LightUL: An Efficient Recommendation Unlearning Framework

Wentao Ning
The University of Hong Kong
Southern University of Science and Technology
nwt9981@connect.hku.hk

Haorui He
The University of Hong Kong
Hong Kong Baptist University
hehaorui11@gmail.com

Reynold Cheng*
The University of Hong Kong
ckcheng@cs.hku.hk

Nur Al Hasan Haldar
Curtin University
nur.haldar@curtin.edu.au

Ben Kao
The University of Hong Kong
kao@cs.hku.hk

Nan Huo
The University of Hong Kong
huonan@connect.hku.hk

Bo Tang†
Southern University of Science and Technology
tangb3@sustech.edu.cn

Yupeng Li
Hong Kong Baptist University
ivanypli@gmail.com

ABSTRACT

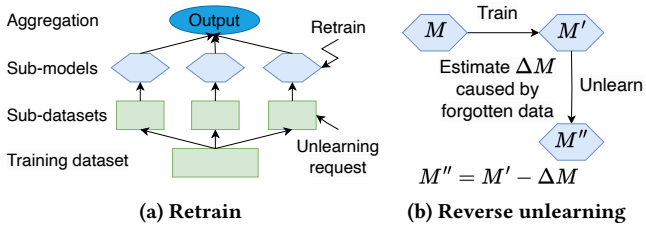
Recommendation unlearning erases the influence of specific data from a well-trained recommender system. However, existing unlearning approaches either require costly retraining or significantly reduce recommendation accuracy. In response, this work introduces a system-agnostic, lightweight unlearning framework, LightUL, which proposes a whitening module to enable efficient unlearning by exclusively training a small MLP on the data to be unlearned and "phantom users" to anonymize interaction data to preserve collaborative information, thereby alleviating performance degradation. Experimental results show that LightUL outperforms existing solutions in both effectiveness and efficiency.

VLDB Workshop Reference Format:
Wentao Ning, Haorui He, Reynold Cheng, Nur Al Hasan Haldar, Ben Kao, Nan Huo, Bo Tang, and Yupeng Li. LightUL: An Efficient Recommendation Unlearning Framework. VLDB 2025 Workshop: GuideAI.2025.

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/Stevenn9981/LightUL.

1 INTRODUCTION

Recommender systems [5, 12, 13] leverage vast amounts of user interaction data (e.g., clicks, purchases) to provide personalized recommendations. However, such data often contains sensitive or personal information. Recent regulations, such as the General Data Protection Regulation (GDPR) [3], mandate that service providers comply with user requests to erase their data. This requires not only deleting such data, referred to as *forgotten set*, from datasets but also ensuring that the trained systems to "forget" them. This process is known as *Recommendation Unlearning* [7, 14, 17].



(a) Retrain **(b) Reverse unlearning**

Figure 1: Overview of the retrain and reverse unlearning. Here, M, $M^{'}$, $M^{''}$ and ΔM are model parameters.

As shown in Fig. 1, existing solutions for recommendation unlearning fall into two main groups[8]: retrain and/or reverse unlearning. Retrain unlearning [1, 18] is a straightforward yet costly approach that involves retraining the system from scratch after deleting the forgotten set, which is impractical for real-world application involving large-scale train sets and frequent unlearning requests. Though some methods[1, 8] propose to partition the data to train multiple sub-models ahead, allowing retraining to be limited to the affected sub-models, this approach is only effective when the forgotten set is not distributed across numerous partitions, which is a condition that rarely holds in practice. Reverse unlearning methods [9, 20] estimate the impact of forgotten data on model parameters using reverse gradient operations. However, their reliance on localized Hessian approximations limits their ability to capture the global influence of graph convolutions, resulting in suboptimal unlearning efficacy and recommendation accuracy.

To address these limitations, we propose a system-agnostic lightweight framework, LightUL, which can be plugged into any recommender system (called *base model*) for unlearning. The framework consists of two key designs, i.e., the *whitening module* and *phantom users*. Firstly, the whitening module fine-tunes the user/item embeddings by a frozen base model with a simple Multi-Layer Perception (MLP), which is trained to aligning the representation of the forgotten interactions with those of non-existing interactions.

