*)$ cannot be directly computed. From the Hessian decomposition, we use the full-data Hessian $H_{\mathcal{D}}(\theta^*)$ to express the required curvature term: $H_{\mathcal{D}}(\theta^*) = \frac{n-m}{n}H_r(\theta^*) + \frac{m}{n}H_f(\theta^*)$. We then obtain $\kappa H_{\mathcal{D}}(\theta^*) = H_r(\theta^*) + \rho H_f(\theta^*)$. Substituting this into the local approximation yields the retaining-gradient proxy $\widehat{g}_r(\theta_t)$. Here, $-\rho\nabla\mathcal{L}_f(\theta_t)$ estimates the remaining-data gradient from the accessible forgetting-data gradient, while $\kappa H_{\mathcal{D}}(\theta^*)\Delta_t$ corrects the shift from θ^* to θ_t.

Proposition 3. (Bounded Proxy-Retain Gradient Error with Full-Data Hessian) *Let θ^* denote the pretrained model optimized over the full dataset $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f$, and let $\widehat{g}_r(\theta_t) = -\rho\nabla\mathcal{L}_f(\theta_t) + \kappa H_{\mathcal{D}}(\theta^*)(\theta_t - \theta^*)$ be the proxy retaining gradient constructed using the full-data Hessian $H_{\mathcal{D}}(\theta^*)$, where $\rho = m/(n-m)$ and $\kappa = n/(n-m)$. Under the local smoothness conditions of Assumption 1, for any unlearning iterate θ_t in a neighborhood of θ^*, the proxy error satisfies*

$$\|\widehat{g}_r(\theta_t) - \nabla\mathcal{L}_r(\theta_t)\| \le \delta_t^H \le (\rho G + \epsilon_H)\|\theta_t - \theta^*\|, \tag{6}$$

where δ_t^H bounds the Hessian-based proxy error, G is the local smoothness constant, and ϵ_H bounds the discrepancy between the scaled full-data and remaining-data Hessians.

The proof of Proposition 3 and the derivation of ϵ_H are provided in Appendix D.3. The analysis bounds the Hessian-based proxy error in terms of parameter displacement and local curvature.

Adam-Based Remaining-Data Gradient Approximation. Building on the remaining-data gradient proxy introduced in Proposition 2, we note that explicitly computing $H_{\mathcal{D}}(\theta^*)\Delta_t$ is infeasible in the source-free setting due to the unavailability of $\mathcal{D}$. Therefore, we further approximate $H_{\mathcal{D}}(\theta^*)\Delta_t$ using Adam's squared-gradient accumulators through the following approximation chain:

$$H_{\mathcal{D}}(\theta^*)\Delta_t \approx \mathrm{diag}(H_{\mathcal{D}}(\theta^*)) \odot \Delta_t \approx \mathrm{diag}(F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*)) \odot \Delta_t \approx v^* \odot \Delta_t, \tag{7}$$

where $F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*)$ denotes the empirical Fisher matrix computed from $\mathcal{L}_{\mathcal{D}}$ around θ^*, and v^* denotes Adam's squared-gradient accumulators at the end of pretraining, which encode collaborative signals and serve as a lightweight diagonal curvature proxy rather than an exact Hessian equivalent. The full Hessian $H_{\mathcal{D}}(\theta^*)$ captures both per-parameter curvature and cross-parameter interactions. However, storing and multiplying $H_{\mathcal{D}}(\theta^*)$ is expensive for deep recommender models, so we adopt a standard diagonal curvature approximation $\mathrm{diag}(H_{\mathcal{D}}(\theta^*))$ that preserves the dominant sensitivity of each parameter. Next, we estimate $\mathrm{diag}(H_{\mathcal{D}}(\theta^*))$ without accessing $\mathcal{D}$. Specifically, under the negative log-likelihood perspective, the Hessian can be approximated by the Fisher Information Matrix (FIM), and common recommendation losses such as BPR can be regarded as pairwise likelihood objectives. The FIM captures the local curvature of the likelihood function.

Proposition 4. (Empirical Fisher-Based Approximation of the Hessian) *Under the likelihood view, the empirical Fisher matrix $F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*)$ is computed from the gradients of observed training instances:*

$$F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*) = \frac{1}{n}\sum_{a=1}^{n} g_a(\theta^*)g_a(\theta^*)^\top, \text{ with } g_a(\theta^*) = \nabla_\theta \ell(z_a;\theta^*). \tag{8}$$

Its diagonal is used to approximate the diagonal Hessian of the full-data objective:

$$\mathrm{diag}(H_{\mathcal{D}}(\theta^*)) \approx \mathrm{diag}(F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*))\,. \tag{9}$$

Proposition 4 connects $\mathrm{diag}(H_{\mathcal{D}}(\theta^*))$ to $\mathrm{diag}(F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*))$. The proof is provided in Appendix D.4. However, as the training dataset $\mathcal{D}$ is inaccessible by definition, computing $\mathrm{diag}(F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*))$ directly is infeasible. We therefore further approximate $\mathrm{diag}(F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*))$ using the squared gradient accumulators stored by Adam during pretraining.

Proposition 5. (Adam-Based Curvature Proxy) *Let v^* be the squared gradient accumulator stored by Adam during pretraining, and let the joint empirical Fisher diagonal be*

$$\mathrm{diag}\left(F_{\mathcal{D}}^{\mathrm{joint,emp}}(\theta^*)\right) = \mathbb{E}_{\mathcal{B}}\left[\left(\sum_{a\in\mathcal{B}} g_a(\theta^*)\right)^2\right]. \tag{10}$$

Then, the diagonal of the full-data Hessian $H_{\mathcal{D}}(\theta^*)$ can be approximated via the mapping:

$$\mathrm{diag}(H_{\mathcal{D}}(\theta^*)) \xrightarrow{\textit{FIM}} \mathrm{diag}(F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*)) \xrightarrow{\textit{Joint}} \mathrm{diag}(F_{\mathcal{D}}^{\mathrm{joint,emp}}(\theta^*)) \xrightarrow{\textit{Adam}} v^*. \tag{11}$$

In Eq. (33)*, using a diagonal curvature approximation, the curvature term in the proxy gradient is*

$$H_{\mathcal{D}}(\theta^*)\Delta_t \approx v^* \odot \Delta_t, \quad \Delta_t = \theta_t - \theta^*. \tag{12}$$

The proof of Proposition 5 is provided in Appendix D.5. We then substitute $H_{\mathcal{D}}(\theta^*)\Delta_t \approx v^* \odot \Delta_t$ into the curvature-aware remaining-data gradient proxy in Eq. (33), yielding the practical estimator:

$$\tilde{g}_r(\theta_t) = -\rho \nabla \mathcal{L}_f(\theta_t) + \kappa \left(v^* \odot (\theta_t - \theta^*)\right), \tag{13}$$

where $\rho = m/(n-m)$, $\kappa = n/(n-m)$, and v^* is the squared gradient accumulators stored by Adam. This estimator only requires the accessible forgetting-data gradient $\nabla \mathcal{L}_f(\theta_t)$, the pretrained model θ^*, and the optimizer statistics v^*. Thus, SRGE obtains $\tilde{g}_r(\theta_t)$ as an estimate of $\nabla \mathcal{L}_r(\theta_t)$, without accessing $\mathcal{D}_r$ or $\mathcal{D}$, and without explicitly computing $H_{\mathcal{D}}(\theta^*)\Delta_t$. Meanwhile, in a neighborhood of θ^*, the estimation error satisfies $\|\tilde{g}_r(\theta_t) - \nabla \mathcal{L}_r(\theta_t)\| \leq \delta_t^H + \delta_t^A$, where δ_t^H is the Hessian-based proxy error bound and δ_t^A accounts for the Adam approximation error.

3.3 Adaptive Gradient-Constrained Unlearning Update

After estimating $\nabla \mathcal{L}_r(\theta_t)$ with $\tilde{g}_r(\theta_t)$, we need to determine how to update the model for effective unlearning while preserving recommendation utility. Traditional methods typically combine the unlearning objective $\mathcal{L}_u(\mathcal{D}_f, \theta)$ and the retaining objective $\mathcal{L}_r(\mathcal{D}_r, \theta)$ linearly with a fixed weight λ, but such a fixed trade-off cannot adapt to varying gradient conflicts during optimization. To address this, we propose an adaptive gradient-constrained unlearning (AGCU) update, which seeks an update direction d_t that sufficiently decreases the unlearning objective while limiting the degradation of the retaining objective. This allows the trade-off between the unlearning and retaining objectives to be adaptively adjusted according to the current gradient geometry.

To implement this unlearning objective concretely, we define $\mathcal{L}_u(\mathcal{D}_f, \theta)$ for recommendation tasks as follows. In these tasks, $\mathcal{L}_f$ denotes the original training loss on $\mathcal{D}_f$ (e.g., BPR loss). For a forgotten interaction $(u, i) \in \mathcal{D}_f$, the original recommender typically ranks the interacted item i above non-interacted items $j \in \mathcal{I}_u^-$. To reverse this effect, we define

$$\mathcal{L}_u(\mathcal{D}_f, \theta) = -\sum_{(u,i)\in\mathcal{D}_f} \sum_{j\in\mathcal{I}_u^-} \log \sigma\left(\hat{y}_{uj} - \hat{y}_{ui}\right), \tag{14}$$

where $\hat{y}_{ui} = f_\theta(u, i)$ is the predicted preference score, and $\mathcal{I}_u^-$ denotes the non-interacted items of user u. Minimizing $\mathcal{L}_u$ effectively suppresses the model's preference for forgotten interactions by reversing their original pairwise ranking relations. Then, at the t-th unlearning step, we formulate d_t as a descent direction and update the model along this direction: $\theta_{t+1} = \theta_t - \alpha_t d_t$, where $\alpha_t > 0$ denotes the step size controlling the magnitude of the parameter update. Since α_t is typically small, we can use a first-order Taylor approximation to estimate the local change of the objectives:

$$\begin{aligned} \mathcal{L}_u(\theta_{t+1}) &\approx \mathcal{L}_u(\theta_t) - \alpha_t g_u(\theta_t)^\top d_t, \\ \mathcal{L}_r(\theta_{t+1}) &\approx \mathcal{L}_r(\theta_t) - \alpha_t \tilde{g}_r(\theta_t)^\top d_t, \end{aligned} \tag{15}$$

where $g_u(\theta_t) = \nabla_\theta \mathcal{L}_u(\theta_t)$ denotes the gradient of the unlearning objective, and $\tilde{g}_r(\theta_t) \approx \nabla_\theta \mathcal{L}_r(\theta_t)$ is an approximation of the retaining objective gradient. Intuitively, the inner products $g_u^\top d_t$ and $\tilde{g}_r^\top d_t$ represent the first-order rate of change in the respective losses along d_t, respectively. To achieve utility-preserving unlearning, we seek a direction d_t that maximizes the decrease of the unlearning objective while constraining the local degradation of the retaining objective:

$$\max_{d_t} \quad g_u(\theta_t)^\top d_t - \frac{1}{2}\|d_t\|^2 \quad \text{s.t.} \quad \tilde{g}_r(\theta_t)^\top d_t \geq -\epsilon_t, \tag{16}$$

where the quadratic regularization term controls the magnitude of the update direction and prevents an unbounded solution, and the constraint bounds the allowed first-order increase of the retaining loss by ϵ_t. Since the local update problem in Eq. (16) is a concave quadratic maximization problem with a linear constraint, it admits a unique closed-form solution. Specifically, let

$$\phi(d_t) = g_u(\theta_t)^\top d_t - \frac{1}{2}\|d_t\|^2, \qquad c(d_t) = -\tilde{g}_r(\theta_t)^\top d_t - \epsilon_t, \tag{17}$$

then the local update problem can be equivalently formulated as $\max_{d_t} \phi(d_t)$ s.t. $c(d_t) \leq 0$, where $\phi(d_t)$ is strictly concave and $c(d_t)$ is affine. By the Karush-Kuhn-Tucker (KKT) conditions, this problem admits a unique global optimum. We summarize the closed-form solution in Proposition 6.

