A Theoretical Validation of Balanced Division

Supposing there is no prior knowledge regarding the distribution of unlearning requests, it is optimal to presume that users submit requests with equal probability. Under this condition, a balanced division in the ensemble retraining framework can achieve maximum unlearning efficiency.

Proof. Let h_i denote the training overhead of shard S_i, which is in proportion to the shard size, i.e., the number of samples in the shard. Since the users submit unlearning requests with equal probability, the probability of a request located in shard S_i is $P_i = h_i/Z$ where $Z = \sum_j h_j$. Thus, the expectation of retraining overhead when dealing with an unlearning request is

$$\mathbb{E}(H) = \sum_{i=1}^{k} P_i \cdot h_i = \frac{1}{Z}\sum_{i=1}^{k} h_i^2. \tag{6}$$

According to Cauchy-Schwarz Inequality [2], for n random variables x_i, we can get a lower bound of $\sum_i x_i^2$ as:

$$\sum_{i=1}^{n} x_i^2 \geq \frac{(\sum_{i=1}^{n} x_i)^2}{n}. \tag{7}$$

Taking it into (6), we have

$$\mathbb{E}(H) = \frac{1}{Z}\sum_{i=1}^{k} h_i^2 \geq \frac{Z}{k}. \tag{8}$$

We can easily find that setting $h_i = Z/k$ achieves this lower bound, which indicates that a balanced division achieves the maximum unlearning efficiency. □

B Sinkhorn Algorithm

In this section, we provide the details of optimizing the objective of the optimal balanced clustering algorithm. Firstly, we smooth the objective with an entropic regularization term:

$$\min_{\boldsymbol{w}\in\Gamma} \underset{s.t.\ \ \|\boldsymbol{w}\|_1=1, \boldsymbol{w}\succeq 0, \sum_i w_{ij}=\frac{1}{k}, \sum_j w_{ij}=\frac{1}{N}}{\left[\sum_{j=1}^{k}\sum_{i=1}^{N} w_{ij}\cdot\|\boldsymbol{x}_i-\boldsymbol{\mu}_j\|_2^2 + \epsilon\cdot\sum_{j=1}^{k}\sum_{i=1}^{N} w_{ij}\cdot(\log(w_{ij})-1)\right]}. \tag{9}$$

We rewrite Eq (9) with Lagrange multipliers as

$$\begin{aligned}\max_{\boldsymbol{f},\boldsymbol{g}}\min_{\boldsymbol{w}} \mathcal{J} = \Bigg\{ &\sum_{j=1}^{k}\sum_{i=1}^{N} w_{ij}c_{ij} + \epsilon\cdot\sum_{j=1}^{k}\sum_{i=1}^{N} w_{ij}\cdot(\log(w_{ij})-1) - \sum_{j=1}^{k} f_j\left[\left(\sum_{i=1}^{N} w_{ij}\right)-\frac{1}{k}\right] \\ &- \sum_{i=1}^{N} g_i\left[\left(\sum_{j=1}^{k} w_{ij}\right)-\frac{1}{N}\right]\Bigg\},\ \ c_{ij} = \|\boldsymbol{x_i}-\boldsymbol{\mu}_j\|_2^2.\end{aligned} \tag{10}$$

Taking the differentiation w.r.t. w_{ij} on Eq (10), we have

$$\frac{\partial\mathcal{J}}{\partial w_{ij}} = 0 \ \ \Rightarrow\ \ c_{ij} + \epsilon\log(w_{ij}) - f_j - g_i = 0. \tag{11}$$

To update our variables, we first fix g_i and update f_j with

$$f_j^{(t+1)} = \epsilon\left\{\log\left(\frac{1}{k}\right) - \log\left[\sum_{i=1}^{N}\exp\left(\frac{g_i^{(t)}-c_{ij}}{\epsilon}\right)\right]\right\}. \tag{12}$$

Then we fix f_j and update g_i with

$$g_i^{(t+1)} = \epsilon\left\{\log\left(\frac{1}{N}\right) - \log\left[\sum_{j=1}^{k}\exp\left(\frac{f_j^{(t)}-c_{ij}}{\epsilon}\right)\right]\right\}. \tag{13}$$

In summary, we can iteratively update f_j and g_i until we obtain the final solutions.

Table 4: Running time of the learning process on ML-10M.

ML-10M	DMF		LightGCN	
	RecEraser	UltraRE	RecEraser	UltraRE
Stage I	872.53m	259.18s	879.06m	257.83s
Stage II	213.55s	208.44s	860.12s	852.32s
Stage III	83.74s	67.26s	384.50s	376.57s
Total	877.48m	534.88s	899.80m	1,486.72s

C Additional Experiments on the Large-scale dataset

As illustrated in Section 7, current experiments do not sufficiently demonstrate the efficiency enhancement of `UltraRE`, because the majority of time is spent during stage II (independent training) while the enhancements occur in stages I (non-overlapping division) and III (model combination). Thus, following [7], we further conduct experiments on ML-10M (the largest dataset used in [7]) with 50 shards (a large shard number), and report the results in Table 4. From it, we observe that `UltraRE` significantly improves efficiency compared to RecEraser. Specifically, in stages I and III, UltraRE demonstrates average efficiency enhancements of 20,227.87% and 13.30% respectively. Note that, in the large-scale dataset, our proposed clustering algorithm (OBC) outperforms the BKM used in RecEraser, reducing clustering time from several hours to just a few minutes (in stage I, 872.53m/879.06m vs 259.18s/257.83s). The experimental results demonstrate higher efficiency improvements when compared to the results reported in the main text, providing additional evidence of the substantial efficiency enhancements achieved by our method.