*Reynold Cheng is a corresponding author.
†Bo Tang is a corresponding author.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.

This method achieves efficient unlearning by exclusively training a small MLP on the forgotten set and an equal-sized set of randomly selected samples, which is typically much smaller than the entire train set. Secondly, to alleviate accuracy decline, we propose an anonymization strategy (detailed in Sec. 3.2) that creates non-existent "phantom users," to preserve anonymized collaborative information in the forgotten set for maintaining accuracy. This strategy is unidirectional (i.e., phantom users cannot be traced back to the original users), complying with regulations, e.g., GDPR [3], that permit the preservation of anonymized data. Experimental results demonstrate that our proposed method outperforms state-of-the-art unlearning methods in both effectiveness and efficiency.

2 PRELIMINARIES

This section introduces the preliminaries relevant to our approach.

Recommender systems: Let $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items, respectively. The set $\mathcal{R} \subset \mathcal{U} \times \mathcal{I}$ represents the user-item interaction records (e.g., clicks, purchases). Recommender systems are trained on $\mathcal{R}$ to predict the likelihood of a user $u \in \mathcal{U}$ interacting with an item $i \in \mathcal{I}$. Specifically, these systems assign a score to each user-item pair, where a higher score indicates a greater probability of interaction. The top-ranked items are then recommended to each user. Typically, the systems learn user and item embeddings from interactions and aggregate them (e.g., via inner product or neural networks) to generate prediction scores.

Recommendation unlearning: Let $\mathcal{R}_f$ denote the set of interactions to be unlearned (*forgotten set*), and $\mathcal{R}_r = \mathcal{R} \setminus \mathcal{R}_f$ represent the set of interactions to be retained (*retained set*). Recommendation unlearning aims to erase the influence of $\mathcal{R}_f$ from a system trained on $\mathcal{R}$. Effective unlearning should achieve three objectives: (1) completely eliminate the influence of the forgotten set, (2) perform unlearning as quickly as possible, and (3) maintain high recommendation accuracy after unlearning.

3 METHODOLOGY

To achieve the objectives outlined above, this section introduces our proposed unlearning framework, LightUL, as depicted in Fig. 2.

3.1 Whitening module

To effectively unlearn the forgotten set, we design a system-agnostic whitening module that can be plugged into existing recommender systems, which fine-tunes the user or item embeddings that affect recommendation results.

Design principles: In designing the whitening module, we propose and follow the three principles: (1) The embeddings of *unaffected* users and items should remain unchanged before and after the unlearning process to preserve accurate information. (2) The predictions for forgotten interactions should exhibit significant differences before and after unlearning. (3) The predictions of forgotten interactions after unlearning should be similar to those of non-existing interactions.

Module design: As shown in Fig. 2, a whitening layer is introduced after the frozen base model, which encodes users and items into embeddings. The purpose of this whitening layer is to adjust the user and item embeddings to achieve unlearned recommendations. Specifically, for a user or item v with embedding $\boldsymbol{e}_v \in \mathbb{R}^d$, the whitening layer WL is defined as:

$$\boldsymbol{e}'_v = \mathrm{WL}(\boldsymbol{e}_v) = \begin{cases} \Omega(\boldsymbol{e}_v) & \text{if } v \text{ is affected,} \\ \boldsymbol{e}_v & \text{otherwise,} \end{cases} \tag{1}$$

where v is considered affected if it is involved in at least one forgotten interaction. Here, $\Omega : \mathbb{R}^d \to \mathbb{R}^d$ is a differentiable function, and $\boldsymbol{e}_v$ remains unchanged for unaffected users or items. This design ensures that the whitening layer exclusively influences the embeddings of users or items involved in forgotten interactions, aligning with the first design principle.

Training: To train the whitening module, we design two loss functions to train the whitening layer, guided by the second and third design principles. First, we compute the system predictions for forgotten interactions before and after unlearning, and then increase their distance using a contrastive learning loss: $L_1 = \sum_{(u,i)\in\mathcal{R}_f} -\ln \sigma(\mathrm{AGG}(\boldsymbol{e}_u, \boldsymbol{e}_i) - \mathrm{AGG}(\boldsymbol{e}'_u, \boldsymbol{e}'_i))$, where $\boldsymbol{e}_u$ and $\boldsymbol{e}'_u$ denote the embedding of user u before and after the whitening layer WL, respectively. σ is the sigmoid function that maps a score into $(0, 1)$ range. AGG represents the aggregation function used in the base model (e.g., inner product, neural networks). In this work, we instantiate our LightUL with two representative recommender systems: BPRMF [13] and LightGCN [5]. Both of their aggregation functions are vector inner products.