Proposition 6. (Closed-form Adaptive Unlearning Direction) *Let $g_u(\theta_t) = \nabla_\theta \mathcal{L}_u(\mathcal{D}_f, \theta_t)$, and let $\tilde{g}_r(\theta_t)$ be the SRGE estimate of $\nabla \mathcal{L}_r(\theta_t)$. For the constrained direction search problem in Eq.* (16)*, if $\|\tilde{g}_r(\theta_t)\| > 0$ (to ensure the denominator in Eq.* (18) *is well-defined), the optimal direction is $d_t^* = g_u(\theta_t) + \lambda_t^* \tilde{g}_r(\theta_t)$, where*

$$\lambda_t^* = \left[\frac{-\tilde{g}_r(\theta_t)^\top g_u(\theta_t) - \epsilon_t}{\|\tilde{g}_r(\theta_t)\|^2} \right]_+, \qquad [x]_+ = \max(x, 0). \tag{18}$$

The proof of Proposition 6 is provided in Appendix D.6. Intuitively, the optimal update direction d_t^* consists of two components: the unlearning gradient $g_u(\theta_t)$, which aims to degrade the model's memory on the forgotten interactions, and a correction term along the retaining gradient $\tilde{g}_r(\theta_t)$, which is activated only when the retaining constraint is active. When the constraint is inactive ($\lambda_t^* = 0$), the update follows the unlearning gradient directly. Otherwise ($\lambda_t^* > 0$), the correction term adjusts the direction to ensure that the degradation of the retaining objective does not exceed the allowed threshold ϵ_t. Hence, unlike a fixed-weight combination of objectives, AGCU adaptively adjusts the retaining strength according to the current gradient geometry. To make the tolerance adaptive to the scale and geometry of the current gradients, we define $\epsilon_t = \epsilon \, \|g_u(\theta_t)\| \, \|\tilde{g}_r(\theta_t)\|, \ \epsilon \geq 0$, where ϵ controls the overall tolerance. This ensures that the allowed first-order degradation is scale-aware, while the activation and magnitude of λ_t^* further depend on the relative orientation between $g_u(\theta_t)$ and $\tilde{g}_r(\theta_t)$. Finally, the parameter update at step t follows the closed-form adaptive direction:

$$\theta_{t+1} = \theta_t - \alpha_t d_t^*, \qquad d_t^* = g_u(\theta_t) + \lambda_t^* \tilde{g}_r(\theta_t), \quad \alpha_t > 0, \tag{19}$$

where λ_t^* is the optimal Lagrange multiplier computed in Proposition 6. This update enforces the estimated retaining-gradient constraint through the adaptive direction $d_t^*(\theta_t)$, with all quantities explicitly dependent on the current time step t. Meanwhile, we further provide a theoretical guarantee for AGCU. Building on the bounded SRGE estimation error $\delta_t = \delta_t^H + \delta_t^A$, covering both Hessian-based proxy and Adam approximation errors, the following proposition establishes AGCU's convergence to an approximate Pareto stationary point of the unlearning and retaining objectives.

Proposition 7. (Approximate Pareto Stationarity of AGCU) *Assume that $\mathcal{L}_u$ and $\mathcal{L}_r$ are differentiable, and that $\mathcal{L}_u$ is G-smooth and bounded below. At each unlearning step, AGCU computes d_t by solving Eq.* (16) *and updates the model via $\theta_{t+1} = \theta_t - \alpha d_t$, with a constant step size $\alpha \leq 1/(2G)$. Let the SRGE remaining-data gradient estimate $\tilde{g}_r(\theta_t)$ satisfy $\|\tilde{g}_r(\theta_t) - \nabla \mathcal{L}_r(\theta_t)\| \leq \delta_t$, where $\delta_t = \delta_t^H + \delta_t^A$ accounts for the Hessian-based proxy and Adam approximation errors. Then, after t unlearning steps, AGCU satisfies the following approximate Pareto stationarity bound:*

$$\min_{i=1,\ldots,t} \min_{(\mu_u,\mu_r)\in\Delta_2} \|\mu_u \nabla \mathcal{L}_u(\theta_i) + \mu_r \nabla \mathcal{L}_r(\theta_i)\| \leq O(t^{-1/2}) + O\left(\frac{1}{t}\sum_{i=1}^{t} \delta_i\right), \tag{20}$$

where $\Delta_2 = \{(\mu_u, \mu_r) \mid \mu_u, \mu_r \geq 0, \ \mu_u + \mu_r = 1\}$.

The proof of Proposition 7 is provided in Appendix D.7. Proposition 7 establishes an approximate Pareto stationarity bound for AGCU, with the residual depending on the average SRGE estimation error. All quantities in this update are computed from the forgetting data $\mathcal{D}_f$, the pretrained model θ^*, and the stored Adam statistics v^*, without accessing $\mathcal{D}_r$ or $\mathcal{D}$. After T iterations, **AdaSRU** produces the final unlearned model θ_T. Through its adaptive and source-free parameter updates, it effectively removes the influence of $\mathcal{D}_f$ while preserving overall recommendation utility.

4 Experiments

4.1 Experimental setup

Datasets. We perform experiments on three commonly used real-world datasets: *MovieLens-1M*[1], *MovieLens-10M*[2], and *Yelp2018*[3]. The two MovieLens datasets reflect different levels of interaction sparsity. For consistency, we binarize the original ratings to indicate whether a user interacted with an item. Detailed statistics of all datasets are provided in Appendix E.1.

[1] `https://grouplens.org/datasets/movielens/1m/`

[2] `https://grouplens.org/datasets/movielens/10m/`

[3] `https://business.yelp.com/data/resources/open-dataset/`

Table 1: Experimental results of different methods for recommendation unlearning.

ML-1M	**BPR**				**WMF**				**LightGCN**			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
Retrain	0.1517	0.2401	0.3429	0.3384	0.1607	0.2519	0.3563	0.3526	0.1551	0.2440	0.3526	0.3464
SISA	0.1072	0.1724	0.2753	0.2659	0.0753	0.1257	0.2043	0.1983	0.1081	0.1730	0.2746	0.2666
RecEraser	0.1354	0.2191	0.3118	0.3096	0.1395	0.2191	0.3224	0.3162	0.1385	0.2202	0.3184	0.3138
CEU	0.1393	0.2255	0.3208	0.3188	0.1435	0.2255	0.3317	0.3255	0.1425	0.2267	0.3276	0.3231
RRL	0.1426	0.2257	0.3223	0.3181	0.1504	0.2368	0.3349	0.3314	0.1458	0.2293	0.3314	0.3256
UnlearnRec	0.1445	0.2342	0.3314	0.3291	0.1488	0.2342	0.3427	0.3360	0.1478	0.2354	0.3384	0.3336
CSFU	0.1214	0.1920	0.2743	0.2707	0.1280	0.2015	0.2851	0.2821	0.1241	0.1952	0.2821	0.2771
ETR	0.1366	0.2161	0.3086	0.3046	0.1440	0.2267	0.3207	0.3173	0.1396	0.2196	0.3173	0.3117
AdaSRU	**0.1472**	**0.2369**	**0.3326**	**0.3309**	**0.1537**	**0.2447**	**0.3456**	**0.3429**	**0.1505**	**0.2367**	**0.3421**	**0.3368**
ML-10M	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
Retrain	0.1964	0.3037	0.3448	0.3558	0.2108	0.3199	0.3729	0.3812	0.2008	0.3080	0.3579	0.3660
SISA	0.1608	0.2488	0.2982	0.3031	0.1322	0.2126	0.2480	0.2567	0.1212	0.2018	0.2272	0.2458
RecEraser	0.1747	0.2729	0.3132	0.3226	0.1938	0.2946	0.3475	0.3542	0.1828	0.2818	0.3274	0.3258
CEU	0.1798	0.2810	0.3223	0.3321	0.1994	0.3033	0.3575	0.3644	0.1881	0.2901	0.3369	0.3354
RRL	0.1846	0.2854	0.3241	0.3345	0.1982	0.3007	0.3506	0.3581	0.1887	0.2895	0.3365	0.3446
UnlearnRec	0.1865	0.2917	0.3329	0.3429	0.2068	0.3149	0.3693	0.3762	0.1951	0.3012	0.3483	0.3463
CSFU	0.1630	0.2524	0.2862	0.2953	0.1753	0.2655	0.3095	0.3162	0.1666	0.2556	0.2971	0.3038
ETR	0.1768	0.2733	0.3104	0.3202	0.1897	0.2879	0.3356	0.3429	0.1807	0.2772	0.3221	0.3294
AdaSRU	**0.1886**	**0.2924**	**0.3337**	**0.3442**	**0.2084**	**0.3171**	**0.3699**	**0.3783**	**0.1973**	**0.3035**	**0.3495**	**0.3513**
Yelp2018	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
Retrain	0.0317	0.0534	0.0353	0.0434	0.0377	0.0635	0.0433	0.0525	0.0376	0.0653	0.0435	0.0537
SISA	0.0152	0.0271	0.0180	0.0223	0.0234	0.0399	0.0276	0.0334	0.0220	0.0390	0.0258	0.0319
RecEraser	0.0257	0.0454	0.0292	0.0365	0.0346	0.0591	0.0398	0.0487	0.0339	0.0587	0.0390	0.0481
CEU	0.0262	0.0462	0.0297	0.0371	0.0352	0.0602	0.0405	0.0495	0.0345	0.0597	0.0396	0.0489
RRL	0.0269	0.0465	0.0307	0.0377	0.0328	0.0552	0.0376	0.0457	0.0327	0.0568	0.0378	0.0467
UnlearnRec	0.0272	0.0486	0.0308	0.0386	0.0366	0.0625	0.0421	0.0515	0.0358	0.0621	0.0412	0.0508
CSFU	0.0232	0.0401	0.0264	0.0325	0.0283	0.0476	0.0324	0.0394	0.0282	0.0490	0.0326	0.0403
ETR	0.0263	0.0449	0.0296	0.0364	0.0317	0.0533	0.0363	0.0441	0.0316	0.0549	0.0365	0.0451
AdaSRU	**0.0284**	**0.0496**	**0.0325**	**0.0392**	**0.0371**	**0.0631**	**0.0428**	**0.0519**	**0.0369**	**0.0636**	**0.0424**	**0.0512**

Compared Methods. We evaluate the effectiveness of **AdaSRU** across different recommendation models and unlearning baselines. (1) *Recommendation Models.* We use three widely adopted recommendation models: **BPR** [44], **WMF** [7, 19], and **LightGCN** [18]. (2) *Unlearning Baselines.* We compare our proposed **AdaSRU** against a diverse set of unlearning baselines, including **Retrain**, **SISA** [3], **RecEraser** [8], **CEU** [51], **RRL** [58], **UnlearnRec** [9], **CSFU** [2] and **ETR** [57]. The details of the recommendation models and unlearning baselines are provided in Appendix E.2.

Evaluation Metrics. We evaluate recommendation performance using *Recall@*K and *NDCG@*K for $K \in \{10, 20\}$. Meanwhile, unlearning completeness is evaluated via a Membership Inference Oracle (MIO), reporting *ACC* and *AUC*, with implementation details of MIO provided in Appendix F. Furthermore, we employ early stopping, terminating training if *Recall@10* on the validation set fails to improve for 10 consecutive epochs to stabilize training.

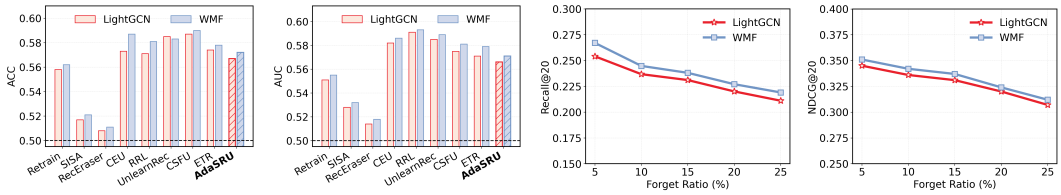
Implementation Details. We split the user–item interactions into 80% for training and 20% for testing, holding out 10% of the training set for validation. User and item embeddings are learned from WMF, with an embedding dimension of $d = 64$, and LightGCN uses $L = 3$ propagation layers. In pretraining, we employ Adam as the optimizer with L2 regularization $\lambda_{\text{reg}} = 1\text{e-}4$. For unlearning, we use a forgetting ratio of $\rho_{\text{forget}} = 10\%$, a maximum training epoch $T = 1000$, step size $\alpha = 1\text{e-}3$, tolerance $\epsilon = 1\text{e-}3$, and maximum Lagrange multiplier $\lambda_{\max} = 1 \times 10^5$ to ensure numerical stability. All experiments are repeated five times with different random seeds, and the averaged results are reported. Details of the environment configurations are provided in Appendix E.3.

4.2 Performance Comparison

Recommendation Performance Comparison. Table 1 reports recommendation performance on three datasets, where R@K and N@K denote Recall@K and NDCG@K, respectively. We have the following observations. (1) **AdaSRU** achieves the best overall performance among non-retraining methods, with stable but moderate gains over the strongest baseline, UnlearnRec, in both Recall and NDCG. This suggests that source-free remaining-data gradient estimation and adaptive constrained updates better preserve recommendation utility during unlearning without accessing the remaining data. (2) Retrain generally obtains the highest Recall and NDCG by retraining on the remaining data, but its full retraining cost makes it impractical for efficient unlearning. Other methods avoid

Table 2: Runtime comparison of different unlearning methods based on the LightGCN.

Dataset	Retrain	SISA	RecEraser	CEU	RRL	UnlearnRec	AdaSRU
ML-1M	649.47m	24.98m	32.64m	7.92m	8.74m	1.59m	1.27m
ML-10M	29148.21m	874.11m	1144.60m	772.44m	651.36m	5.96m	5.12m
Yelp2018	1558.73m	60.71m	88.04m	64.37m	54.28m	3.64m	2.85m



(a) MIO ACC (b) MIO AUC (c) Recall@20 (d) NDCG@20

Figure 2: Unlearning completeness and recommendation utility on ML-1M. In MIO results, the dashed horizontal line denotes the random-guessing level of 0.5, where values closer to 0.5 indicate more complete unlearning.

full retraining but generally suffer from performance degradation. (3) The advantage of **AdaSRU** is more evident on denser datasets such as ML-1M and ML-10M, while its gains become smaller on the sparser Yelp2018 dataset. This indicates that sparse interactions provide limited collaborative signals, making recommendation structures more sensitive to unlearning operations.

