ERASE - A Real-World Aligned Benchmark for Unlearning in Recommender Systems

Pierre Lubitzsch
BIFOLD & TU Berlin
Berlin, Germany
lubitzsch@tu-berlin.de

Maarten de Rijke
University of Amsterdam
Amsterdam, Netherlands
m.derijke@uva.nl

Sebastian Schelter
BIFOLD & TU Berlin
Berlin, Germany
schelter@tu-berlin.de

Abstract

Machine unlearning (MU) enables the removal of selected training data from models to address privacy compliance, security, and liability issues in recommender systems. Existing MU benchmarks poorly reflect real-world recommender settings: they focus primarily on collaborative filtering, assume unrealistically large deletion requests, and overlook practical constraints such as sequential unlearning and efficiency. We present ERASE, a large-scale benchmark for MU in recommender systems designed to align with real-world usage. ERASE spans three core tasks—collaborative filtering, session-based recommendation, and next-basket recommendation—and includes unlearning realistic scenarios such as sequentially removing sensitive interactions or spam. The benchmark covers seven unlearning algorithms, including general-purpose and recommender-specific methods, across nine public datasets and nine state-of-the-art models, producing over 600 GB of reusable artifacts, such as extensive logs and over a thousand model checkpoints. The artifacts that we release enable systematic analyses of where current unlearning methods succeed and where they fail. ERASE shows that approximate unlearning matches retraining in some settings, but robustness varies widely across datasets and architectures. Repeated unlearning exposes weaknesses in general-purpose methods, especially for attention-based and recurrent models, while recommender-specific approaches behave more reliably. ERASE offers an empirical basis for the community to assess, drive, and track progress toward practical MU in recommender systems.

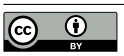
CCS Concepts

• **Computing methodologies** → *Machine learning*; • **Information systems** → **Recommender systems**; • **Security and privacy** → **Human and societal aspects of security and privacy**.

Keywords

Unlearning, Recommender systems, Benchmarking

ACM Reference Format:
Pierre Lubitzsch, Maarten de Rijke, and Sebastian Schelter. 2026. ERASE - A Real-World Aligned Benchmark for Unlearning in Recommender Systems. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3805712.3808596



SIGIR '26, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3808596

1 Introduction

Modern recommender systems rely heavily on user interaction data, such as clicks, ratings, and purchases, to build personalized models. Personalization enhances the user experience, but at the same time makes it challenging to adhere to privacy rights such as the "right-to-be-forgotten" from the GDPR [14] regulation in Europe and similar regulations adopted outside Europe [3, 46]. These regulations mandate that responsible, ethical, and legally compliant AI systems give their users the right to request the timely deletion of their personal data from databases and models trained on it [43, 44]. Being able to efficiently remove data from models is also important for security, e.g., to quickly react to spam interactions [2, 34], unintended data leakage [5], and to handle liability issues, for example when a model consumed copyrighted content [57].

Machine unlearning. The outlined challenges give rise to *machine unlearning* (MU), the problem of efficiently removing the influence of selected training data points from a machine learning model post training. Modern machine learning models are known to encode information about their training data, ranging from statistical influence to explicit memorization of individual examples [6, 60]. After unlearning data points, there should be no information about these points present in the model; it should behave as if they had never been in the training set. Machine unlearning is an active and growing area of research [4, 15, 22, 42, 49, 54, 55]. However, the evaluation of unlearning algorithms is especially challenging since multiple aspects, such as model utility after unlearning, the amount of information about the unlearned data still contained in the model, and efficiency, have to be considered simultaneously.

Shortcomings of existing benchmarks for unlearning in recommender systems. While MU has been studied broadly across machine learning tasks [47], recommender systems pose distinct challenges. Recommendation models are trained on large-scale, highly sparse user–item interaction data, are frequently updated, and often rely on sequential or graph-based architectures. Unlearning requests in recommender systems typically arrive continuously and must be handled under strict latency and cost constraints. Current unlearning benchmarks like CURE4Rec [8] do not sufficiently address the real-world demands of unlearning in recommender systems (Section 3). They focus on collaborative filtering (CF) [17, 36, 37, 39, 56], overlooking tasks that are common in e-commerce and entertainment platforms, such as session-based recommendation (SBR) [18, 23, 25, 29, 45, 53] and next-basket recommendation (NBR) [13, 20, 21, 35, 38, 58]. Existing benchmarks simulate unlearning by deleting large chunks of data—up to 5% of all interactions—whereas actual systems face continuous, small-scale deletion requests, in-between regular retraining cycles. The

choice of data to unlearn is also misrepresented: rather than being random, real scenarios often evolve around specific types of content, like interactions with sensitive items [40] or interactions from spammers [2]. Promising general-purpose unlearning algorithms, e.g., from the NeurIPS'23 unlearning competition [47], remain underexplored for recommendation-specific applications. Operational efficiency [26] is another overlooked factor: unlearning must be fast, inexpensive, and ideally executable directly on deployed models [42]. The unlearning procedures in CURE4Rec [8] are only an order of magnitude faster than retraining, which is not attractive for real-world deployment. These limitations align with feedback that we gathered through interviews with senior team leaders at European companies operating recommender systems with millions of users. They report that data deletion requests are currently only reflected after weekly or monthly retraining cycles (and retraining itself can take up to a day), and expressed concerns about whether existing unlearning methods would be efficient enough for deployment.

ERASE - A real-world aligned benchmark for unlearning in recommender systems. We present ERASE, a real-world aligned benchmark for unlearning in recommender systems, building on our prior work from FAccTRec'25 [33]. We formulate four guiding questions to address shortcomings of existing benchmarks and describe our design decisions in Section 4. We cover NBR and SBR in addition to CF. For each task, we include multiple models, unlearning scenarios, and datasets. We consider approximate unlearning algorithms for neural recommendation models [32], graph neural networks (GNNs) [11, 51, 52], and methods adapted from the NeurIPS'23 unlearning competition [47]. Unlike existing benchmarks, ERASE issues small unlearning requests sequentially instead of unlearning a single forget set at once, reflecting domain-specific and time-sensitive requests. The unlearned models are evaluated based on retained utility, unlearning effectiveness, and computational efficiency.

Contributions. Our contributions are as follows:

- We identify shortcomings of existing unlearning benchmarks and design ERASE, a real-world aligned benchmark for unlearning in recommender systems. ERASE covers three recommendation tasks using two unlearning scenarios with seven unlearning algorithms across nine datasets and nine state-of-the-art models.
- We run ERASE to produce an extensive number of reusable artifacts, enabling researchers to evaluate new unlearning algorithms without additional training of recommenders. We illustrate the value of these artifacts for assessing the current state of knowledge regarding MU for recommender systems.
- We provide our benchmark code, reusable artifacts, and usage guides under an open license at https://github.com/deem-data/erase-bench.