Besides, we make the prediction scores of the forgotten interactions close to those of non-existing interactions as follows.

$$L_2 = \frac{1}{|O|} \sum_{(u,i^+,i^-)\in O} (\mathrm{AGG}(\boldsymbol{e}'_u, \boldsymbol{e}'_{i^+}) - \mathrm{AGG}(\boldsymbol{e}'_u, \boldsymbol{e}'_{i^-}))^2,$$

where $O = \{(u, i^+, i^-) | (u, i^+) \in \mathcal{R}_f, (u, i^-) \in \mathcal{R}^-\}$ and (u, i^+, i^-) is a training sample for this loss function. Here, $\mathcal{R}^-$ contains randomly sampled user-item pairs that do not exist in $\mathcal{R}$; i^- is a randomly sampled negative sample that u has not interacted before. Finally, the loss function is a combination of L_1 and L_2: $L_{WM} = \alpha L_1 + (1-\alpha) L_2$, where α is a hyper-parameter to balance the two loss functions. **The whitening module achieves efficient recommendation unlearning from the following perspectives.** (1) It modifies only the original embeddings generated by the base model, avoiding the need for prohibited retraining of different sub-models or reverse gradient operations, which require significantly more GPU memory. (2) We freeze the base model and implement Ω using a Multi-Layer Perception (MLP), which has significantly fewer parameters than the base model. (3) Training involves only the forgotten set and an equal-sized set of randomly selected samples, eliminating the need to access the substantially larger entire train set.

3.2 Phantom Users

Unlearning interaction data from a recommender system often leads to a decline in accuracy. To address this issue, we introduce "phantom users," which are non-existent users designed to anonymize real users while retaining a portion of the forgotten interactions. By preserving part of the anonymized interaction patterns, phantom users help maintain recommendation accuracy.

Generation strategy: Each real user is assigned to a phantom user, and a single phantom user can be associated with multiple real users to ensure anonymization. Specifically, we first apply k-means clustering [15] on user embeddings derived from the base

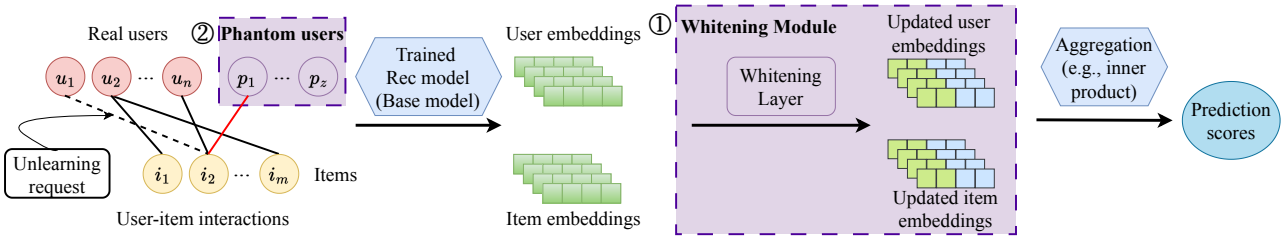


Figure 2: Overview of LightUL framework, which consists of two key designs, i.e., ① whitening module and ② phantom users.

Table 1: Statistical details of datasets.

	ML-1M	Gowalla	Yelp
#User	6,038	29,858	31,668
#Item	3,883	40,981	38,048
#Interaction	1,000,209	5,946,257	8,827,696

model to divide users into clusters. Each cluster corresponds to a phantom user and represents multiple real users, where k is set to $n\%$ of the total number of real users, and $0 < n < 100$ is a hyper-parameter. This strategy ensures that users within each cluster share similar interaction patterns. As illustrated in Fig. 2, when a real user submits an unlearning request, the interactions involving that user are linked to its assigned phantom user, thereby preserving the anonymized collaborative information of the real users.