Unlearning Efficiency Comparison. We use runtime to evaluate unlearning efficiency. Table 2 compares the runtime of different methods under LightGCN. We observe that Retrain incurs the highest cost, as it rebuilds the model from scratch on the remaining data. In contrast, **AdaSRU** performs direct source-free parameter updates without retraining on the remaining set, substantially reducing computational overhead. In comparison, partition-based methods, such as SISA and RecEraser, reduce the cost of full retraining but still require retraining affected shards and aggregating submodels, which introduces non-negligible overhead. **AdaSRU** also shows clear efficiency advantages over CEU and RRL, since these optimization-based methods often rely on Hessian-vector products or second-order gradient computation to estimate the effect of removing interactions. By avoiding explicit second-order computation and using saved Adam second-moment statistics to approximate curvature-aware remaining-data gradients, **AdaSRU** achieves efficient source-free unlearning.

Membership Inference Evaluation. Figures 2a and 2b report the membership inference results on ML-1M. From the results, we have the following observations. (1) SISA and RecEraser provide stronger membership privacy, with RecEraser achieving the lowest ACC and AUC in most cases. This suggests that partition-based reconstruction can effectively suppress membership signals, but it requires data partitioning and partial retraining, which may introduce structural disruption and additional computational cost. (2) Among non-retraining methods, **AdaSRU** achieves competitive membership privacy. It generally obtains lower ACC and AUC than CEU, UnlearnRec, CSFU, and ETR, and remains close to the best non-retraining baseline across both backbones. This demonstrates its ability to suppress membership information without accessing the remaining training data. Although **AdaSRU** does not achieve the lowest membership inference scores compared with RecEraser, it offers a practical trade-off under the source-free setting by improving unlearning completeness while avoiding expensive retraining or data partitioning.

4.3 Sensitivity and Ablation Analysis

Effect of Forgetting Ratio. We study the sensitivity to different forgetting ratios on ML-1M by varying the ratio in $\{5\%, 10\%, 15\%, 20\%, 25\%\}$ and reporting Recall@20 and NDCG@20 in Fig. 2c and Fig. 2d. Both LightGCN and WMF show a consistent performance decline as the forgetting ratio increases, since removing more user–item interactions weakens collaborative signals and makes utility preservation more challenging. The degradation becomes more evident at 25%, indicating that a larger amount of interaction removal harms collaborative representation learning.

Ablation Study. We conduct an ablation study to evaluate the contribution of SRGE, curvature-aware correction, and AGCU in **AdaSRU**, with results provided in Appendix G.1. Specifically, *w/o SRGE* removes source-free remaining-data gradient estimation, *w/o Curvature* removes the Adam-statistics-based correction, and *w/o AGCU* replaces the adaptive constrained update with a fixed weighted update. The full **AdaSRU** achieves the best overall performance, confirming that these components are jointly effective. The performance drops of the three variants show that estimating the remaining-data gradient, incorporating curvature information, and adaptively constraining the update direction are all important for preserving utility during source-free recommendation unlearning.

5 Conclusion

In this paper, we propose an adaptive source-free recommendation unlearning (**AdaSRU**) method. **AdaSRU** first estimates the inaccessible remaining-data gradient by exploiting the first-order optimality condition of the pretrained recommender and constructing a source-free retaining-gradient estimator from the forgetting data, the pretrained model, and squared gradient accumulators from adaptive optimizers. Based on this estimator, **AdaSRU** performs a gradient-constrained update that maximizes forgetting while dynamically controlling utility degradation. We further provide theoretical analysis showing that **AdaSRU** approaches an approximate Pareto stationary solution under bounded gradient-estimation error. Extensive experiments on three real-world datasets demonstrate that **AdaSRU** achieves strong performance in both unlearning and recommendation.

Acknowledgments

The work was supported by National Natural Science Foundation of China (No. 62572486), Natural Science Foundation of Shandong Province (No. ZR2023MF007), European Union CoGNETs project under the Horizon Europe programme grant number 101135930 and in part by the UKRI through the UK Government's Horizon Europe funding Guarantee (grant number: 10119564).

References

[1] Sk Miraj Ahmed, Umit Yigit Basaran, Dripta S Raychaudhuri, Arindam Dutta, Rohit Kundu, Fahim Faisal Niloy, Basak Guler, and Amit K Roy-Chowdhury. Towards source-free machine unlearning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4948–4957, 2025.

[2] Umit Yigit Basaran, Sk Miraj Ahmed, Amit Roy-Chowdhury, and Basak Guler. A certified unlearning approach without access to source data. In *Forty-second International Conference on Machine Learning*, 2025.

[3] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE, 2021.

[4] Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 11186–11194, 2024.

[5] Chaochao Chen, Jiaming Zhang, Yizhao Zhang, Li Zhang, Lingjuan Lyu, Yuyuan Li, Biao Gong, and Chenggang Yan. Cure4rec: A benchmark for recommendation unlearning with deeper influence. *Advances in Neural Information Processing Systems*, 37:99128–99144, 2024.

[6] Chaochao Chen, Yizhao Zhang, Yuyuan Li, Jun Wang, Lianyong Qi, Xiaolong Xu, Xiaolin Zheng, and Jianwei Yin. Post-training attribute unlearning in recommender systems. *ACM Transactions on Information Systems*, 43(1):1–28, 2024.

[7] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems*, 38(2):1–28, 2020.

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

[9] Guoxuan Chen, Lianghao Xia, and Chao Huang. Pre-training for recommendation unlearning. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2112–2121, 2025.

[10] Dasol Choi, Soora Choi, Eunsun Lee, Jinwoo Seo, and Dongbin Na. Towards efficient machine unlearning with data augmentation: Guided loss-increasing (gli) to prevent the catastrophic model utility drop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 93–102, 2024.

[11] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7210–7217, 2023.

[12] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18:2345–2354, 2023.

[13] Yizhou Dang, Yuting Liu, Enneng Yang, Guibing Guo, Linying Jiang, Jianzhe Zhao, and Xingwei Wang. Efficient and adaptive recommendation unlearning: A guided filtering framework to erase outdated preferences. *ACM Transactions on Information Systems*, 43(2):1–25, 2025.

[14] Arnav Devalapally, Poornima Jain, Kartik Srinivas, and Vineeth N. Balasubramanian. ⊘ Source Models Leak What They Shouldn't ↛: Unlearning Zero-Shot Transfer in Domain Adaptation Through Adversarial Optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[15] Haocheng Dou, Tao Lian, and Xin Xin. Measuring interaction-level unlearning difficulty for collaborative filtering. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pages 279–288, 2025.

[16] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[17] Yongjing Hao, Fuzhen Zhuang, Deqing Wang, Guanfeng Liu, Victor S Sheng, and Pengpeng Zhao. A general strategy graph collaborative filtering for recommendation unlearning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 799–808, 2024.

[18] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 639–648, 2020.

[19] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the IEEE International Conference on Data Mining*, pages 263–272, 2008.

[20] Yuzheng Hu, Ruicheng Xian, Qilong Wu, Qiuling Fan, Lang Yin, and Han Zhao. Revisiting scalarization in multi-task learning: A theoretical perspective. *Advances in Neural Information Processing Systems*, 36:48510–48533, 2023.

[21] Zhiyu Hu, Yang Zhang, Minghao Xiao, Wenjie Wang, Fuli Feng, and Xiangnan He. Exact and efficient unlearning for large language model-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[22] Yu Hua, Weiming Liu, Gui Xu, Yaqing Hou, Yew-Soon Ong, and Qiang Zhang. Deterministic-to-stochastic diverse latent feature mapping for human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22724–22734. IEEE, 2025.

[23] Bei Hui, Lizong Zhang, Xue Zhou, Xiao Wen, and Yuhui Nian. Personalized recommendation system based on knowledge embedding and historical behavior. *Applied Intelligence*, 52(1): 954–966, 2022.

[24] Erin Illman and Paul Temple. California consumer privacy act. *The Business Lawyer*, 75(1): 1637–1646, 2019.

[25] Alexey Kravets and Vinay Namboodiri. Zero-shot class unlearning in clip with synthetic samples. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 6456–6464, 2025.

[26] Lin Li, Shengda Zhuo, Hongguang Lin, Jinchun He, Wangjie Qiu, Qinnan Zhang, Changdong Wang, and Shuqiang Huang. Enhancing partition distinction: A contrastive policy to recommendation unlearning. *Neural Networks*, 190:107667, 2025.

[27] YuXin Li, Felix Dangel, Derek Tam, and Colin Raffel. Fishers for free? approximating the fisher information matrix by recycling the squared gradient accumulator. *arXiv preprint arXiv:2507.18807*, 2025.

[28] Yuyuan Li, Chaochao Chen, Yizhao Zhang, Weiming Liu, Lingjuan Lyu, Xiaolin Zheng, Dan Meng, and Jun Wang. Ultrare: Enhancing receraser for recommendation unlearning via error decomposition. *Advances in Neural Information Processing Systems*, 36:12611–12625, 2023.

[29] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Yizhao Zhang, Biao Gong, Jun Wang, and Linxun Chen. Selective and collaborative influence function for efficient recommendation unlearning. *Expert Systems with Applications*, 234:121025, 2023.

[30] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Yizhao Zhang, Zhongxuan Han, Dan Meng, and Jun Wang. Making users indistinguishable: Attribute-wise unlearning in recommender systems. In *Proceedings of the 31st ACM international conference on multimedia*, pages 984–994, 2023.

[31] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Junlin Liu, and Jun Wang. Making recommender systems forget: Learning and unlearning for erasable recommendation. *Knowledge-Based Systems*, 283:111124, 2024.

[32] Yuyuan Li, Xiaohua Feng, Chaochao Chen, and Qiang Yang. A survey on recommendation unlearning: Fundamentals, taxonomy, evaluation, and open questions. *IEEE Transactions on Knowledge and Data Engineering*, 38(2):781–799, 2025.

[33] Wu Lin, Felix Dangel, Runa Eschenhagen, Juhan Bae, Richard E Turner, and Alireza Makhzani. Can we remove the square-root in adaptive gradient methods? a second-order perspective. *arXiv preprint arXiv:2402.03496*, 2024.

[34] Weiming Liu, Chaochao Chen, Xinting Liao, Mengling Hu, Jiajie Su, Yanchao Tan, and Fan Wang. User distribution mapping modelling with collaborative filtering for cross domain recommendation. In *Proceedings of the ACM web conference 2024*, pages 334–343, 2024.

[35] Weiming Liu, Chaochao Chen, Xinting Liao, Mengling Hu, Yanchao Tan, Fan Wang, Xiaolin Zheng, and Yew Soon Ong. Learning accurate and bidirectional transformation via dynamic embedding transportation for cross-domain recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8815–8823, 2024.

[36] Weiming Liu, Xiaolin Zheng, Chaochao Chen, Jiahe Xu, Xinting Liao, Fan Wang, Yanchao Tan, and Yew-Soon Ong. Reducing item discrepancy via differentially private robust embedding alignment for privacy-preserving cross domain recommendation. In *Forty-first International Conference on Machine Learning*, 2024.

[37] Weiming Liu, Jun Dan, Fan Wang, Xinting Liao, Junhao Dong, Hua Yu, Shunjie Dong, and Lianyong Qi. Distinguish then exploit: Source-free open set domain adaptation via weight barcode estimation and sparse label assignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4927–4938. IEEE, 2025.

[38] Weiming Liu, Xinting Liao, Jun Dan, Fan Wang, Hua Yu, Junhao Dong, Shunjie Dong, Lianyong Qi, and Yew Soon Ong. Solving discrete (semi) unbalanced optimal transport with equivalent transformation mechanism and kkt-multiplier regularization. *Advances in Neural Information Processing Systems*, 38:105330–105365, 2026.

[39] Zheqi Lv, Tianyu Zhan, Wenjie Wang, Xinyu Lin, Shengyu Zhang, Wenqiao Zhang, Jiwei Li, Kun Kuang, and Fei Wu. Collaboration of large language models and small recommendation models for device-cloud recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 962–973, 2025.

[40] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

[41] Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–46, 2025.

[42] Gaurav Patel and Qiang Qiu. Learning to unlearn while retaining: Combating gradient conflicts in machine unlearning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4211–4221, 2025.

[43] Data Protection. General data protection regulation. *Intersoft Consulting, Accessed in October*, 24(1), 2018.

[44] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 452–461, 2009.

[45] Alyssa Shuang Sha, Bernardo Nunes, and Armin Haller. Selective forgetting in machine learning and beyond: A survey. *ACM Computing Surveys*, 58(9):1–41, 2026.

[46] Alexander Soen and Ke Sun. Trade-offs of diagonal fisher information matrix estimators. *Advances in Neural Information Processing Systems*, 37:5870–5912, 2024.