2 Background on Machine Unlearning

Machine unlearning removes the influence of specific data from a trained model. Formally, let D be the training dataset, $\mathcal{A}$ be a training algorithm, U be an unlearning algorithm, $D_f \subseteq D$ be the forget set, and $D_r = D \backslash D_f$ be the retain set. U is an exact unlearning algorithm iff for all measurable $T \subseteq \mathcal{H}$:

$$\mathbb{P}(\mathcal{A}(D_r) \in T) = \mathbb{P}(U(\mathcal{A}(D), D_r, D_f) \in T), \tag{1}$$

where $\mathcal{H}$ is the hypothesis space. The unlearned model should match the distribution of a model retrained from scratch on D_r. For approximate unlearning, U is a (ε, δ)-unlearning algorithm iff for all measurable $T \subseteq \mathcal{H}$:

$$\mathbb{P}(\mathcal{A}(D_r) \in T) \leq \exp(\varepsilon)\, \mathbb{P}(U(\mathcal{A}(D), D_r, D_f) \in T) + \delta \tag{2}$$

and

$$\mathbb{P}(U(\mathcal{A}(D), D_r, D_f) \in T) \leq \exp(\varepsilon)\, \mathbb{P}(\mathcal{A}(D_r) \in T) + \delta, \tag{3}$$

with $\varepsilon \in [0, \infty), \delta \in [0, 1)$ [47]. This formulation is conceptually similar to differential privacy [24], but differs in what is being varied: instead of comparing neighboring datasets, we compare different training procedures. Intuitively, retraining from scratch on D_r provides a gold standard for unlearning, as it guarantees that no information from D_f is retained. However, retraining is computationally expensive. The above definition therefore quantifies how closely an unlearning algorithm approximates this ideal by bounding the multiplicative and additive deviation between the distributions over models produced by retraining and unlearning. In this view, a perfect unlearning algorithm induces the same distribution over the hypothesis space as retraining, assigning the same probabilities to all (measurable) subsets of models.

The multiplicative factor $\exp(\varepsilon)$ is used instead of ε to align with the standard formulation in differential privacy and to yield an additive interpretation in log-probability space. In particular, for random variables A, B, the bound

$$\mathbb{P}(A) \leq \exp(\varepsilon)\mathbb{P}(B)$$

is equivalent to

$$\log \mathbb{P}(A) \leq \log \mathbb{P}(B) + \varepsilon,$$

so that ε can be interpreted as an additive deviation between log-probabilities. Lower ε and δ values indicate stronger unlearning guarantees. $(0, 0)$-unlearning is equivalent to exact unlearning.

3 Motivation for a New Benchmark

We argue that existing unlearning benchmarks for recommender systems, such as CURE4Rec [8], fail to reflect real-world recommendation scenarios.

Lack of coverage of diverse recommendation tasks. In CURE4Rec [8], the only recommendation task considered is collaborative filtering. But the field of recommendation is much broader. There are many sequential recommendation tasks, which make use of temporal information to model the development of a user's preferences over time. Important real-world applications of such tasks include e-commerce and retail, which are typically domains of SBR [18, 23, 25, 29, 45, 53] and NBR [13, 20, 21, 35, 38, 58]. A comprehensive unlearning benchmark should also cover these tasks.

Lack of domain-specific patterns in interactions to unlearn. In some domains, interactions to unlearn follow clear patterns, such as removing sensitive items (e.g., alcohol for users with addiction [40, 41]) or filtering malicious actors like spammers [2]. Current benchmarks do not account for such domain-specific unlearning patterns.

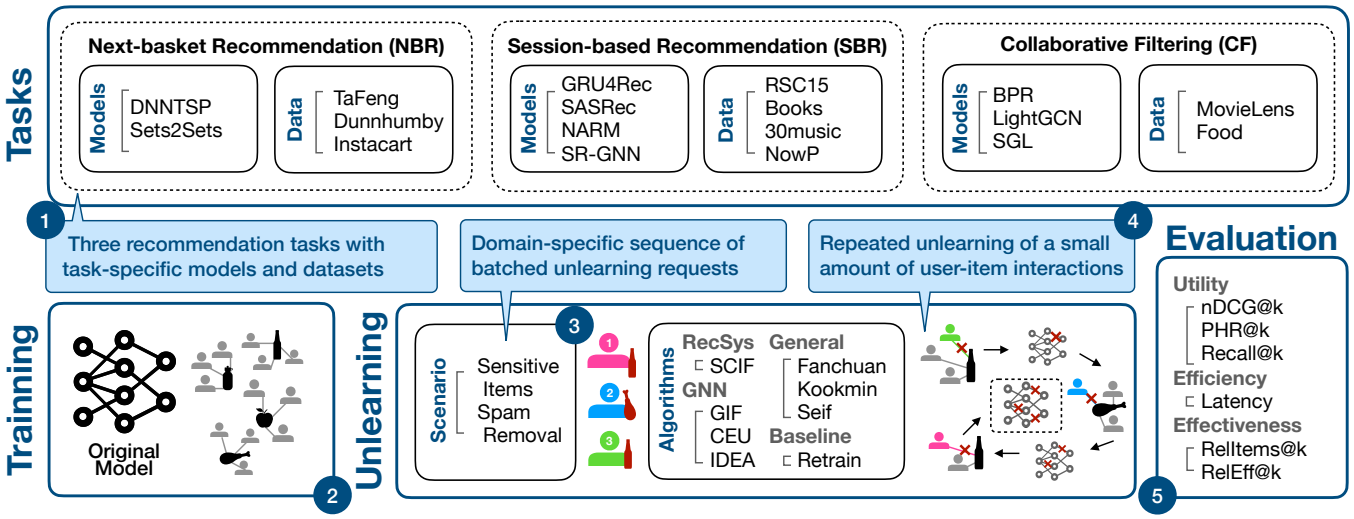


Figure 1: Overview of ERASE. ❶ Each experiment first trains a task-specific model on a chosen dataset ❷. A sequence of unlearning requests is defined ❸ (based on a scenario), and an unlearning algorithm is applied repeatedly ❹. Finally, the unlearned model is evaluated for utility, efficiency, and effectiveness ❺, in comparison to a model retrained on the retain set.