Training: After creating the phantom users prior to the unlearning process, we randomly initialize their user embeddings and train them while simultaneously updating the related item embeddings during unlearning. We employ the widely used Bayesian Personalized Ranking (BPR) loss [11, 12]:

$$L_{PU} = \sum_{(p,i^+,i^-)\in\mathcal{P}} -\ln\sigma(\mathrm{AGG}(\boldsymbol{e}'_p, \boldsymbol{e}'_{i^+}) - \mathrm{AGG}(\boldsymbol{e}'_p, \boldsymbol{e}'_{i^-})),$$

where (p, i^+, i^-) is a training sample for this loss function and $\mathcal{P}$ denotes the set of them. p is a phantom user and i^+ is an item that has an interaction record associated with p. i^- is a randomly selected negative sample with no interaction associated with p.

Now, we can get the final loss function: $L = L_{WM} + \lambda L_{PU} + \gamma\|\Theta\|_2^2$, where λ is a tunable hyper-parameter that controls the weight of phantom user loss term. Θ denotes all trainable parameters in the model, and γ is the weight of regularization.

4 EXPERIMENTS

We conduct experiments to evaluate our proposed methods.

4.1 Experimental Setups

This subsection details our exerimental setups.

Datasets. We conduct experiments on three datasets, i.e., ML-1M [10], Gowalla [4], and Yelp [19]. Their statistics are in Table 1.

Evaluation methodology. We randomly split the dataset into train/validation/test sets by 8:1:1 ratio. Same as [1, 2, 16], we randomly select 1% interactions from the train set as the forgotten set. To evaluate the recommendation accuracy, we adopt the all-ranking protocol [5, 12] and report NDCG@20 and NDCG@50 [6] for recommendation accuracy. To evaluate unlearning completeness, we follow previous works [2] to construct a test set containing all forgotten interactions and an equal number of randomly sampled retained interactions, labeling forgotten interactions as 0 and retained ones as 1. Unlearning completeness is measured by the Area Under the Curve (AUC) on this set, with higher AUC values indicating better unlearning completeness.

Baselines. We instantiate our LightUL with two representative recommendation systems (BPRMF [13] and LightGCN [5]), and then compare with state-of-the-art unlearning methods:

- **Original**: The trained system before unlearning.
- **Retrain**: This method deletes all the forgotten set and retrains the system from scratch.
- **RecEraser** [1]: This method splits the data and system into shards and only retrains the affected shards.
- **LASER** [8]: This method retrains the system from the latest unaffected intermediate checkpoint.
- **IFRU** [20]: This method employs influence functions with Hessian-Vector Product (HVP) to estimate and mitigate the impact of forgotten data on model parameters.
- **LightUL-WM**: An ablation of LightUL, which only contains the whitening module without using the phantom users.

Implementation details. We tune hyper-parameters carefully using a grid search to achieve optimal performance for all methods. For LASER and RecEraser, we also optimize them for batch unlearning by grouping the data from the same partition together. For IFRU, we adopt the grid search space used in its original paper [20]. For LightUL, we set the learning rate to 0.005, and the training batch size is 512. The embedding dimension size for all users and items is 64. The number of layers in LightGCN is 3. By default, the trade-off coefficient α is set to 0.5, the phantom user loss weight λ is 2, and the regularization weight γ is 1e-4. The phantom user ratio is set to $n\% = 40\%$ by default.

4.2 Recommendation Accuracy

Table 2 presents the accuracy of the systems after unlearning using different methods. We made the following observations: (1) Our methods (LightUL and LightUL-WM) outperform all state-of-the-art baselines in terms of recommendation accuracy. The methods effectively narrow the performance gap with Retrain. In certain cases, such as on the ML-1M and GOWALLA datasets using the LightGCN system, LightUL even outperforms Retrain. The results highlight

Table 2: NDCG@20 and @50 after unlearning.