[47] Haoran Sun, Fei Xiong, Yuanzhe Hu, and Liang Wang. Path-enhanced contrastive learning for recommendation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[48] Yi-Lin Sung, Varun Nair, and Colin A Raffel. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems*, 34:24193–24205, 2021.

[49] Xiuyuan Wang, Chaochao Chen, Weiming Liu, Xinting Liao, Fan Wang, and Xiaolin Zheng. Efficient source-free unlearning via energy-guided data synthesis and discrimination-aware multitask optimization. In *Forty-second International Conference on Machine Learning*, 2025.

[50] Jiahao Wu, Xialun Yun, Feng Ji, Li Peng, and Jielong Yang. Machine unlearning for alleviating negative transfer in partial-set source-free unsupervised domain adaptation. *IEEE Transactions on Artificial Intelligence*, 2025.

[51] Kun Wu, Jie Shen, Yue Ning, Ting Wang, and Wendy Hui Wang. Certified edge unlearning for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2606–2617, 2023.

[52] Xiaodong Wu, Wenyi Yu, Chao Zhang, and Philip Woodland. An improved empirical fisher approximation for natural gradient descent. *Advances in Neural Information Processing Systems*, 37:134151–134194, 2024.

[53] Yidong Wu, Siyuan Chen, Binrui Wu, Fan Li, and Jiechao Gao. Adaptive gradient masking for balancing id and mllm-based representations in recommendation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[54] Yang Xiao, Weiming Liu, Jun Dan, Tengyue Xu, Fan Wang, Hua Yu, Junhao Dong, Jiao Liu, Shunjie Dong, and Lianyong Qi. Bypassing the transport plan: Dynamic reweighting for out-of-distribution detection with optimal transport. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 32026–32036, 2026.

[55] Bo Yan, Yang Cao, Haoyu Wang, Wenchuan Yang, Junping Du, and Chuan Shi. Federated heterogeneous graph neural network for privacy-preserving recommendation. In *Proceedings of the ACM Web Conference 2024*, pages 3919–3929, 2024.

[56] Jielong Yang, Peiran Li, Xialun Yun, Xionghu Zhong, and Di Wu. Machine unlearning for source-free unsupervised partial-domain adaptation in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–12, 2025.

[57] Zhe-Rui Yang, Jindong Han, Chang-Dong Wang, and Hao Liu. Erase then rectify: A training-free parameter editing approach for cost-effective graph unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13044–13051, 2025.

[58] Xiaoyu You, Jianwei Xu, Mi Zhang, Zechen Gao, and Min Yang. Rrl: Recommendation reverse learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 9296–9304, 2024.

[59] Wei Yuan, Hongzhi Yin, Fangzhao Wu, Shijie Zhang, Tieke He, and Hao Wang. Federated unlearning for on-device recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pages 393–401, 2023.

[60] Qianru Zhang, Peng Yang, Junliang Yu, Haixin Wang, Xingwei He, Siu-Ming Yiu, and Hongzhi Yin. A survey on point-of-interest recommendation: Models, architectures, and security. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[61] Yang Zhang, Zhiyu Hu, Yimeng Bai, Jiancan Wu, Qifan Wang, and Fuli Feng. Recommendation unlearning via influence function. *ACM Transactions on Recommender Systems*, 3(2):1–23, 2024.

[62] Kairan Zhao, Meghdad Kurmanji, George-Octavian Barbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. *Advances in Neural Information Processing Systems*, 37:12293–12333, 2024.

[63] Xuyang Zhong, Haochen Luo, and Chen Liu. Dualoptim: Enhancing efficacy and stability in machine unlearning with dual optimizers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[64] Shiji Zhou, Tianbai Yu, Zhi Zhang, Heng Chang, Xiao Zhou, Dong Wu, and Han Zhao. Efficient utility-preserving machine unlearning with implicit gradient surgery. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

A Notation Table

We provide the important notations and their descriptions for clarification in Table 3.

B Algorithm Details

Algorithm 1 summarizes the procedure of the proposed **AdaSRU**. Algorithm 2 summarizes the SRGE procedure, which estimates the unavailable remaining-data gradient.

C Description for Assumptions

In this section, we make the following assumptions to provide theoretical guarantees for our unlearning analysis, where $\mathcal{L}_f(\mathcal{D}_f, \theta)$ denotes the original recommendation loss on $\mathcal{D}_f$ and $\mathcal{L}_u(\mathcal{D}_f, \theta)$ denotes the unlearning objective that reverses the effect of $\mathcal{L}_f(\mathcal{D}_f, \theta)$.

Assumption 1. (Smooth objectives) *The unlearning objective $\mathcal{L}_u(\theta)$, the retaining objective $\mathcal{L}_r(\theta)$, and the forgetting-data training objective $\mathcal{L}_f(\theta)$ are differentiable and G-smooth, i.e.,*

$$\|\nabla \mathcal{L}_j(\theta) - \nabla \mathcal{L}_j(\theta')\| \leq G\|\theta - \theta'\|, \quad j \in \{u, r, f\}. \tag{21}$$

Table 3: Summary of main notations.

Notation	Description
$\mathcal{U}, \mathcal{I}$	User and item sets, respectively
(u, i)	A user-item interaction pair
y_{ui}	Interaction indicator between user u and item i
$\hat{y}_{ui}$	Predicted preference score of user u for item i
$\mathcal{D}$	Full training set
$\mathcal{D}_f$	Forget set
$\mathcal{D}_r$	Retain set, remaining data after forgetting, $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$
n	Number of interactions in $\mathcal{D}$
m	Number of interactions in $\mathcal{D}_f$
$\mathcal{L}_{\mathcal{D}}(\theta)$	Objective over the full training set $\mathcal{D}$
$\mathcal{L}_f(\mathcal{D}_f, \theta)$	Original recommendation loss over the forget set $\mathcal{D}_f$
$\mathcal{L}_u(\mathcal{D}_f, \theta)$	Unlearning objective on $\mathcal{D}_f$
$\mathcal{L}_r(\mathcal{D}_r, \theta)$	Retain objective on $\mathcal{D}_r$
θ^*	Pretrained model parameters on $\mathcal{D}$
θ_t	Model parameters at unlearning step t
Δ_t	Parameter displacement, $\Delta_t = \theta_t - \theta^*$
ρ	Forget-retain scale factor, $\rho = m/(n-m)$
κ	Curvature scale factor, $\kappa = n/(n-m)$
$H_{\mathcal{D}}(\theta^*)$	Hessian of the full-data objective at θ^*
$\widehat{g}_r(\theta_t)$	Hessian-based proxy remaining-data gradient
$\tilde{g}_r(\theta_t)$	Adam-based SRGE remaining-data gradient estimate
v^*	squared gradient accumulators from adaptive optimizers (e.g., Adam) at θ^*
d_t	Unlearning update direction
λ_t	Lagrange multiplier for the retain-side constraint
ϵ_t	Allowed first-order degradation on the retain-side objective
α	Unlearning step size

Algorithm 1 Adaptive Source-Free Recommendation Unlearning (**AdaSRU**)

Require: Pretrained model θ^*, forgetting set $\mathcal{D}_f$, saved Adam/AdamW accumulator v^*, step size α, tolerance ϵ, number of steps T
Ensure: Unlearned model θ_T
Initialize $\theta_0 \leftarrow \theta^*$
for $t = 0, \ldots, T-1$ **do**
  Compute forgetting-data gradient $g_f(\theta_t) = \nabla \mathcal{L}_f(\theta_t)$
  Compute unlearning gradient $g_u(\theta_t) = \nabla \mathcal{L}_u(\theta_t)$
  Compute displacement $\Delta_t = \theta_t - \theta^*$
  Construct proxy remaining-data gradient: $\tilde{g}_r(\theta_t) = -\rho g_f(\theta_t) + \kappa\,(v^* \odot \Delta_t)$
  Compute adaptive Lagrange multiplier λ_t
  Clip multiplier: $\lambda_t \leftarrow \min(\lambda_t, \Lambda)$
  Compute adaptive update direction: $d_t = g_u(\theta_t) + \lambda_t \tilde{g}_r(\theta_t)$
  Update model: $\theta_{t+1} = \theta_t - \alpha d_t$
end for
return θ_T

Assumption 2. (Bounded unlearning objective) *The unlearning objective $\mathcal{L}_u(\theta)$ is bounded, i.e., there exists a constant $B > 0$ such that*

$$|\mathcal{L}_u(\theta)| \leq B \quad \textit{for all } \theta \in \mathbb{R}^d. \tag{22}$$

Assumption 3. (Bounded multiplier) *The Lagrange multiplier λ_t produced by the constrained update is bounded within a known range, i.e.,*

$$0 \leq \lambda_t \leq \Lambda \quad \forall t, \tag{23}$$

where $\Lambda > 0$ is a predetermined upper bound ensuring stability of the update.

Algorithm 2 Source-Free Remaining-Data Gradient Estimation (SRGE)

Require: Forgetting set $\mathcal{D}_f$, current model θ_t, pretrained model θ^*, saved Adam/AdamW accumulator v^*, dataset sizes $n = |\mathcal{D}|$, $m = |\mathcal{D}_f|$
Ensure: Proxy remaining-data gradient $\tilde{g}_r(\theta_t)$
1: Initialize $g_f(\theta_t) \leftarrow 0$
2: **for** each mini-batch $\mathcal{B}_f \subset \mathcal{D}_f$ **do**
3: Compute mini-batch forgetting-set gradient: $g_f^{\mathcal{B}}(\theta_t) = \nabla_\theta \mathcal{L}_f(\mathcal{B}_f, \theta_t)$
4: Accumulate gradient: $g_f(\theta_t) \leftarrow g_f(\theta_t) + g_f^{\mathcal{B}}(\theta_t)$
5: **end for**
6: Average forgetting-set gradient: $g_f(\theta_t) \leftarrow \frac{1}{|\mathcal{D}_f|} g_f(\theta_t)$
7: Compute displacement from pretrained model: $\Delta_t = \theta_t - \theta^*$
8: Set scaling factors: $\rho = \frac{m}{n-m}, \quad \kappa = \frac{n}{n-m}$
9: Estimate remaining-data gradient without accessing $\mathcal{D}_r$: $\tilde{g}_r(\theta_t) = -\rho g_f(\theta_t) + \kappa\,(v^* \odot \Delta_t)$
10: **return** $\tilde{g}_r(\theta_t)$

Assumption 1 is standard in non-convex first-order optimization. Assumption 2 bounds the accumulated ascent of the unlearning objective. Assumption 3 aligns with practical implementation, where the multiplier is clipped to avoid unstable updates.

D Proofs

In this section, we present the proof details.

D.1 Proof of Proposition 1

Proposition 1. (Gradient Relation at the Original Optimum) *Given the full-data objective $\mathcal{L}_\mathcal{D}(\theta)$, the retaining objective $\mathcal{L}_r(\theta)$, and the forget-set training objective $\mathcal{L}_f(\theta)$, suppose that the original model θ^* satisfies the first-order optimality condition for the full-data objective, i.e., $\nabla \mathcal{L}_\mathcal{D}(\theta^*) = 0$. Then the remaining-data gradient at θ^* is exactly related to the forget-set training gradient as:*

$$\nabla \mathcal{L}_r(\theta^*) = -\rho \nabla \mathcal{L}_f(\theta^*), \qquad \rho = \frac{m}{n-m}. \tag{24}$$

Proof. We decompose the full training set $\mathcal{D}$ into two disjoint subsets: the retaining set $\mathcal{D}_r$ and the forgetting set $\mathcal{D}_f$, where $n = |\mathcal{D}|$ and $m = |\mathcal{D}_f|$:

$$\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f, \qquad \mathcal{D}_r \cap \mathcal{D}_f = \emptyset. \tag{25}$$

The empirical objectives over the full set, retaining set, and forgetting set are defined as:

$$\begin{cases} \mathcal{L}_\mathcal{D}(\theta) = \dfrac{1}{n} \displaystyle\sum_{z \in \mathcal{D}} \ell(z;\theta), \\ \mathcal{L}_r(\theta) = \dfrac{1}{n-m} \displaystyle\sum_{z \in \mathcal{D}_r} \ell(z;\theta), \\ \mathcal{L}_f(\theta) = \dfrac{1}{m} \displaystyle\sum_{z \in \mathcal{D}_f} \ell(z;\theta). \end{cases} \tag{26}$$

Then the full-data objective can be decomposed as:

$$\begin{aligned} \mathcal{L}_\mathcal{D}(\theta) &= \frac{1}{n} \left(\sum_{z \in \mathcal{D}_r} \ell(z;\theta) + \sum_{z \in \mathcal{D}_f} \ell(z;\theta) \right) \\ &= \frac{n-m}{n} \mathcal{L}_r(\theta) + \frac{m}{n} \mathcal{L}_f(\theta). \end{aligned} \tag{27}$$

Taking the gradient with respect to θ, we obtain:

$$\nabla \mathcal{L}_\mathcal{D}(\theta) = \frac{n-m}{n} \nabla \mathcal{L}_r(\theta) + \frac{m}{n} \nabla \mathcal{L}_f(\theta). \tag{28}$$