Unrealistic forget set sizes. In real-world settings, unlearning requests typically arrive as many small updates over time, for example when users withdraw consent for personalization [42] or when security teams remove spam users. Such requests should be handled with low latency, ideally within seconds. However, existing benchmarks focus on a single, unrealistically large unlearning request. For example, the NeurIPS'23 unlearning competition [47] and CURE4Rec [8] benchmarks remove up to 5% of the data at once. In practice, since models are retrained periodically anyway (e.g., weekly [1] or monthly [9]), unlearning is most useful for efficiently removing small batches of data between retraining cycles.

Insufficient focus on efficiency. Unlearning needs to be executed fast and with no or negligible utility loss. In CURE4Rec [8], the runtime of unlearning is only about one order of magnitude lower than retraining from scratch, which is not efficient enough for real-world use cases. Models are often trained for a day, according to our interviews, so reducing the time needed for carrying out the unlearning request to seconds or minutes requires a decrease of at least three orders of magnitude.

Insufficient coverage of available unlearning algorithms. The only approximate unlearning method in CURE4Rec is SCIF [32], which conducts a single second-order parameter update. However, the NeurIPS'23 unlearning competition [47] contains several high-scoring first-order iterative approaches to unlearning, which also need to be evaluated for recommendation tasks.

4 The ERASE Benchmark

Design criteria. To address the shortcomings of existing benchmarks and arrive at a wishlist for MU benchmarks for recommender systems, we pose the following guiding questions:

What is the unlearning effectiveness of approximate unlearning methods? Due to a lack of work on evaluating approximate unlearning across different recommendation tasks and scenarios, an overview of the performance of these methods is still missing in literature.

Does approximate unlearning approach achieve robust, consistent performance across CF, SBR, and NBR recommendation tasks? Our preliminary work [33] indicates that unlearning algorithms are highly sensitive to hyperparameter choices. Different unlearning algorithms also have different ways to remove data from the model. An approach with robust performance across multiple tasks and scenarios is desirable. It is an open question whether any single method achieves consistently strong performance across tasks, or whether effective unlearning requires careful hyperparameter and algorithm selection per task.

How does model architecture influence approximate unlearning effectiveness in recommendation systems? Model architecture plays a central role in determining a model's learning capacity and the representations it acquires. Hence, it is crucial to understand how unlearning effectiveness differs across architecture families and whether architecture-specific unlearning methods outperform general approaches.

Do current approximate unlearning algorithms achieve sufficient efficiency for real-world deployment? Efficiency is critical for MU: an approximate method with runtime comparable to full retraining offers little practical value.

These questions guide the design of a real-world aligned benchmark for unlearning in recommender systems: (i) task settings (models and datasets), (ii) training setups, (iii) unlearning scenarios, and (iv) evaluation facilities. Figure 1 provides an overview of the resulting ERASE benchmark.

Recommendation tasks and models. We consider a diverse set of recommendation tasks and representative models. For collaborative filtering (CF), we include classic matrix factorization [37] as well as two graph-based approaches, `LightGCN` [17] and `SGL` [50]. To cover session-based recommendation (SBR), a key task in applications such as next-item prediction in e-commerce and music streaming, we evaluate two recurrent models, `GRU4Rec` [45] and `NARM` [29], the attention-based model `SASRec` [25], and the graph-based model

Table 1: Choice of datasets across recommendation tasks for ERASE. All datasets are publicly available.

Dataset	Domain	Task	#Users/Sessions	#Items	#Interactions	Available at
TaFeng [10]	Grocery shopping	NBR	32,266	23,812	817,741	Kaggle
Dunnhumby [12]	Retail shopping	NBR	2,500	92,339	2,595,732	dunnhumby.com
Instacart [27]	Grocery shopping	NBR	19,435	13,897	3,346,083	GitHub
RSC15 [18]	E-commerce interactions	SBR	9,249,729	52,739	33,003,944	Kaggle
Books [19]	Amazon book reviews	SBR	10,300,000	4,400,000	29,500,000	GitHub
30music [48]	Music streams	SBR	2,764,474	5,600,000	31,351,954	Remap Lab
NowP [59]	Music on social media	SBR	738,200	3,734,862	1,291,251,677	Zenodo
MovieLens [16]	Movie ratings	CF	138,493	26,744	20,000,263	Kaggle
Food [19]	Amazon product reviews	CF	7,000,000	603,200	14,300,000	GitHub

`SR-GNN` [53]. For next-basket recommendation, which focuses on predicting a user's subsequent shopping basket, we include the graph-based model `DNNTSP` [58] and the attention-based model `Sets2Sets` [20].

Datasets. We use widely adopted, publicly available datasets for our benchmark (Table 1). The three NBR datasets relate to grocery or retail shopping. For SBR, two of the four datasets capture e-commerce interactions, while the other two are music datasets derived from social media posts about songs (NowP) or collected from internet radio stations (30music). For CF, we select two datasets containing user ratings for movies and food products.

Unlearning algorithms. We include recommendation-specific as well as general unlearning algorithms in the benchmark.

Custom unlearning algorithms for recommender systems and graph neural networks. We do not include general exact unlearning algorithms like RecEraser [7] or UltraRE [31], because their running time depends on the training set size, which is by design not efficient enough to handle a lot of queries with small forget sets. We focus on approximate unlearning algorithms with a time complexity linear in the forget set size. We include the following algorithms:

- `SCIF` [32] directly removes the influence of data in the forget set using influence functions. The influence of the data in the forget set is calculated similarly to a Newton step, which additionally includes retain data to prevent catastrophic forgetting via $\tilde{\theta}_{z\to\tilde{z}} = \tilde{\theta} - H_{\tilde{\theta}}^{-1}\nabla_{\tilde{\theta}}(\frac{1}{2+bs}(-\ell(z,\theta) + \ell(\tilde{z},\theta) + \sum_{i=1}^{bs}\ell(z_i,\theta)))$ where z_i are retain samples, z is an unlearning sample, $\tilde{z}$ is the modified unlearning sample, H is the Hessian, bs is the number of retain samples used for the update, and $\tilde{\theta}$ is a subset of the parameters θ of the model. `SCIF` uses approximations of second-order information to simultaneously remove knowledge about the forget set while circumventing catastrophic forgetting using samples from the retain set. This update is only calculated and applied for a subset of weights that are manually chosen as most important for the task, depending on the model architecture (e.g., user embeddings for CF).
- `GIF` [51] works similarly to `SCIF`, but the parameters considered for the parameter update are chosen differently: The embedding of node u in the graph is contained in the parameter update if there exists a node v such that v is in the forget set and the distance from u to v is smaller than a hyperparameter d.
- `CEU` [52] unlearns edges from a GNN. To get theoretical guarantees for linear models, `CEU` first adds a linear noise term $b^T\theta$ with $b \sim \mathcal{N}(0, \sigma^2 I)$ to the training loss and fine-tunes aiming to hide the real gradient residual. After that, `CEU` unlearns using an influence function. The intuition behind `CEU` is similar to `SCIF` without parameter subselection, but with adding noise to the parameters to get theoretical guarantees for linear models with strongly convex losses.
- `IDEA` [11] analyzes the original loss over the whole training set and the pruned loss (which excludes nodes and edges we want to unlearn). It then conducts a second-order update step based on the difference between the original and pruned objective loss.