System	Method	ML-1M		GOWALLA		Yelp	
		@20	@50	@20	@50	@20	@50
BPRMF	Original	0.2260	0.2649	0.0814	0.1105	0.0277	0.0438
	Retrain	0.2255	0.2640	0.0854	0.1141	0.0286	0.0445
	RecEraser	0.0956	0.1224	0.0325	0.0483	0.0133	0.0215
	LASER	0.1542	0.1916	0.0408	0.0500	0.0096	0.0155
	IFRU	0.1870	0.2203	0.0561	0.0813	0.0219	0.0351
	LightUL-WM	0.2196	0.2589	0.0814	**0.1105**	**0.0278**	**0.0439**
	LightUL	**0.2260**	**0.2642**	**0.0815**	**0.1105**	0.0277	0.0438
LightGCN	Original	0.2855	0.3145	0.1355	0.1642	0.0460	0.0658
	Retrain	0.2855	0.3099	0.1354	0.1636	0.0461	0.0657
	RecEraser	0.1002	0.1266	0.1026	0.1260	0.0374	0.0545
	LASER	0.1389	0.1668	0.0635	0.0761	0.0205	0.0295
	IFRU	0.1801	0.2172	0.1241	0.1522	0.0394	0.0580
	LightUL-WM	0.2822	0.3062	0.1356	0.1642	0.0437	0.0636
	LightUL	**0.2902**	**0.3186**	**0.1358**	**0.1644**	**0.0449**	**0.0650**

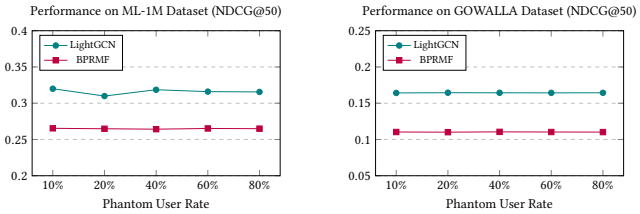


Figure 3: Robustness analysis of phantom user ratio.

the effectiveness of LightUL in maintaining recommendation accuracy. (2) LightUL outperforms the ablation variant LightUL-WM in most scenarios, except for BPRMF on the GOWALLA and Yelp datasets, where LightUL-WM achieves comparable performance. These results indicate that our proposed phantom user strategy can effectively preserve collaborative signals for maintaining recommendation accuracy.

Besides, Fig. 3 shows the NDCG@50 of LightUL using LightGCN and BPRMF across different phantom user ratios on the ML-1M and GOWALLA datasets.[1] The results demonstrate that, although our default setting is 40%, the performance remains robust for phantom user ratios ranging from 10% to 80%. These findings validate the robustness of LightUL to hyper-parameter variations.

4.3 Unlearning Completeness

Table 3 reports the results for unlearning completeness. We make the following observations: (1) The system before unlearning (referred to as Original) achieves an AUC of approximately 0.5 across all datasets and base models, indicating its inability to differentiate between forgotten and retained interactions. In contrast, Retrain demonstrates significantly improved performance compared to Original, confirming the effectiveness of our evaluation methodology in examining a system's ability to distinguish between forgotten and retained interactions. (2) Our methods outperform all of the baselines, demonstrating its superiority in erase the influence of forgotten interactions. However, LightUL performs worse than LightUL-WM, suggesting that using phantom users to retain collaborative information may reduce unlearning completeness.

[1]We omit the results for Yelp dataset for this experiment due to space constraints.

Table 3: Unlearning AUC across different baselines.

Method	BPRMF			LightGCN		
	ML-1M	Gowalla	Yelp	ML-1M	Gowalla	Yelp
Original	0.5045	0.4941	0.5088	0.4898	0.5017	0.5021
Retrain	0.6485	0.7547	0.7286	0.6353	0.7599	0.7613
RecEraser	0.6292	0.6690	0.6491	0.6234	0.7710	0.7712
LASER	0.5910	0.6246	0.6184	0.5716	0.5690	0.5465
IFRU	0.6367	0.8788	0.8274	0.5231	0.5888	0.5715
LightUL-WM	**0.7891**	**0.9283**	**0.8331**	**0.7892**	**0.8456**	**0.9517**
LightUL	0.6891	0.9006	0.8327	0.6654	0.8428	0.8580

Table 4: Running time (minutes) of unlearning.