Since θ^* satisfies the first-order optimality condition on $\mathcal{D}$, we have:

$$\nabla \mathcal{L}_{\mathcal{D}}(\theta^*) = 0. \tag{29}$$

Substituting $\theta = \theta^*$ into the gradient decomposition gives:

$$\frac{n-m}{n} \nabla \mathcal{L}_r(\theta^*) + \frac{m}{n} \nabla \mathcal{L}_f(\theta^*) = 0. \tag{30}$$

Therefore,

$$\nabla \mathcal{L}_r(\theta^*) = -\frac{m}{n-m} \nabla \mathcal{L}_f(\theta^*). \tag{31}$$

Letting $\rho = \frac{m}{n-m}$, we obtain:

$$\nabla \mathcal{L}_r(\theta^*) = -\rho \nabla \mathcal{L}_f(\theta^*). \tag{32}$$

Therefore, we conclude the proof of Proposition 1. □

D.2 Proof of Proposition 2

Proposition 2. (Curvature-Aware Remaining-Data Gradient Approximation) *Given $\mathcal{L}_r(\theta)$, $\mathcal{L}_f(\theta)$, and $\mathcal{L}_{\mathcal{D}}(\theta)$, let θ_t be the current model during unlearning. Define the parameter displacement as $\Delta_t = \theta_t - \theta^*$. Under the first-order Taylor approximations of both $\nabla \mathcal{L}_r(\theta_t)$ and $\nabla \mathcal{L}_f(\theta_t)$ around θ^*, the remaining-data gradient at θ_t can be approximated by the proxy:*

$$\nabla \mathcal{L}_r(\theta_t) \approx \widehat{g}_r(\theta_t) = -\rho \nabla \mathcal{L}_f(\theta_t) + \kappa H_{\mathcal{D}}(\theta^*) \Delta_t, \qquad \rho = \frac{m}{n-m}, \quad \kappa = \frac{n}{n-m}, \tag{33}$$

where $H_{\mathcal{D}}(\theta^*) = \nabla^2 \mathcal{L}_{\mathcal{D}}(\theta^*)$ denotes the full-data Hessian at θ^*.

Proof. To start with, let $\Delta_t = \theta_t - \theta^*$. By applying the first-order Taylor expansion to the retaining and forgetting gradients around θ^*, we have:

$$\begin{cases} \nabla \mathcal{L}_r(\theta_t) \approx \nabla \mathcal{L}_r(\theta^*) + H_r(\theta^*) \Delta_t, \\ \nabla \mathcal{L}_f(\theta_t) \approx \nabla \mathcal{L}_f(\theta^*) + H_f(\theta^*) \Delta_t, \end{cases} \tag{34}$$

where $H_r(\theta^*) = \nabla^2 \mathcal{L}_r(\theta^*)$ and $H_f(\theta^*) = \nabla^2 \mathcal{L}_f(\theta^*)$. The second relation gives:

$$\nabla \mathcal{L}_f(\theta^*) \approx \nabla \mathcal{L}_f(\theta_t) - H_f(\theta^*) \Delta_t. \tag{35}$$

According to Proposition 1, the retaining and forgetting gradients at the original stationary point satisfy:

$$\nabla \mathcal{L}_r(\theta^*) = -\rho \nabla \mathcal{L}_f(\theta^*), \qquad \rho = \frac{m}{n-m}. \tag{36}$$

Then, substituting this relation and the approximation of $\nabla \mathcal{L}_f(\theta^*)$ into the Taylor expansion of $\nabla \mathcal{L}_r(\theta_t)$, we obtain:

$$\begin{aligned} \nabla \mathcal{L}_r(\theta_t) &\approx \nabla \mathcal{L}_r(\theta^*) + H_r(\theta^*) \Delta_t \\ &\approx -\rho \nabla \mathcal{L}_f(\theta^*) + H_r(\theta^*) \Delta_t \\ &\approx -\rho \left(\nabla \mathcal{L}_f(\theta_t) - H_f(\theta^*) \Delta_t \right) + H_r(\theta^*) \Delta_t \\ &\approx -\rho \nabla \mathcal{L}_f(\theta_t) + \left(H_r(\theta^*) + \rho H_f(\theta^*) \right) \Delta_t. \end{aligned} \tag{37}$$

We next re-express the combined curvature term using the loss decomposition from Proposition 1:

$$\mathcal{L}_{\mathcal{D}}(\theta) = \frac{n-m}{n} \mathcal{L}_r(\theta) + \frac{m}{n} \mathcal{L}_f(\theta), \tag{38}$$

and taking the second-order derivative with respect to θ, we obtain:

$$H_{\mathcal{D}}(\theta) = \frac{n-m}{n} H_r(\theta) + \frac{m}{n} H_f(\theta). \tag{39}$$

Evaluating it at θ^* gives:

$$H_{\mathcal{D}}(\theta^*) = \frac{n-m}{n} H_r(\theta^*) + \frac{m}{n} H_f(\theta^*). \tag{40}$$

Multiplying both sides by $\frac{n}{n-m}$, we obtain:

$$\frac{n}{n-m}H_{\mathcal{D}}(\theta^*) = H_r(\theta^*) + \frac{m}{n-m}H_f(\theta^*). \tag{41}$$

Since $\rho = \frac{m}{n-m}$, let $\kappa = \frac{n}{n-m}$, we have:

$$H_r(\theta^*) + \rho H_f(\theta^*) = \kappa H_{\mathcal{D}}(\theta^*). \tag{42}$$

Then, we substitute the value from Eq. (42) for the curvature term in Eq. (37) to obtain the final local gradient approximation:

$$\nabla\mathcal{L}_r(\theta_t) \approx -\rho\nabla\mathcal{L}_f(\theta_t) + \kappa H_{\mathcal{D}}(\theta^*)\Delta_t. \tag{43}$$

Therefore, we conclude the proof of Proposition 2. □

D.3 Proof of Proposition 3

Proposition 3. (Bounded Proxy-Retain Gradient Error for Full-Data Hessian) *Let θ^* denote the pretrained model optimized over the full dataset $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f$, and let $\widehat{g}_r(\theta_t) = -\rho\nabla\mathcal{L}_f(\theta_t) + \kappa H_{\mathcal{D}}(\theta^*)(\theta_t - \theta^*)$ be the proxy retaining gradient constructed using the full-data Hessian $H_{\mathcal{D}}(\theta^*)$, where $\rho = m/(n-m)$ and $\kappa = n/(n-m)$. Under the local smoothness conditions of Assumption 1, for any unlearning iterate θ_t in a neighborhood of θ^*, the proxy error satisfies*

$$\|\widehat{g}_r(\theta_t) - \nabla\mathcal{L}_r(\theta_t)\| \le \delta_t^H \le (\rho G + \epsilon_H)\|\theta_t - \theta^*\|, \tag{44}$$

where δ_t^H bounds the Hessian-based proxy error, G is the local smoothness constant, and ϵ_H bounds the discrepancy between the scaled full-data and remaining-data Hessians.

Proof. Following the first-order Taylor approximations in Proposition 2, the retaining gradient around θ^* satisfies

$$\nabla\mathcal{L}_r(\theta_t) \approx \nabla\mathcal{L}_r(\theta^*) + H_r(\theta^*)(\theta_t - \theta^*), \tag{45}$$

where $H_r(\theta^*)$ is the Hessian of $\mathcal{L}_r$ at θ^*.

By Proposition 1, at θ^* we have $\nabla\mathcal{L}_r(\theta^*) = -\rho\nabla\mathcal{L}_f(\theta^*)$. Substituting this into the Taylor approximation gives $\nabla\mathcal{L}_r(\theta_t) \approx -\rho\nabla\mathcal{L}_f(\theta^*) + H_r(\theta^*)(\theta_t - \theta^*)$.

The proxy gradient is defined as $\widehat{g}_r(\theta_t) = -\rho\nabla\mathcal{L}_f(\theta_t) + \kappa H_{\mathcal{D}}(\theta^*)(\theta_t - \theta^*)$.

Let $\Delta_t = \theta_t - \theta^*$. Subtracting the true gradient gives the proxy error:

$$\begin{aligned} e_t^H &:= \widehat{g}_r(\theta_t) - \nabla\mathcal{L}_r(\theta_t) \\ &\approx -\rho\left(\nabla\mathcal{L}_f(\theta_t) - \nabla\mathcal{L}_f(\theta^*)\right) \\ &\quad + \left(\kappa H_{\mathcal{D}}(\theta^*) - H_r(\theta^*)\right)\Delta_t \\ &= \Delta_{\text{grad}} + \Delta_H. \end{aligned} \tag{46}$$

By the smoothness of $\mathcal{L}_f$,

$$\begin{aligned} \|\Delta_{\text{grad}}\| &= \| - \rho(\nabla\mathcal{L}_f(\theta_t) - \nabla\mathcal{L}_f(\theta^*))\| \\ &\le \rho G\|\theta_t - \theta^*\|. \end{aligned} \tag{47}$$

We next bound the curvature discrepancy. Since $\kappa = n/(n-m)$, we have $\kappa - 1 = m/(n-m) = \rho$. The difference between the scaled full-data Hessian and the remaining-data Hessian can be decomposed as

$$\begin{aligned} &\kappa H_{\mathcal{D}}(\theta^*) - H_r(\theta^*) \\ &= (\kappa - 1)H_{\mathcal{D}}(\theta^*) + \left(H_{\mathcal{D}}(\theta^*) - H_r(\theta^*)\right). \end{aligned} \tag{48}$$

By the triangle inequality and the homogeneity of matrix norms,

$$\begin{aligned} &\|\kappa H_{\mathcal{D}}(\theta^*) - H_r(\theta^*)\| \\ &\le \|(\kappa - 1)H_{\mathcal{D}}(\theta^*)\| + \|H_{\mathcal{D}}(\theta^*) - H_r(\theta^*)\| \\ &= \frac{m}{n-m}\|H_{\mathcal{D}}(\theta^*)\| + \|H_{\mathcal{D}}(\theta^*) - H_r(\theta^*)\| \\ &\triangleq \epsilon_H. \end{aligned} \tag{49}$$

In the unlearning setting, when $m \ll n$, the additional scaling term $\frac{m}{n-m}\|H_{\mathcal{D}}(\theta^*)\|$ is relatively small provided that the full-data Hessian norm is bounded. By the sub-multiplicativity of matrix norms, the curvature discrepancy term satisfies

$$\begin{aligned}\|\Delta_H\| &= \|(\kappa H_{\mathcal{D}}(\theta^*) - H_r(\theta^*))(\theta_t - \theta^*)\| \\ &\leq \|\kappa H_{\mathcal{D}}(\theta^*) - H_r(\theta^*)\|\|\theta_t - \theta^*\| \\ &\leq \left(\frac{m}{n-m}\|H_{\mathcal{D}}(\theta^*)\| + \|H_{\mathcal{D}}(\theta^*) - H_r(\theta^*)\|\right)\|\theta_t - \theta^*\| \\ &= \epsilon_H\|\theta_t - \theta^*\|.\end{aligned} \tag{50}$$

Combining the two terms under the local Taylor approximation gives

$$\begin{aligned}\|e_t^H\| &\leq \|\Delta_{\text{grad}}\| + \|\Delta_H\| \\ &\leq (\rho G + \epsilon_H)\|\theta_t - \theta^*\| \\ &= \delta_t^H.\end{aligned} \tag{51}$$

Therefore, we conclude the proof of Proposition 3. □

D.4 Proof of Proposition 4

Proposition 4. (Empirical Fisher-Based Approximation of the Hessian) *Under the likelihood view, the empirical Fisher matrix $F_{\mathcal{D}}^{\text{emp}}(\theta^*)$ is computed from the gradients of observed training instances:*

$$F_{\mathcal{D}}^{\text{emp}}(\theta^*) = \frac{1}{n}\sum_{a=1}^{n} g_a(\theta^*)g_a(\theta^*)^\top, \quad \textit{with} \quad g_a(\theta^*) = \nabla_\theta \ell(z_a;\theta^*). \tag{52}$$

Its diagonal is used to approximate the diagonal Hessian of the full-data objective:

$$\text{diag}(H_{\mathcal{D}}(\theta^*)) \approx \text{diag}(F_{\mathcal{D}}^{\text{emp}}(\theta^*)). \tag{53}$$

Proof. Consider the full-data objective defined as the sample average of per-instance losses:

$$\mathcal{L}_{\mathcal{D}}(\theta) = \frac{1}{n}\sum_{a=1}^{n}\ell(z_a;\theta). \tag{54}$$

For a model optimized via maximum likelihood estimation (MLE), the loss $\ell(z_a;\theta)$ is typically interpreted as the negative log-likelihood $-\log p(z_a \mid \theta)$. In the context of BPR-based recommendation, each instance $z_a = (u, i, j)$ represents a pairwise preference, and the BPR loss is formulated as:

$$\ell_{\text{BPR}}(z_a;\theta) = -\log\sigma(\hat{y}_{ui} - \hat{y}_{uj}) = -\log p(i \succ_u j \mid \theta). \tag{55}$$

The per-instance gradient is thus $g_a(\theta) = -\nabla_\theta \log p(z_a \mid \theta)$.