General unlearning algorithms for gradient-based machine learning. We additionally include the following unlearning algorithms from the NeurIPS'23 unlearning challenge [47], which have not specifically been designed for recommendation models. We select these particular approaches, as they scored the highest in the final ranking of the competition:

- `Fanchuan` first minimizes the KL-divergence between the output of the model on the unlearning samples and a uniform pseudo-label. Next, the method varies between two modes: in the first mode, a contrastive loss maximizes the distance between representations of samples in the forget set and samples in the retain set. In the second mode, it fine-tunes the model on a subset of the retain set. The idea behind the first stage is to make the distribution of outputs for forget samples more uniform, which should push them away from other similar samples in the retain set without explicitly using the retain set. The next stage alternates between an explicit contrastive loss to push away embeddings from forget set samples and retain set samples, and a repair stage fine-tuning with the retain set to mitigate catastrophic forgetting.
- `Kookmin` resets a subset of the model parameters to their states before training. This is followed by a fine-tuning stage on a subset of the retain set, where non-reset parameters have a smaller learning rate. The subset of parameters to reset is chosen as the $p\,|\theta|$ parameters having the most similar gradients on the forget set and a subset of the retain set, where $p \in (0, 1)$ is a hyperparameter. Resetting parameters that exhibit similar gradients on the forget and retain sets helps decouple their optimization trajectories during the repair phase. Since these parameters respond similarly to both sets, continuing training from their current values may preserve residual influence from the forget set. By reverting them to their pre-training states and then fine-tuning exclusively on the retain set, the model is encouraged to re-learn these shared components in a way that is solely informed by the retained data. This enables the repair phase to better align the model with the retain set, without being constrained by representations that were previously shaped by the forget set.

- `Seif` adds Gaussian noise to the model parameters, followed by fine-tuning on a subset of the retain set. The underlying intuition is that, when the forget set is small, the optimal parameters of the retrained model remain close in weight space to those of the original model. In this regime, fine-tuning on the retain set alone induces only minor updates, as gradients are already near zero around the original optimum. Injecting noise perturbs the parameters away from this stationary point, thereby enabling the optimization process to escape the original solution and converge toward an optimum defined only by the retain set.

Evaluation metrics. We discuss the dimensions and metrics to evaluate the performance of unlearning algorithms.

Recommendation utility. We report common ranking metrics such as Recall, Personalized Hit Ratio, and Normalized Discounted Cumulative Gain.

Unlearning effectiveness. In addition to post-unlearning utility, it is important to assess to which degree a model actually forgets the forget set in approximate unlearning. This is challenging because most approximate unlearning algorithms lack theoretical guarantees (Section 2). There is no single metric for unlearning effectiveness that is broadly applicable across models while remaining both meaningful and interpretable. Common evaluation approaches compare unlearned and retrained models in weight space or output distribution. While a small distance suggests successful unlearning, a large distance is inconclusive, as model multiplicity and training randomness can yield alternative retrained models that are equally valid yet far apart by these metrics. Consequently, such comparisons cannot reliably determine unlearning effectiveness. Another common evaluation approach uses membership inference attacks to assess unlearning effectiveness. While successful attacks indicate failed unlearning, the absence of successful attacks is inconclusive, as results depend heavily on attacker strength and cannot account for all possible adversaries. Instead of using these evaluation methods, we evaluate unlearning effectiveness with scenario-specific metrics that are easier to interpret and more meaningful for real-world usage. We include the following two scenarios:

- *Sensitive item unlearning* [40]: In this scenario, items from sensitive categories are removed for specific users, e.g., meat for users adopting a vegetarian diet or alcohol for users with addiction issues. Effectiveness is measured by counting how many users still receive sensitive recommendations after the unlearning request, compared to a retrained model. To evaluate recommendation utility, sensitive items are removed from the affected users' test data. Let U_s be the set of users that submitted a forget request for sensitive items, R_u^k be the top k recommended items for user u, and I_s be the set of sensitive items. We define

$$\text{Sensitive@}k := \frac{1}{|U_s|} \sum_{u \in U_s} [R_u^k \cap I_s \neq \emptyset] \in [0, 1]$$

  as the percentage of users who submitted forget requests but still have sensitive items in their top k predictions. To compare a retrained model θ_r and an unlearned model θ_u with respect to this metric, we define

$$\text{RelItems@}k := \text{Sensitive}_r\text{@}k - \text{Sensitive}_u\text{@}k \in [-1, 1]$$

  as the delta between the sensitive item percentage of the retrained model ($\text{Sensitive}_r@k$) and the unlearned model ($\text{Sensitive}_u@k$). $\text{RelItems@}k = p \geq 0$ implies that the unlearned model successfully removes sensitive items from the top k recommendations for $p\,|U_s|$ more users compared to the retrained model. Analogously, $\text{RelItems@}k < 0$ indicates that the unlearned model performs worse than the retrained model.
- *Removal of poisonous data*: This scenario evolves around maliciously injected interactions in session-based or next-basket recommendation. An actor injects clicks on random items and items they want to promote, biasing the model toward these items. Successful unlearning restores model performance by removing the influence of the fake interactions, measured using standard performance metrics. To compare the performance of a retrained model θ_r and an unlearned model θ_u, we define

$$\text{RelEff@}k := \frac{\text{nDCG}_u@k}{\text{nDCG}_r@k} - 1 \in [-1, \infty).$$

  $\text{RelEff@}k = p\%$ implies that the unlearned model's nDCG@k is $p\%$ better than the retrained models nDCG@k. A model can achieve strong RelEff@k performance while still retaining information from interactions that should have been removed. This may occur, for example, when the original model is underfitted and the repair phase effectively acts as additional training. However, when the trained model is already near-optimal, this concern is negligible. In such settings, any observed performance gains are more plausibly attributed to the removal of noisy or spam interactions, as the model is assumed to have already captured the relevant structure of the retained data.