Method	BPRMF			LightGCN		
	ML-1M	Gowalla	Yelp	ML-1M	Gowalla	Yelp
Retrain	20.51	41.32	49.50	7.39	41.80	166.95
RecEraser	49.53	44.06	42.38	27.15	105.82	203.20
LASER	30.09	28.16	41.02	29.73	64.35	75.20
IFRU	**0.06**	**0.15**	**0.26**	**0.24**	1.52	**2.46**
LightUL-WM	0.80	0.63	0.90	1.75	**1.41**	8.73
LightUL	1.25	0.68	0.95	2.69	2.75	10.68

4.4 Unlearning Efficiency

Table 4 shows the running time (in minutes) for each unlearning method. Notably, IFRU offers the fastest times (0.06-2.46 minutes), while our LightUL maintains competitive efficiency with a modest increase (0.63-10.68 minutes), achieving a 40x speedup over Retrain on Gowalla with BPRMF. This underscores LightUL's strong unlearning efficiency. Moreover, as shown in Sec. 4.3, LightUL excels in unlearning completeness, especially for LightGCN, outperforming IFRU by better mitigating the forgotten data's influence while preserving recommendation quality, due to its ability to handle complex graph structures and global influences that IFRU's localized Hessian approximations struggle to address. In some instances, LASER and RecEraser exceed Retrain's runtime, likely due to distributed unlearning across numerous partitions, necessitating extensive sub-model retraining.

5 CONCLUSIONS

In this paper, we introduced LightUL, a novel system-agnostic framework for efficient recommendation unlearning. The framework incorporates two key components: a whitening module that enables efficient unlearning by focusing exclusively on forgotten set and fine-tuning a small MLP, and phantom users , which preserve collaborative information anonymously to maintain recommendation accuracy. Our experiments proved the efficiency and effectiveness of LightUL. This work offers a practical solution for real-world recommender systems with large-scale datasets and frequent unlearning requests.

REFERENCES

[1] Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. 2022. Recommendation Unlearning. In *WWW*.

[2] Jiali Cheng, George Dasoulas, Huan He, Chirag Agarwal, and Marinka Zitnik. 2023. GNNDelete: A General Strategy for Unlearning in Graph Neural Networks. In *ICLR*.

[3] EU. 2016. General Data Protection Regulation. https://gdpr-info.eu/.

[4] gowalla [n.d.]. Gowalla Dataset. https://snap.stanford.edu/data/loc-gowalla.html.

[5] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. 639–648.

[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. 173–182.

[7] 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. In *NeurIPS*.

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

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

[10] ml1m [n.d.]. Movielens 1M Dataset. https://grouplens.org/datasets/movielens/1m/.

[11] Wentao Ning, Reynold Cheng, Jiajun Shen, Nur Al Hasan Haldar, Ben Kao, Xiao Yan, Nan Huo, Wai Kit Lam, Tian Li, and Bo Tang. 2022. Automatic Meta-Path Discovery for Effective Graph-Based Recommendation. In *CIKM*. 1563–1572.

[12] Wentao Ning, Reynold Cheng, Xiao Yan, Ben Kao, Nan Huo, Nur Al Hasan Haldar, and Bo Tang. 2024. Debiasing Recommendation with Popular Popularity. In *WWW*.

[13] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. *CoRR* abs/1205.2618 (2012).

[14] Hangyu Wang, Jianghao Lin, Bo Chen, Yang Yang, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards Efficient and Effective Unlearning of Large Language Models for Recommendation. *CoRR* abs/2403.03536 (2024).

[15] J Wilpon and L Rabiner. 1985. A modified K-means clustering algorithm for use in isolated work recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33, 3 (1985), 587–594.

[16] Jiancan Wu, Yi Yang, Yuchun Qian, Yongduo Sui, Xiang Wang, and Xiangnan He. 2023. GIF: A General Graph Unlearning Strategy via Influence Function. In *WWW*.

[17] Xin Xin, Liu Yang, Ziqi Zhao, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. 2024. On the Effectiveness of Unlearning in Session-Based Recommendation. In *WSDM*.

[18] Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. 2022. ARCANE: An Efficient Architecture for Exact Machine Unlearning. In *IJCAI*.

[19] yelp [n.d.]. Yelp2018 Dataset. https://www.yelp.com/dataset.

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