According to the Fisher Information Identity, under the assumption that the model is correctly specified and has converged to the optimum θ^*, the Hessian of the log-likelihood is related to the expected (standard) Fisher Information Matrix F_{std} by:

$$\mathbb{E}_{z\sim p(\cdot|\theta^*)}\left[\nabla_\theta^2(-\log p(z \mid \theta^*))\right] = \mathbb{E}_{z\sim p(\cdot|\theta^*)}[g(\theta^*)g(\theta^*)^\top] = F_{\text{std}}(\theta^*). \tag{56}$$

In practice, calculating the expectation over the entire distribution is often computationally prohibitive. We thus employ the *Empirical Fisher Information Matrix (EFIM)*, which replaces the model's predictive distribution with the empirical distribution of the training data. The EFIM is defined as the average of the outer products of the gradients:

$$F_{\mathcal{D}}^{\text{emp}}(\theta^*) = \frac{1}{n}\sum_{a=1}^{n} g_a(\theta^*)g_a(\theta^*)^\top. \tag{57}$$

The diagonal of the EFIM consists of the sum of squared coordinates of the per-sample gradients:

$$[\text{diag}(F_{\mathcal{D}}^{\text{emp}}(\theta^*))]_k = \frac{1}{n}\sum_{a=1}^{n}(g_{a,k}(\theta^*))^2, \tag{58}$$

where $g_{a,k}$ is the k-th component of the gradient vector for instance z_a.

By adopting the diagonal empirical Fisher as a proxy for the Hessian curvature, we obtain a tractable approximation:

$$\mathrm{diag}(H_{\mathcal{D}}(\theta^*)) \approx \mathrm{diag}(F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*)). \tag{59}$$

Therefore, we conclude the proof of Proposition 4. □

Discussion. The diagonal of the EFIM provides a tractable approximation to the Hessian diagonal, avoiding explicit Hessian computation while capturing curvature information of the loss landscape near θ^* for likelihood-based objectives such as BPR. The EFIM is widely used in practical applications for parameter importance evaluation due to its lower computational cost [40, 27, 48, 46, 52], which justifies our adoption of the EFIM diagonal as a tractable approximation to the Hessian diagonal.

D.5 Proof of Proposition 5

Proposition 5. (Adam-Based Curvature Proxy) *Let v^* be the squared gradient accumulator stored by Adam during pretraining, and let the joint empirical Fisher diagonal be*

$$\mathrm{diag}\left(F_{\mathcal{D}}^{\mathrm{joint,emp}}(\theta^*)\right) = \mathbb{E}_{\mathcal{B}}\left[\left(\sum_{a\in\mathcal{B}} g_a(\theta^*)\right)^2\right]. \tag{60}$$

Then, the diagonal of the full-data Hessian $H_{\mathcal{D}}(\theta^*)$ can be approximated via the mapping:

$$\mathrm{diag}(H_{\mathcal{D}}(\theta^*)) \xrightarrow{\textit{FIM}} \mathrm{diag}(F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*)) \xrightarrow{\textit{Joint}} \mathrm{diag}(F_{\mathcal{D}}^{\mathrm{joint,emp}}(\theta^*)) \xrightarrow{\textit{Adam}} v^*. \tag{61}$$

In Eq. (33)*, using a diagonal curvature approximation, the curvature term in the proxy gradient is*

$$H_{\mathcal{D}}(\theta^*)\Delta_t \approx v^* \odot \Delta_t, \quad \Delta_t = \theta_t - \theta^*. \tag{62}$$

Proof. From Proposition 4, the Hessian diagonal is approximated by the empirical Fisher:

$$\mathrm{diag}(H_{\mathcal{D}}(\theta^*)) \approx \mathrm{diag}(F_{\mathcal{D}}^{\mathrm{emp}}(\theta^*)) = \frac{1}{n}\sum_{a=1}^{n} g_a(\theta^*)^2. \tag{63}$$

At pretraining step s, let $\mathcal{B}_s$ denote the sampled batch. The batch gradient is

$$G_s = \nabla_\theta \mathcal{L}_{\mathcal{B}_s}(\theta_s) = \frac{1}{|\mathcal{B}_s|}\sum_{a\in\mathcal{B}_s} g_a(\theta_s), \tag{64}$$

and Adam updates the uncentered second-moment accumulator:

$$v_s = \beta_2 v_{s-1} + (1-\beta_2)G_s^2. \tag{65}$$

Unrolling gives

$$v_s = (1-\beta_2)\sum_{k=1}^{s}\beta_2^{s-k}G_k^2 + \beta_2^s v_0, \quad v_0 = 0. \tag{66}$$

Thus, v_s is an exponential moving average of squared batch gradients, with v^* denoting the final accumulator.

Meanwhile, the empirical Fisher diagonal has a square-then-average structure:

$$\mathrm{diag}(F_{\mathcal{D}}^{\mathrm{emp}}(\theta)) = \frac{1}{n}\sum_{a=1}^{n} g_a(\theta)^2. \tag{67}$$

Adam, instead, accumulates the square of the averaged batch gradient:

$$\begin{aligned} G_{\mathcal{B}}(\theta)^2 &= \left(\frac{1}{|\mathcal{B}|}\sum_{a\in\mathcal{B}} g_a(\theta)\right)^2 \\ &= \frac{1}{|\mathcal{B}|^2}\left(\sum_{a\in\mathcal{B}} g_a(\theta)^2 + \sum_{\substack{a,b\in\mathcal{B}\\ a\neq b}} g_a(\theta)\odot g_b(\theta)\right), \end{aligned} \tag{68}$$

where $\odot$ denotes element-wise product. Hence, Adam implicitly captures both per-instance squared terms and cross-gradient interactions.

For a batch $\mathcal{B}$ with joint tuples $z_{\mathcal{B}} = \{z_a\}_{a\in\mathcal{B}}$, the joint likelihood is

$$\begin{aligned} p(z_{\mathcal{B}} \mid \theta) &= \prod_{a\in\mathcal{B}} p(z_a \mid \theta), \\ G_{\mathcal{B}}^{\mathrm{sum}}(\theta) &= \sum_{a\in\mathcal{B}} g_a(\theta). \end{aligned} \tag{69}$$

This gives the batch-level joint empirical Fisher [33]:

$$\begin{aligned} F_{\mathrm{joint,emp}}(\theta;\mathcal{B}) &= G_{\mathcal{B}}^{\mathrm{sum}}(\theta)\big(G_{\mathcal{B}}^{\mathrm{sum}}(\theta)\big)^{\top}, \\ \mathrm{diag}(F_{\mathrm{joint,emp}}(\theta;\mathcal{B})) &= \left(\sum_{a\in\mathcal{B}} g_a(\theta)\right)^2. \end{aligned} \tag{70}$$

Taking the expectation over training batches, we recover the joint empirical Fisher diagonal defined in Proposition 5:

$$\begin{aligned} &\mathrm{diag}\left(F_{\mathcal{D}}^{\mathrm{joint,emp}}(\theta)\right) \\ &= \mathbb{E}_{\mathcal{B}}\left[\mathrm{diag}(F_{\mathrm{joint,emp}}(\theta;\mathcal{B}))\right] \\ &= \mathbb{E}_{\mathcal{B}}\left[\left(\sum_{a\in\mathcal{B}} g_a(\theta)\right)^2\right]. \end{aligned} \tag{71}$$

Since $G_{\mathcal{B}}(\theta) = \frac{1}{|\mathcal{B}|} G_{\mathcal{B}}^{\mathrm{sum}}(\theta)$, we have

$$G_{\mathcal{B}}(\theta)^2 = \frac{1}{|\mathcal{B}|^2}\,\mathrm{diag}(F_{\mathrm{joint,emp}}(\theta;\mathcal{B})). \tag{72}$$

For batches of fixed size B, taking expectations yields

$$\mathbb{E}_{\mathcal{B}}[G_{\mathcal{B}}(\theta)^2] = \frac{1}{B^2}\,\mathrm{diag}\left(F_{\mathcal{D}}^{\mathrm{joint,emp}}(\theta)\right). \tag{73}$$

Therefore, Adam's second-moment accumulator provides an exponentially weighted estimate of the scaled joint empirical Fisher diagonal over training batches.

The batch-level joint empirical Fisher includes cross-gradient interactions:

$$\mathrm{diag}(F_{\mathrm{joint,emp}}(\theta;\mathcal{B})) = \sum_{a\in\mathcal{B}} g_a(\theta)^2 + \sum_{\substack{a,b\in\mathcal{B}\\ a\neq b}} g_a(\theta)\odot g_b(\theta). \tag{74}$$

When cross-gradient correlations are weak and their expected contribution is negligible, the expected squared batch gradient is approximately

$$\mathbb{E}_{\mathcal{B}}[G_{\mathcal{B}}(\theta)^2] \approx \mathbb{E}_{\mathcal{B}}\left[\frac{1}{|\mathcal{B}|^2}\sum_{a\in\mathcal{B}} g_a(\theta)^2\right]. \tag{75}$$

For uniformly sampled batches of fixed size B, this reduces to

$$\mathbb{E}_{\mathcal{B}}[G_{\mathcal{B}}(\theta)^2] \approx \frac{1}{B}\,\mathrm{diag}(F_{\mathcal{D}}^{\mathrm{emp}}(\theta)). \tag{76}$$

Thus, the squared batch gradients and the empirical Fisher diagonal are related through batch-size scaling and cross-gradient terms. Adam further applies exponential smoothing across training steps.

Consequently, the stored accumulator v^* serves as a practical diagonal curvature proxy rather than an exact Hessian estimate. Neglecting off-diagonal Hessian entries, we adopt the approximation

$$\begin{aligned} \mathrm{diag}(H_{\mathcal{D}}(\theta^*)) &\approx v^*, \\ H_{\mathcal{D}}(\theta^*)\Delta_t &\approx v^* \odot \Delta_t. \end{aligned} \tag{77}$$

Therefore, we conclude the proof of Proposition 5. □

D.6 Proof of Proposition 6

Proposition 6. (Closed-form Adaptive Unlearning Direction) *Let $g_u(\theta_t) = \nabla_\theta \mathcal{L}_u(\mathcal{D}_f, \theta_t)$, and let $\tilde{g}_r(\theta_t)$ be the SRGE estimate of $\nabla \mathcal{L}_r(\theta_t)$. For the constrained direction search problem in Eq.* (16)*, if $\|\tilde{g}_r(\theta_t)\| > 0$ (to ensure the denominator in Eq.* (79) *is well-defined), the optimal direction is*

$$d_t^* = g_u(\theta_t) + \lambda_t^* \tilde{g}_r(\theta_t), \tag{78}$$

where

$$\lambda_t^* = \left[\frac{-\tilde{g}_r(\theta_t)^\top g_u(\theta_t) - \epsilon_t}{\|\tilde{g}_r(\theta_t)\|^2} \right]_+, \qquad [x]_+ = \max(x, 0). \tag{79}$$

Proof. We update the model via $\theta_{t+1} = \theta_t - \alpha_t d_t$, where d_t is a descent direction. To achieve the most efficient descent of the unlearning loss $\mathcal{L}_u$ while strictly controlling the degradation of $\mathcal{L}_r$, we solve for d_t by maximizing its alignment with the negative gradient direction under the retaining constraint. For notational simplicity, let

$$g_t = g_u(\theta_t), \qquad r_t = \tilde{g}_r(\theta_t). \tag{80}$$

Then the constrained direction search problem in Eq. (16) can be written as

$$\begin{aligned} \max_{d_t} \quad & g_t^\top d_t - \frac{1}{2}\|d_t\|^2, \\ \text{s.t.} \quad & r_t^\top d_t \geq -\epsilon_t. \end{aligned} \tag{81}$$

The objective is strictly concave with respect to d_t, and the constraint is affine. Since $\|r_t\| > 0$, the feasible set is non-empty. Therefore, the Karush-Kuhn-Tucker (KKT) conditions are both necessary and sufficient for the unique global optimum.

Rewriting the constraint in standard inequality form:

$$c(d_t) = -r_t^\top d_t - \epsilon_t \leq 0, \tag{82}$$

we introduce a non-negative Lagrange multiplier $\lambda_t \geq 0$. The Lagrangian is

$$\mathcal{L}(d_t, \lambda_t) = g_t^\top d_t - \frac{1}{2}\|d_t\|^2 - \lambda_t c(d_t) = g_t^\top d_t - \frac{1}{2}\|d_t\|^2 + \lambda_t(r_t^\top d_t + \epsilon_t). \tag{83}$$

Stationarity condition gives

$$\nabla_{d_t} \mathcal{L}(d_t, \lambda_t) = g_t - d_t + \lambda_t r_t = 0 \quad \Longrightarrow \quad d_t = g_t + \lambda_t r_t. \tag{84}$$

Complementary slackness requires

$$\lambda_t(-r_t^\top d_t - \epsilon_t) = 0, \qquad \lambda_t \geq 0. \tag{85}$$

There are two cases:

Case 1: The unconstrained update satisfies the retaining constraint, i.e., $r_t^\top g_t \geq -\epsilon_t$. Then $\lambda_t^* = 0$ and $d_t^* = g_t$.