Unlearning efficiency. We measure unlearning latency, i.e., the time it takes to execute unlearning requests, both in absolute terms and in comparison to full retraining.

Experimentation protocol for ERASE. A single experiment using ERASE combines a recommendation task, a dataset D, a training algorithm, an unlearning algorithm U, and a scenario (e.g., removal of poisonous data or unlearning sensitive item interactions). In each experiment, we sequentially unlearn a sequence of batches $D_f^{(0)}, \ldots, D_f^{(n-1)}$. Let $\theta^{(0)}$ denote the model trained on the full data D, then we sequentially compute $\theta^{(i+1)} \leftarrow U(\theta^{(i)}, D_r^{(i)}, D_f^{(i)})$ where $D_r^{(i)} = D \setminus \bigcup_{j=0}^{i} D_f^{(j)}$. The number of batches n and the data contained in each batch depend on the scenario. For unlearning interactions with sensitive items, we sequentially unlearn by user, e.g., each batch consists of a single user's sensitive interactions to forget. Such small batches are natural, as there is an urgency to carry out the unlearning requests fast in this scenario due to ethical concerns. We choose the number of batches to account for 0.01% of the interactions in total. For the removal of poisonous data it is natural to unlearn larger batches and an overall larger fraction of the dataset, as spammers typically use various accounts to pollute the data [28]. Furthermore, they must inject a large volume of interactions to meaningfully influence a recommender system, and there are no privacy constraints that require companies to perform unlearning under strict latency or bounded removal-time guarantees. Thus, we add spam interactions with a size of 1% of the training set for poisoning, and make each batch to unlearn contain the data of 256 spam users. Hyperparameters were tuned on RSC15

Table 2: Unlearning effectiveness for the best model per scenario-task-dataset combination. Cells with "n/a" indicate that an unlearning algorithm is not applicable to a particular model while "div." indicates that the unlearning diverged. The effectiveness varies substantially, but for most combinations at least one algorithm matches the retrained model.

		Unlearning Spam Interactions (**RelEff@20** ↑)				*Unlearning Interactions with Sensitive Items* (**RelItems@20** ↑)					
		SBR		**NBR**		**CF**		**NBR**		**SBR**	
Algorithm	**Type**	NowP NARM	RSC15 SR-GNN	TaFeng DNNTSP	Dunnhumby Sets2Sets	Food SGL	MovieLens BPR	Instacart DNNTSP	Dunnhumby Sets2Sets	Books SR-GNN	30music SR-GNN
SCIF	RecSys	-4.63	+0.02	-1.16	+5.14	+4.29	-0.92	-45.01	+0.00	-5.19	-9.05
GIF	GNN	n/a	div.	-3.59	n/a	+18.33	n/a	div.	n/a	div.	div.
CEU	GNN	n/a	-50.78	-0.66	n/a	+3.70	n/a	-38.86	n/a	-21.84	-2.26
IDEA	GNN	n/a	div.	-0.69	n/a	div.	n/a	-52.10	n/a	+14.17	div.
Fanchuan	General	-16.99	-22.82	-7.07	+9.85	+0.32	-1.43	-37.02	-0.25	-19.30	-3.54
Kookmin	General	-34.72	-20.54	-2.06	+4.90	+2.06	+0.06	-37.40	-0.18	+5.17	-13.42
Seif	General	-23.34	-99.91	-11.61	+2.74	-2.64	+0.07	-22.71	-6.32	-2.27	-9.56

and Food or adopted from best-performing configurations reported in prior repositories. A complete overview of the hyperparameters is provided in our supplementary material.[1]

ERASE implementation and resources. We implement ERASE on top of the RecBole library [61] and release it under an open license at https://github.com/deem-data/erase-bench/. Following the protocol described above, ERASE supports follow-up research with flexible experimentation across its models, datasets, and unlearning settings. Beyond the benchmark code, we release the full results of running ERASE at scale. We execute training, retraining, and unlearning runs over five random seeds, consuming more than 13,000 GPU hours and producing over 600 GB of reusable artifacts, including 1,069 model checkpoints. By providing pretrained and retrained checkpoints, we offload the most computationally expensive steps and enable the community to evaluate new unlearning algorithms using only additional unlearning runs. Specifically, we release (i) training logs, (ii) 150 trained model checkpoints, (iii) 150 retrained model checkpoints, (iv) 769 unlearned model checkpoints, and (v) corresponding efficiency, effectiveness, and utility metrics. We provide pointers to our artifacts,[2] with additional experimental details and results in our online supplementary material.[1]

5 Example Use of ERASE

We showcase how ERASE can be used to gain insights and directions to inform progress in MU for recommender systems.

A snapshot of the current progress in MU for recommender systems. We analyze the experimental logs of ERASE and revisit the guiding questions outlined in Section 4. Our goal is to assess not only whether existing MU methods are effective, but also under which conditions they succeed or fail. While we summarize the main findings below, we refer to our supplementary material[1] for a detailed breakdown and extended discussion of the observed trends.

Unlearning effectiveness. To assess the unlearning effectiveness of the algorithms in ERASE, we aggregate the experimental logs in a structured manner. For each scenario–task–dataset combination, we first identify the best-performing original model based on nDCG@20. We then compute its mean RelEff@20 in the case of spam unlearning, or mean RelItems@20 in the case of sensitive interaction unlearning. This aggregation ensures that we evaluate unlearning quality under realistic conditions where practitioners would deploy the strongest available model. The resulting scores in Table 2 indicate that the effectiveness of approximate unlearning algorithms varies substantially across datasets. In most experiments, at least one approximate method matches the retrained model in both unlearning effectiveness and recommendation utility.

Robustness. To quantify the robustness of unlearning algorithms, we collect the relative effectiveness scores (RelEff@20 or RelItems@20 depending on the scenario) per algorithm for all datasets and models of a given scenario-task combination from the experimental logs of ERASE, and plot their distributions in Figure 2. No unlearning algorithm is uniformly robust across all CF, SBR, and NBR tasks; many methods show instability or performance degradation under repeated unlearning. The recommender-specific method SCIF stands out as the most consistent and robust approach across tasks, exhibiting low variance in unlearning effectiveness, with remaining challenges primarily in sensitive item unlearning for NBR.

Influence of model architecture. We investigate the experimental logs of ERASE for different models, and find that model architecture has a strong impact: the effectiveness and utility of general unlearning algorithms often degrade under repeated unlearning requests, particularly for attention-based and recurrent models. Architecture- and recommender-specific methods—most notably SCIF, and in some cases GNN-specific algorithms—consistently achieve higher effectiveness and utility. Table 2 and Figure 3 therefore report results for the best-performing original model for each (unlearning scenario, recommendation task) combination, as these models are the most practically relevant. Comprehensive results across the full product space are provided in the supplementary material.[1]

Efficiency and deployability. For practical deployment, an unlearning algorithm must update a model substantially faster than full retraining while maintaining comparable effectiveness. To assess deployability, we extract from the ERASE logs the mean unlearning runtime and relative effectiveness of each algorithm for the best model in each scenario–task–dataset combination. Next, we plot the resulting speedup over retraining (retraining time divided by unlearning time) against relative effectiveness in Figure 3. Based on our interviews with practitioners from industry, where retraining can take up to a day, we consider a three-orders-of-magnitude runtime reduction—bringing unlearning down to minutes—sufficient for deployment, provided effectiveness remains comparable (e.g.,

[1] https://github.com/deem-data/erase-bench/blob/main/supplementary-material.pdf
[2] https://github.com/deem-data/erase-bench/blob/main/artifacts.md

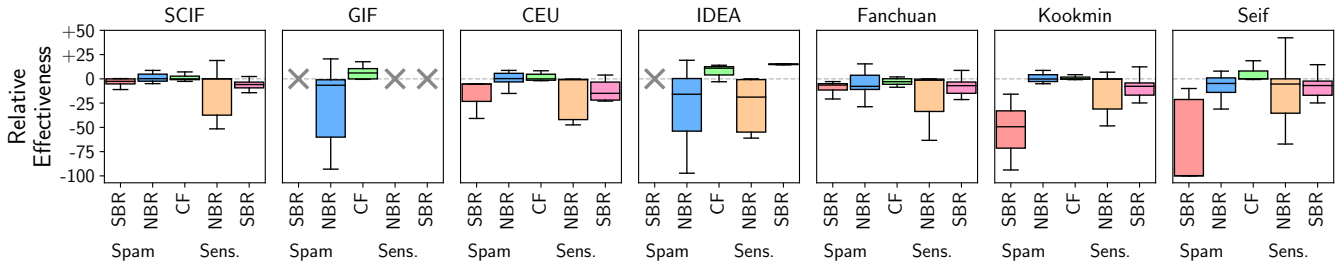


Figure 2: Robustness of unlearning algorithms. We plot the distribution of relative effectiveness aggregated over datasets, models, and runs. The recommender-specific algorithm SCIF is the most consistent and robust approach across tasks. Algorithms with no data for a scenario-task combination are either not applicable to the respective models or diverge while unlearning.

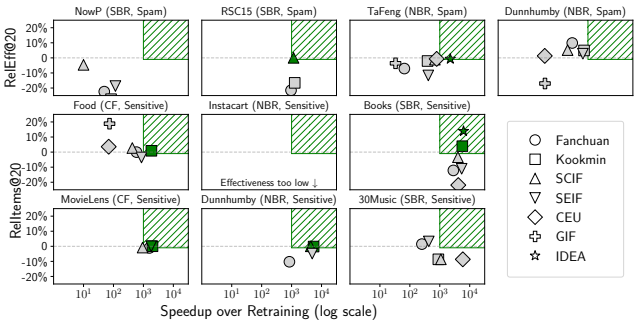


Figure 3: Speedup over retraining against relative effectiveness for unlearning algorithms in ERASE. The green area marks algorithms attractive for deployment.

within a 1% margin). We indicate this target region in green as a "deployability area" in the figure and highlight methods that meet this criterion. Overall, the results show that most current unlearning algorithms fall short of real-world deployment requirements due to either excessive latency or insufficient effectiveness.

Directions for follow-up research. ERASE uncovers new research directions for MU in recommender systems. Since efficient, effective, and high-utility unlearning is possible in most scenarios, unlearning may be viewed as an algorithm selection problem. This selection should account for algorithms that degrade sharply after a small number of requests. Stabilizing unlearning performance is another key challenge. Since unlearning is computationally cheap, ensembling differently seeded instances or various algorithms in parallel could effectively reduce variance.

Finally, as illustrated in Figure 3, unlearning latency must be further reduced for practical deployment, due to the lack of deployable methods for several scenarios. There has been progress on low-latency MU for neighborhood-based models [40, 41], but how to accelerate approximate MU for neural network-based recommender systems is an open question. ERASE constitutes an experimental testbed for all these research directions: newly proposed methods can directly be evaluated on our datasets, scenarios, and model checkpoints.

6 Resource Reproducibility & Extensibility

Our repository contains pointers to the dataset locations and to downloadable checkpoints of trained, retrained, and unlearned models hosted on Hugging Face.

Re-running the benchmark. Our benchmark can be re-run via two dedicated scripts to execute and evaluate (re)training and unlearning, see README.md for instructions. Both scripts offer a wide range of configuration options and automatically save log files and model checkpoints. The benchmark can also use private data; no external APIs are required. To provide reproducibility, we give the option to fix random seeds, which we also use for our experiments, and log all relevant hyperparameters, enabling consistent comparisons across runs.

In the future, we will maintain ERASE and welcome contributions. Our codebase has predefined extension points for new datasets, models, unlearning algorithms, and scenarios, as documented in README.md. New models have to extend the standard RecBole model class; new unlearning methods have to extend our unlearning script and a corresponding pipeline. A new scenario requires a generator to produce forget-set artifacts for a given dataset.

Extending the benchmark. We will maintain ERASE and welcome contributions. Our codebase has predefined extension points for new datasets, models, unlearning algorithms, and scenarios. See README.md for detailed guidelines on each extension point.

7 Conclusion

We have introduced ERASE, a real-world aligned benchmark for machine unlearning in recommender systems that addresses key gaps in prior benchmarks and is easy to extend. It provides a large collection of reusable artifacts, including pretrained and retrained model checkpoints, allowing researchers to avoid costly retraining and to evaluate new unlearning algorithms with only additional unlearning runs. Together, these features position ERASE as a common testbed to systematically assess, advance, and track practical machine unlearning in recommender systems.

Acknowledgments

This research was (partially) supported by the Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.-183, and KICH3.LTP.20.006, and the European Union under grant agreement No. 101201510 (UNITE). The authors acknowledge the Scientific Computing of the IT Division at the Charité - Universitätsmedizin Berlin for providing computational resources that have contributed to the research results reported in this paper. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of their respective employers, funders and/or granting authorities.