Case 2: The unconstrained update violates the constraint, i.e., $r_t^\top g_t < -\epsilon_t$. The constraint is active, $r_t^\top d_t = -\epsilon_t$. Substituting Eq. (84) gives

$$r_t^\top(g_t + \lambda_t r_t) = -\epsilon_t \quad \Longrightarrow \quad \lambda_t = \frac{-r_t^\top g_t - \epsilon_t}{\|r_t\|^2}. \tag{86}$$

Combining both cases and enforcing $\lambda_t^* \geq 0$ gives the compact form

$$\lambda_t^* = \left[\frac{-r_t^\top g_t - \epsilon_t}{\|r_t\|^2} \right]_+. \tag{87}$$

Finally, the optimal direction is

$$d_t^* = g_t + \lambda_t^* r_t, \tag{88}$$

which, by substituting back the definitions of g_t and r_t, yields

$$d_t^* = g_u(\theta_t) + \lambda_t^* \tilde{g}_r(\theta_t), \qquad \lambda_t^* = \left[\frac{-\tilde{g}_r(\theta_t)^\top g_u(\theta_t) - \epsilon_t}{\|\tilde{g}_r(\theta_t)\|^2} \right]_+. \tag{89}$$

Therefore, we conclude the proof of Proposition 6. □

D.7 Proof of Proposition 7

Proposition 7. (Approximate Pareto Stationarity of AGCU) *Assume that $\mathcal{L}_u$ and $\mathcal{L}_r$ are differentiable, and that $\mathcal{L}_u$ is G-smooth and bounded below. At each unlearning step, AGCU computes d_t by solving Eq.* (16) *and updates the model via $\theta_{t+1} = \theta_t - \alpha d_t$, with a constant step size $\alpha \le 1/(2G)$. Let the SRGE remaining-data gradient estimate $\tilde{g}_r(\theta_t)$ satisfy $\|\tilde{g}_r(\theta_t) - \nabla\mathcal{L}_r(\theta_t)\| \le \delta_t$, where $\delta_t = \delta_t^H + \delta_t^A$ accounts for the Hessian-based proxy and Adam approximation errors. Then, after t unlearning steps, AGCU satisfies the following approximate Pareto stationarity bound:*

$$\min_{i=1,\dots,t} \min_{(\mu_u,\mu_r)\in\Delta_2} \|\mu_u \nabla\mathcal{L}_u(\theta_i) + \mu_r \nabla\mathcal{L}_r(\theta_i)\| \le O(t^{-1/2}) + O\left(\frac{1}{t}\sum_{i=1}^{t}\delta_i\right), \tag{90}$$

where $\Delta_2 = \{(\mu_u, \mu_r) \mid \mu_u, \mu_r \ge 0,\ \mu_u + \mu_r = 1\}$.

Proof. For notation simplicity, let $g_t = g_u(\theta_t),\ r_t = \tilde{g}_r(\theta_t)$. By Proposition 6, the AGCU direction search admits the closed form:

$$d_t = g_t + \lambda_t r_t, \quad \lambda_t = \left[\frac{-r_t^\top g_t - \epsilon_t}{\|r_t\|^2}\right]_+. \tag{91}$$

Define the normalized direction

$$d_t^n = \frac{1}{1+\lambda_t} d_t = \frac{1}{1+\lambda_t} g_t + \frac{\lambda_t}{1+\lambda_t} r_t. \tag{92}$$

Let

$$\mu_u^t = \frac{1}{1+\lambda_t}, \quad \mu_r^t = \frac{\lambda_t}{1+\lambda_t}. \tag{93}$$

Then $\mu_u^t, \mu_r^t \ge 0$, $\mu_u^t + \mu_r^t = 1$, and thus $(\mu_u^t, \mu_r^t) \in \Delta_2$. Consequently, $d_t^n = \mu_u^t g_t + \mu_r^t r_t$, showing that the normalized AGCU direction is a convex combination of the unlearning gradient and the SRGE proxy.

Next, we bound the accumulated magnitude of the update directions. Since d_t maximizes the proxy-constrained quadratic objective and the zero direction is feasible when $\epsilon_t \ge 0$, we have

$$g_t^\top d_t - \frac{1}{2}\|d_t\|^2 \ge 0 \quad \Longrightarrow \quad g_t^\top d_t \ge \frac{1}{2}\|d_t\|^2. \tag{94}$$

By G-smoothness of $\mathcal{L}_u$ and using $\theta_{t+1} = \theta_t - \alpha d_t$,

$$\begin{aligned} \mathcal{L}_u(\theta_{t+1}) &\le \mathcal{L}_u(\theta_t) - \alpha g_t^\top d_t + \frac{G\alpha^2}{2}\|d_t\|^2, \\ &\le \mathcal{L}_u(\theta_t) - \frac{\alpha}{2}(1 - G\alpha)\|d_t\|^2, \\ &\le \mathcal{L}_u(\theta_t) - \frac{\alpha}{4}\|d_t\|^2. \end{aligned} \tag{95}$$

where the last inequality follows from $\alpha \le 1/(2G)$. Rearranging gives

$$\|d_t\|^2 \le \frac{4}{\alpha}(\mathcal{L}_u(\theta_t) - \mathcal{L}_u(\theta_{t+1})). \tag{96}$$

Summing over $i = 1, \dots, t$ and using the lower bound $\underline{B}$ of $\mathcal{L}_u$ yields

$$\sum_{i=1}^{t}\|d_i\|^2 \le \frac{4}{\alpha}(\mathcal{L}_u(\theta_1) - \underline{B}) = O(1). \tag{97}$$

Moreover, since $\lambda_i \ge 0$,

$$\begin{aligned} \|d_i^n\| &= \left\|\frac{1}{1+\lambda_i} d_i\right\| \le \|d_i\|, \\ \sum_{i=1}^{t}\|d_i^n\|^2 &\le O(1), \\ \frac{1}{t}\sum_{i=1}^{t}\|d_i^n\|^2 &\le O(t^{-1}), \\ \min_{i=1,\dots,t}\|d_i^n\| &\le O(t^{-1/2}). \end{aligned} \tag{98}$$

To transfer the proxy stationarity bound to the true retaining gradient, we observe from the convex combination that

$$d_i^n = \mu_u^i \nabla \mathcal{L}_u(\theta_i) + \mu_r^i r_i. \tag{99}$$

Using the SRGE error bound, it follows

$$\begin{aligned} \left\| \mu_u^i \nabla \mathcal{L}_u(\theta_i) + \mu_r^i \nabla \mathcal{L}_r(\theta_i) \right\| &= \left\| d_i^n + \mu_r^i (\nabla \mathcal{L}_r(\theta_i) - r_i) \right\| \\ &\leq \|d_i^n\| + \mu_r^i \|\nabla \mathcal{L}_r(\theta_i) - r_i\| \\ &\leq \|d_i^n\| + \delta_i. \end{aligned} \tag{100}$$

Finally, taking the best iterate and the best convex combination over Δ_2, we obtain

$$\begin{aligned} &\min_{i=1,\ldots,t} \min_{(\mu_u,\mu_r)\in\Delta_2} \|\mu_u \nabla \mathcal{L}_u(\theta_i) + \mu_r \nabla \mathcal{L}_r(\theta_i)\| \\ &\leq \frac{1}{t} \sum_{i=1}^{t} (\|d_i^n\| + \delta_i) \\ &\leq \sqrt{\frac{1}{t} \sum_{i=1}^{t} \|d_i^n\|^2} + \frac{1}{t} \sum_{i=1}^{t} \delta_i \\ &\leq O(t^{-1/2}) + O\left(\frac{1}{t} \sum_{i=1}^{t} \delta_i \right). \end{aligned} \tag{101}$$

Therefore, we conclude the proof of Proposition 7. □

We further analyze the effects of practical SRGE approximation errors and residual optimization errors on the Pareto stationarity bound in Appendix D.8.

D.8 Practical SRGE Error Analysis

Total SRGE Approximation Error. Proposition 3 analyzes the Hessian-based proxy gradient

$$\widehat{g}_r^H(\theta_t) = -\rho \nabla \mathcal{L}_f(\theta_t) + \kappa H_{\mathcal{D}}(\theta^*) \Delta_t, \tag{102}$$

where $\Delta_t = \theta_t - \theta^*$. In practice, AdaSRU replaces the Hessian curvature term with Adam's second-moment accumulator v^*:

$$\widetilde{g}_r(\theta_t) = -\rho \nabla \mathcal{L}_f(\theta_t) + \kappa (v^* \odot \Delta_t). \tag{103}$$

Let $V^* = \mathrm{Diag}(v^*)$. By the triangle inequality, the practical SRGE estimation error satisfies

$$\begin{aligned} &\|\widetilde{g}_r(\theta_t) - \nabla \mathcal{L}_r(\theta_t)\| \\ &\leq \|\widetilde{g}_r(\theta_t) - \widehat{g}_r^H(\theta_t)\| \\ &\quad + \|\widehat{g}_r^H(\theta_t) - \nabla \mathcal{L}_r(\theta_t)\|. \end{aligned} \tag{104}$$

The first term represents the additional Adam-based curvature approximation error:

$$\begin{aligned} \delta_t^A &= \|\widetilde{g}_r(\theta_t) - \widehat{g}_r^H(\theta_t)\| \\ &= \kappa \|(V^* - H_{\mathcal{D}}(\theta^*))\Delta_t\| \\ &\leq \kappa \|V^* - H_{\mathcal{D}}(\theta^*)\| \|\Delta_t\|. \end{aligned} \tag{105}$$

The second term is the Hessian-based proxy error bounded by δ_t^H in Proposition 3. Therefore, under its stated error bound,

$$\begin{aligned} &\|\widetilde{g}_r(\theta_t) - \nabla \mathcal{L}_r(\theta_t)\| \\ &\leq \delta_t^H + \delta_t^A \\ &\leq (\rho G + \epsilon_H + \kappa \|V^* - H_{\mathcal{D}}(\theta^*)\|) \, \|\Delta_t\|. \end{aligned} \tag{106}$$

Thus, the practical SRGE error consists of the Hessian-based proxy error and the additional Adam curvature approximation error.

Residual Error under Approximate Stationarity. The preceding analysis assumes that the pretrained parameters satisfy the stationary condition $\nabla\mathcal{L}_{\mathcal{D}}(\theta^*) = 0$. In practice, early stopping introduces a residual optimization error, which can be characterized by

$$\|\nabla\mathcal{L}_{\mathcal{D}}(\theta^*)\| \leq \epsilon_0. \tag{107}$$

From the decomposition of the full-data objective, we have

$$\begin{aligned}\nabla\mathcal{L}_{\mathcal{D}}(\theta^*) = \frac{n-m}{n}\nabla\mathcal{L}_r(\theta^*) \\ + \frac{m}{n}\nabla\mathcal{L}_f(\theta^*).\end{aligned} \tag{108}$$

Therefore, we obtain

$$\nabla\mathcal{L}_r(\theta^*) = -\rho\nabla\mathcal{L}_f(\theta^*) + \kappa\nabla\mathcal{L}_{\mathcal{D}}(\theta^*), \tag{109}$$

where $\rho = m/(n-m)$ and $\kappa = n/(n-m)$. Hence, the additional first-order residual satisfies

$$\|\kappa\nabla\mathcal{L}_{\mathcal{D}}(\theta^*)\| \leq \kappa\epsilon_0. \tag{110}$$

Accordingly, the proxy error analysis under approximate stationarity introduces an additional bounded term $\kappa\epsilon_0$. Under the same local approximation used in Proposition 3, the practical SRGE error becomes

$$\begin{aligned}&\|\widetilde{g}_r(\theta_t) - \nabla\mathcal{L}_r(\theta_t)\| \\ &\lesssim (\rho G + \epsilon_H + \kappa\|V^* - H_{\mathcal{D}}(\theta^*)\|)\,\|\Delta_t\| + \kappa\epsilon_0.\end{aligned} \tag{111}$$

In practice, the observed residual gradient norm remains small. Since κ is close to one, the additional term $\kappa\epsilon_0$ remains small in the overall error bound.

Extension to Approximate Pareto Stationarity. Proposition 7 assumes that the practical SRGE estimator satisfies $\|\widetilde{g}_r(\theta_t) - \nabla\mathcal{L}_r(\theta_t)\| \leq \delta_t$. Under exact pretraining stationarity, the total estimation error is bounded by $\delta_t^H + \delta_t^A$. With early stopping, the residual optimization error contributes an additional term $\kappa\epsilon_0$. Thus, the extended error bound is $\delta_t^{\mathrm{ES}} = \delta_t^H + \delta_t^A + \kappa\epsilon_0$.

Substituting this error allowance into Proposition 7 gives

$$\begin{aligned}&\min_{i=1,\ldots,t}\ \min_{(\mu_u,\mu_r)\in\Delta_2} \|\mu_u\nabla\mathcal{L}_u(\theta_i) + \mu_r\nabla\mathcal{L}_r(\theta_i)\| \\ &\leq O(t^{-1/2}) + O\left(\frac{1}{t}\sum_{i=1}^{t}(\delta_i^H + \delta_i^A)\right) + O(\kappa\epsilon_0).\end{aligned} \tag{112}$$

Therefore, the residual optimization error introduces an additional constant term in the stationarity bound without changing the convergence rate of the optimization term.