References

[1] Amazon Web Services. 2025. *Maintaining domain recommenders.* https://docs.aws.amazon.com/personalize/latest/dg/maintaining-relevance.html#maintaining-domain-recommenders Accessed: 2026-02-12.

[2] Buzzfeed. 2019. "Amazon's Choice" Does Not Necessarily Mean A Product Is Good. https://www.buzzfeednews.com/article/nicolenguyen/amazons-choice-bad-products

[3] California Privacy Protection Agency. 2025. California Consumer Privacy Act – Frequently Asked Questions. https://cppa.ca.gov/faq.html. Accessed: 2026-02-12.

[4] Yinzhi Cao and Junfeng Yang. 2015. Towards Making Systems Forget with Machine Unlearning. In *2015 IEEE Symposium on Security and Privacy*. 463–480. doi:10.1109/SP.2015.35

[5] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *28th USENIX security symposium (USENIX security 19)*. 267–284.

[6] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Ùlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting Training Data from Large Language Models. In *30th USENIX Security Symposium (USENIX Security 2021)*. USENIX Association, 2633–2650. https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting

[7] Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. 2022. Recommendation Unlearning. In *Proceedings of the ACM Web Conference 2022*. 2768–2777.

[8] Chaochao Chen, Jiaming Zhang, Yizhao Zhang, Li Zhang, Lingjuan Lyu, Yuyuan Li, Biao Gong, and Chenggang Yan. 2024. CURE4Rec: A Benchmark for Recommendation Unlearning with Deeper influence. In *NeurIPS*.

[9] Zihao Chen, Chenyang Zhang, Chen Xu, Zhao Zhang, Jiaqiang Wang, Weining Qian, and Aoying Zhou. 2025. Scheduling Data Processing Pipelines for Incremental Training on MLP-based Recommendation Models. In *Companion of the 2025 International Conference on Management of Data* (Berlin, Germany) *(SIGMOD/PODS '25)*. Association for Computing Machinery, New York, NY, USA, 350–363. doi:10.1145/3722212.3724454

[10] Chiranjiv Das. 2025. Ta-Feng Grocery Dataset. https://www.kaggle.com/datasets/chiranjivdas09/ta-feng-grocery-dataset. Accessed: 2026-02-12.

[11] Yushun Dong, Binchi Zhang, Zhenyu Lei, Na Zou, and Jundong Li. 2024. IDEA: A Flexible Framework of Certified Unlearning for Graph Neural Networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. Association for Computing Machinery, New York, NY, USA, 621–630. doi:10.1145/3637528.3671744

[12] dunnhumby. 2025. Source Files Datasets. https://www.dunnhumby.com/source-files/. Accessed: 2026-02-12.

[13] Guglielmo Faggioli, Mirko Polato, and Fabio Aiolli. 2020. Recency Aware Collaborative Filtering for Next Basket Recommendation. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization* (Genoa, Italy) *(UMAP '20)*. Association for Computing Machinery, New York, NY, USA, 80–87. doi:10.1145/3340631.3394850

[14] GDPR.eu. 2016. Article 17: Right to be forgotten. https://gdpr.eu/article-17-right-to-be-forgotten. Accessed: 2026-02-12.

[15] Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. 2019. Making AI Forget You: Data Deletion in Machine Learning. arXiv:1907.05012 [cs.LG] https://arxiv.org/abs/1907.05012

[16] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4 (2015), 19:1–19:19. doi:10.1145/2827872

[17] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 639–648. doi:10.1145/3397271.3401063

[18] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. arXiv:1511.06939 [cs.LG] https://arxiv.org/abs/1511.06939

[19] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging Language and Items for Retrieval and Recommendation. *arXiv preprint arXiv:2403.03952* (2024).

[20] Haoji Hu and Xiangnan He. 2019. Sets2Sets: Learning from Sequential Sets with Neural Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 1491–1499. doi:10.1145/3292500.3330979

[21] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling Personalized Item Frequency Information for Next-basket Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1071–1080. doi:10.1145/3397271.3401066

[22] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. 2021. Approximate Data Deletion from Machine Learning Models. arXiv:2002.10077 [cs.LG] https://arxiv.org/abs/2002.10077

[23] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) *(RecSys '17)*. Association for Computing Machinery, New York, NY, USA, 306–310. doi:10.1145/3109859.3109872

[24] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2015. The Composition Theorem for Differential Privacy. arXiv:1311.0776 [cs.DS] https://arxiv.org/abs/1311.0776

[25] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. 197–206. doi:10.1109/ICDM.2018.00035

[26] Barrie Kersbergen, Olivier Sprangers, and Sebastian Schelter. 2022. Serenade-Low-latency Session-based Recommendation in E-commerce at Scale. In *Proceedings of the 2022 International Conference on Management of Data*. 150–159.

[27] Khanh Nam Le. 2025. Instacart Market Basket Analysis Dataset. https://github.com/khanhnamle1994/instacart-orders/tree/master/data. Accessed: 2026-02-12.

[28] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. 2016. Data poisoning attacks on factorization-based collaborative filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) *(NIPS'16)*. Curran Associates Inc., Red Hook, NY, USA, 1893–1901.

[29] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Singapore, Singapore) *(CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 1419–1428. doi:10.1145/3132847.3132926

[30] Xiang Li, Wenqi Wei, and Bhavani Thuraisingham. 2025. MUBox: A Critical Evaluation Framework of Deep Machine Unlearning [Systematization of Knowledge Paper]. In *Proceedings of the 30th ACM Symposium on Access Control Models and Technologies* (USA) *(SACMAT '25)*. Association for Computing Machinery, New York, NY, USA, 175–188. doi:10.1145/3734436.3734454

[31] Yuyuan Li, Chaochao Chen, Yizhao Zhang, Weiming Liu, Lingjuan Lyu, Xiaolin Zheng, Dan Meng, and Jun Wang. 2023. UltraRE: Enhancing RecEraser for Recommendation Unlearning via Error Decomposition. *Advances in Neural Information Processing Systems* 36 (2023), 12611–12625.

[32] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Yizhao Zhang, Biao Gong, and Jun Wang. 2023. Selective and Collaborative Influence Function for Efficient Recommendation Unlearning. arXiv:2304.10199 [cs.IR] https://arxiv.org/abs/2304.10199

[33] Pierre Lubitzsch, Olga Ovcharenko, Hao Chen, Maarten de Rijke, and Sebastian Schelter. 2025. Towards a Real-World Aligned Benchmark for Unlearning in Recommender Systems. *FAccTRec Workshop: Responsible Recommendation at the ACM Conference on Recommender Systems* (2025).

[34] Bill Marino, Meghdad Kurmanji, and Nicholas D. Lane. 2025. Position: Bridge the Gaps between Machine Unlearning and AI Regulation. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems Position Paper Track*. https://openreview.net/forum?id=0ngi2StMwC

[35] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. The World is Binary: Contrastive Learning for Denoising Next Basket Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 859–868. doi:10.1145/3404835.3462836

[36] Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. 2023. Disentangled Contrastive Collaborative Filtering. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1137–1146. doi:10.1145/3539618.3591665

[37] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[38] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation. In *Proceedings of the 19th International Conference on World Wide Web* (Raleigh, North Carolina, USA) *(WWW '10)*. Association for Computing Machinery, New York, NY, USA, 811–820. doi:10.1145/1772690.1772773

[39] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web* (Hong Kong, Hong Kong) *(WWW '01)*. Association for Computing Machinery, New York, NY, USA, 285–295. doi:10.1145/371920.372071

[40] Sebastian Schelter, Mozhdeh Ariannezhad, and Maarten de Rijke. 2023. Forget Me Now: Fast and Exact Unlearning in Neighborhood-based Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2011–2015.

[41] Sebastian Schelter, Stefan Grafberger, and Maarten de Rijke. 2024. Snapcase-Regain Control over Your Predictions with Low-Latency Machine Unlearning. *Proceedings of the VLDB Endowment* 17, 12 (2024), 4273–4276.

[42] Sebastian Schelter, Stefan Grafberger, and Ted Dunning. 2021. HedgeCut: Maintaining Randomised Trees for Low-Latency Machine Unlearning. In *Proceedings of the 2021 International Conference on Management of Data* (Virtual Event, China) *(SIGMOD '21)*. Association for Computing Machinery, New York, NY, USA, 1545–1557. doi:10.1145/3448016.3457239

[43] Sebastian Schelter and Julia Stoyanovich. 2020. Taming technical bias in machine learning pipelines. *Bulletin of the Technical Committee on Data Engineering* 43, 4 (2020).

[44] Julia Stoyanovich, Serge Abiteboul, Bill Howe, HV Jagadish, and Sebastian Schelter. 2022. Responsible Data Management. *Commun. ACM* 65, 6 (2022), 64–74.

[45] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (Boston, MA, USA) *(DLRS 2016)*. Association for Computing Machinery, New York, NY, USA, 17–22. doi:10.1145/2988450.2988452

[46] Helen Toner, Rogier Creemers, and Graham Webster. 2022. Internet Information Service Algorithmic Recommendation Management Provisions. https://digichina.stanford.edu/work/translation-internet-information-servicealgorithmic-recommendation-management-provisions-opinon-seeking-draft/. Accessed: 2026-02-12.

[47] Eleni Triantafillou, Peter Kairouz, Fabian Pedregosa, Jamie Hayes, Meghdad Kurmanji, Kairan Zhao, Vincent Dumoulin, Julio Jacques Junior, Ioannis Mitliagkas, Jun Wan, Lisheng Sun Hosoya, Sergio Escalera, Gintare Karolina Dziugaite, Peter Triantafillou, and Isabelle Guyon. 2024. Are We Making Progress in Unlearning? Findings from the First NeurIPS Unlearning Competition. arXiv:2406.09073 [cs.LG] https://arxiv.org/abs/2406.09073

[48] Roberto Turrin, Massimo Quadrana, Andrea Condorelli, Roberto Pagano, and Paolo Cremonesi. 2015. 30Music Listening and Playlists Dataset. In *Proceedings of the 9th ACM Conference on Recommender Systems (RecSys 2015), Poster Proceedings (CEUR Workshop Proceedings, Vol. 1441)*. Vienna, Austria. https://ceur-ws.org/Vol-1441/recsys2015_poster13.pdf

[49] Benjamin Longxiang Wang and Sebastian Schelter. 2022. Efficiently Maintaining Next Basket Recommendations under Additions and Deletions of Baskets and Items. arXiv:2201.13313 [cs.IR] https://arxiv.org/abs/2201.13313

[50] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 726–735. doi:10.1145/3404835.3462862

[51] Jiancan Wu, Yi Yang, Yuchun Qian, Yongduo Sui, Xiang Wang, and Xiangnan He. 2023. GIF: A General Graph Unlearning Strategy via Influence Function. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. ACM, 651–661. doi:10.1145/3543507.3583521

[52] Kun Wu, Jie Shen, Yue Ning, Ting Wang, and Wendy Hui Wang. 2023. Certified Edge Unlearning for Graph Neural Networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) *(KDD '23)*. Association for Computing Machinery, New York, NY, USA, 2606–2617. doi:10.1145/3580305.3599271

[53] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 346–353. doi:10.1609/aaai.v33i01.3301346

[54] Yinjun Wu, Edgar Dobriban, and Susan B. Davidson. 2020. DeltaGrad: Rapid retraining of machine learning models. arXiv:2006.14755 [cs.LG] https://arxiv.org/abs/2006.14755

[55] Yinjun Wu, Val Tannen, and Susan B. Davidson. 2020. PrIU: A Provenance-Based Approach for Incrementally Updating Regression Models. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD/PODS '20)*. ACM, 447–462. doi:10.1145/3318464.3380571

[56] Lianghao Xia, Chao Huang, Jiao Shi, and Yong Xu. 2023. Graph-less Collaborative Filtering. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. ACM, 17–27. doi:10.1145/3543507.3583196

[57] Yuanshun Yao, Xiaojun Xu, and YangLiu. 2024. Large Language Model Unlearning. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 105425–105475. doi:10.52202/079017-3346

[58] Le Yu, Leilei Sun, Bowen Du, Chuanren Liu, Hui Xiong, and Weifeng Lv. 2020. Predicting Temporal Sets with Deep Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. ACM, 1083–1091. doi:10.1145/3394486.3403152

[59] Eva Zangerle, Martin Pichl, Wolfgang Gassler, and Günther Specht. 2014. #nowplaying Music Dataset: Extracting Listening Behavior from Twitter. In *Proceedings of the First International Workshop on Internet-Scale Multimedia Management* (Orlando, Florida, USA) *(WISMM '14)*. Association for Computing Machinery, New York, NY, USA, 21–26. doi:10.1145/2661714.2661719

[60] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* 64, 3 (Feb. 2021), 107–115. doi:10.1145/3446776

[61] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. arXiv:2011.01731 [cs.IR] https://arxiv.org/abs/2011.01731