E Empirical Study

E.1 Datasets

We conduct experiments on three commonly used real-world datasets: *MovieLens-1M*[4], *MovieLens-10M*[5], and *Yelp2018*[6]. These datasets are widely adopted for benchmarking recommendation models and vary significantly in both scale and sparsity. For consistency, all explicit ratings are binarized to indicate whether a user has interacted with an item. Statistical details are summarized in Table 4.

E.2 Compared Methods

Recommendation Models. We consider three commonly adopted recommendation models: (1) **BPR** [44], which optimizes matrix factorization via the Bayesian Personalized Ranking objective and is widely applied for top-K recommendation; (2) **WMF** [19, 7], employing weighted matrix

[4] `https://grouplens.org/datasets/movielens/1m/`

[5] `https://grouplens.org/datasets/movielens/10m/`

[6] `https://business.yelp.com/data/resources/open-dataset/`

Table 4: Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	Density
MovieLens-1M	6,940	3,706	1,000,209	3.89%
MovieLens-10M	71,567	10,681	10,000,054	1.31%
Yelp2018	31,668	38,048	1,561,406	0.13%

factorization to capture user–item interactions while treating missing entries as weighted negatives; (3) **LightGCN** [18], a simplified GNN model that removes feature transformations and nonlinearities to better capture high-order collaborative signals.

Unlearning Baselines. We compare our proposed **AdaSRU** with a set of representative unlearning baselines: (1) **Retrain** removes revoked samples and retrains the model from scratch as a straightforward baseline; (2) **SISA** [3] partitions the training data into independent shards, trains sub-models separately, and aggregates their predictions to enable efficient unlearning; (3) **RecEraser** [8] extends SISA for recommendation tasks, mitigating performance drops due to data sharding; (4) **CEU** [51] speeds up unlearning by updating the parameters of a pre-trained GNN; (5) **RRL** [58] reverses the effect of forgotten interactions on the model through targeted gradient manipulation; (6) **UnlearnRec** [9] leverages an influence encoder to directly generate unlearned model parameters from unlearning requests and existing parameters; (7) **CSFU** [2] removes the influence of forgotten data without the original training set, by leveraging a surrogate dataset and noise calibrated according to the statistical distance between surrogate and source distributions; (8) **ETR** [57] achieves unlearning without additional training by strategically editing parameters and applying gradient approximation.

E.3 Environment

Most experiments are conducted on Linux servers equipped with an Intel(R) Core(TM) i9-14900KF CPU @ 6.00GHz, 256GB RAM, and NVIDIA GeForce RTX 4090 GPUs. The code is executed using Python 3.8. This setup provides a high-performance environment for computationally intensive tasks.

F Implementation Details of the Membership Inference Oracle (MIO)

Training. We implement the MIO as a three-layer fully connected neural network for binary membership inference. Following [5], MIO uses user and item embeddings to capture collaborative signals from queried interactions. The network consists of a 64-unit input layer, a 16-unit hidden layer with ReLU activation, and a Softmax output layer for binary membership classification. It is trained using stochastic gradient descent (SGD) for 100 epochs at a learning rate of 0.001 to maximize membership classification accuracy on the labeled training and validation split.

Evaluation. To evaluate unlearning completeness, we use MIO to assess whether forgotten interactions remain identifiable as training members. Ideally, an unlearned model should treat forgotten interactions as unseen, making them indistinguishable from genuine non-members. In a white-box setting, the attacker accesses the dataset and model parameters and performs membership inference using user and item embeddings. We report (1) *Accuracy* (ACC) for membership classification and (2) *Area Under the ROC Curve* (AUC) for distinguishing members from non-members based on predicted membership probabilities [16]. ACC and AUC values closer to 0.5 indicate lower distinguishability between forgotten interactions and non-members, reflecting more complete unlearning.

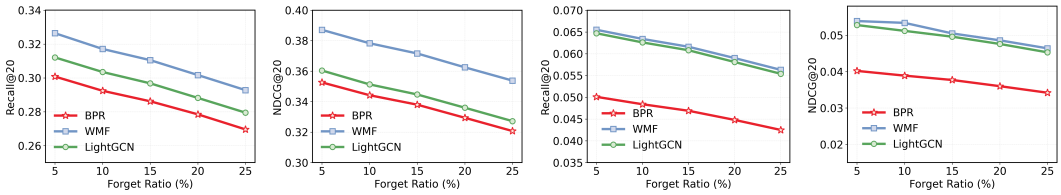
G Additional Experimental Results

G.1 Ablation Study

We conduct ablation studies on ML-1M, ML-10M, and Yelp2018 to evaluate the contribution of SRGE, curvature-aware correction, and AGCU in **AdaSRU**. As shown in Table 5, "w/o" denotes removing the corresponding component from the full **AdaSRU**. Specifically, *w/o SRGE* removes source-free remaining-data gradient estimation, *w/o Curvature* removes the Adam-statistics-based curvature correction, and *w/o AGCU* replaces the adaptive constrained update with a fixed weighted update. Across the three datasets, removing SRGE leads to the largest performance drop, indicating

Table 5: Ablation results of **AdaSRU** on three datasets. R@K and N@K denote Recall@K and NDCG@K, respectively.

Dataset	Variant	BPR				WMF				LightGCN			
		R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
ML-1M	w/o SRGE	0.1398	0.2259	0.3195	0.3172	0.1411	0.2304	0.3310	0.3268	0.1418	0.2249	0.3260	0.3205
	w/o Curvature	0.1435	0.2312	0.3264	0.3241	0.1456	0.2389	0.3392	0.3357	0.1462	0.2314	0.3351	0.3298
	w/o AGCU	0.1444	0.2338	0.3290	0.3266	0.1468	0.2405	0.3411	0.3379	0.1471	0.2330	0.3374	0.3315
	AdaSRU	0.1472	0.2369	0.3326	0.3309	0.1537	0.2447	0.3456	0.3429	0.1505	0.2367	0.3421	0.3368
ML-10M	w/o SRGE	0.1795	0.2807	0.3214	0.3316	0.1986	0.3032	0.3578	0.3657	0.1876	0.2892	0.3364	0.3382
	w/o Curvature	0.1835	0.2864	0.3274	0.3372	0.2027	0.3098	0.3638	0.3711	0.1921	0.2963	0.3427	0.3450
	w/o AGCU	0.1851	0.2889	0.3301	0.3405	0.2046	0.3124	0.3661	0.3740	0.1936	0.2991	0.3453	0.3482
	AdaSRU	0.1886	0.2924	0.3337	0.3442	0.2084	0.3171	0.3699	0.3783	0.1973	0.3035	0.3495	0.3513
Yelp2018	w/o SRGE	0.0259	0.0457	0.0295	0.0368	0.0347	0.0592	0.0395	0.0489	0.0340	0.0585	0.0390	0.0482
	w/o Curvature	0.0265	0.0470	0.0302	0.0377	0.0356	0.0610	0.0409	0.0501	0.0349	0.0602	0.0402	0.0493
	w/o AGCU	0.0268	0.0476	0.0305	0.0381	0.0360	0.0617	0.0414	0.0507	0.0354	0.0611	0.0407	0.0500
	AdaSRU	0.0284	0.0496	0.0325	0.0392	0.0371	0.0631	0.0428	0.0519	0.0369	0.0636	0.0424	0.0512



(a) ML-10M Recall@20 (b) ML-10M NDCG@20 (c) Yelp2018 Recall@20 (d) Yelp2018 NDCG@20

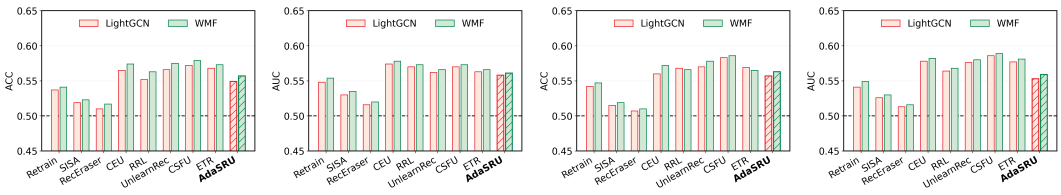
Figure 3: Effect of different forgetting ratios on ML-10M and Yelp2018 under three recommendation backbones.

that estimating the remaining-data gradient is crucial for preserving recommendation utility in the source-free setting. Removing the curvature-aware correction also degrades performance, showing that the Adam-statistics-based curvature proxy helps correct the deviation from the pretrained model during unlearning. In addition, replacing AGCU with a fixed weighted update consistently reduces Recall and NDCG, suggesting that the adaptive constrained update better balances unlearning and utility preservation than a static trade-off. The performance drop is more evident on ML-1M and ML-10M, where richer collaborative signals make gradient estimation and constrained updates more beneficial, while the degradation on Yelp2018 is relatively smaller in absolute value due to its higher sparsity. Overall, the full **AdaSRU** achieves the best performance across datasets and backbones, confirming that the three components are jointly effective.

G.2 Effect of Forgetting Ratio on Additional Datasets

We further evaluate the sensitivity of **AdaSRU** to different forgetting ratios on ML-10M and Yelp2018. Specifically, we vary the forgetting ratio from 5% to 25% and report Recall@20 and NDCG@20 under BPR, WMF, and LightGCN backbones. From Fig. 3, Recall@20 and NDCG@20 consistently decrease as the forgetting ratio increases across ML-10M and Yelp2018. This is because removing more interactions weakens the collaborative signals available for preserving recommendation utility during unlearning. The decline is smoother on ML-10M, while Yelp2018 is more sensitive due to its higher sparsity. These results show that **AdaSRU** maintains stable utility under different forgetting ratios, but also confirm that source-free recommendation unlearning becomes more challenging when more interactions are removed.

Additional Membership Inference Results. We further report MIO ACC and AUC on ML-10M and Yelp2018 in Fig. 4. **AdaSRU** achieves lower ACC and AUC than CEU, RRL, UnlearnRec, CSFU, and ETR, indicating that it can effectively suppress membership signals without accessing the remaining data. RecEraser obtains the lowest MIO scores due to partition-based reconstruction, but it requires data partitioning and partial retraining. In contrast, **AdaSRU** provides a more practical trade-off between unlearning completeness, recommendation utility, and computational efficiency under the source-free setting.



(a) ML-10M MIO ACC (b) ML-10M MIO AUC (c) Yelp2018 MIO ACC (d) Yelp2018 MIO AUC

Figure 4: Additional membership inference results on ML-10M and Yelp2018. The dashed line denotes the random-guessing level of 0.5; values closer to 0.5 indicate more complete unlearning.

H Limitations

First, since the remaining data are unavailable in the source-free setting, the retaining gradient cannot be computed exactly. Existing post-training unlearning methods usually restrict the deviation from the original embeddings or parameters to preserve recommendation behavior [6, 23, 30]. **AdaSRU** instead estimates the retaining gradient using the forgetting-set gradient, pretrained parameters, and saved optimizer statistics, and constrains the update direction at the gradient level. This enables source-free unlearning, but the estimation remains approximate, especially when the forgetting ratio is large or the unlearning trajectory deviates substantially from the pretrained model.

Second, the curvature-aware correction relies on the saved Adam second-moment accumulator as a diagonal proxy for the full-data curvature. While this design is efficient and avoids accessing source data, it cannot capture full Hessian information or non-diagonal parameter interactions.

Finally, our experiments mainly focus on collaborative filtering backbones and interaction-level unlearning. Extending **AdaSRU** to sequential, multimodal, or user-level recommendation unlearning remains an important direction for future work.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction sections reflect the paper's contributions and scope in the paper.

   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: The paper includes a dedicated limitation discussion in Appendix H, where we discuss the assumptions and practical constraints of the proposed source-free unlearning framework, including the dependence on gradient estimation accuracy and the trade-off between unlearning effectiveness and utility preservation.

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: All theoretical claims are correctly induced with reasonable assumptions and correct proofs.

   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 provide the experimental implementation details in Section 4 of the main paper, the algorithms in Appendix B, and a link to the code in the abstract.

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: [Yes]

   Justification: We provide a link to the code in the abstract, together with instructions and experimental details for reproducing the main results. The datasets used in our experiments are publicly available and properly cited in the paper.

   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 full experimental details with the code, in the main paper and appendix.

   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: We repeat all experiments five times with different random seeds and report the mean to reflect the variability across runs.

   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 provide the experimental hardware and software configurations in Appendix E.3, including the compute workers, memory, and execution environment, and report runtime results in the main paper to reflect the computational cost.

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: This paper focuses on theoretical results and does not involve ethical issues.

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 the broader impacts of recommendation unlearning, including its potential benefits for privacy protection and the right to be forgotten, as well as possible risks if unlearning is incomplete or improperly deployed in real-world recommender systems.

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: This paper poses no such risks.

    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: [N/A]

    Justification: The paper does not use existing assets.

    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: The paper does not release new assets.

    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: The paper does not involve crowdsourcing nor 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: The paper does not involve crowdsourcing nor research with 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 method development in this research 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.