Table 7: The best hyperparameters of TALE on five datasets.

Dataset	λ	c	τ_{time}	N
ML-1M	100	0.2	1/512	180
Beauty	100	0.4	1/2	180
Toys	100	0.3	1	360
Sports	500	0.4	4	180
Yelp	0.001	0.2	1/32	180

A Mathematical Proof

A.1 Multi-Target Augmentation

Through the following expansion, we prove that multi-target augmentation learns more items with large position gaps.

$$\hat{\mathbf{B}}_{\text{multi}} = \operatorname*{argmin}_{\mathbf{B}} \sum_{u'=1}^{m'} \|\mathbf{T}^{\text{multi}}_{u',*} - \mathbf{S}_{u',*}\mathbf{B}\|_F^2 \tag{16}$$

$$= \operatorname*{argmin}_{\mathbf{B}} \sum_{u=1}^{m} \sum_{l=1}^{|\mathcal{S}^u|-1} \|\mathbf{x}(i^u_{l+1:|\mathcal{S}^u|}) - \mathbf{x}(i^u_{1:l})\mathbf{B}\|_F^2 \tag{17}$$

$$= \operatorname*{argmin}_{\mathbf{B}} \sum_{u=1}^{m} \sum_{l=1}^{|\mathcal{S}^u|-1} \Big\{ \sum_{s=1}^{l} \Big(\|\mathbf{x}(i^u_{l+1:|\mathcal{S}^u|}) - \mathbf{x}(i^u_s)\mathbf{B}\|_F^2$$

$$- \|\mathbf{x}(i^u_s)\mathbf{B}\|_F^2 \Big) + \|\mathbf{x}(i^u_{1:l})\mathbf{B}\|_F^2 \Big\} \tag{18}$$

$$= \operatorname*{argmin}_{\mathbf{B}} \sum_{u=1}^{m} \sum_{l=1}^{|\mathcal{S}^u|-1} \sum_{s=1}^{l} \|\mathbf{x}(i^u_{l+1:|\mathcal{S}^u|}) - \mathbf{x}(i^u_s)\mathbf{B}\|_F^2 + \tilde{\mathbf{B}}^S_1 \tag{19}$$

$$= \operatorname*{argmin}_{\mathbf{B}} \sum_{u=1}^{m} \sum_{l=1}^{|\mathcal{S}^u|-1} \Big\{ \sum_{s=1}^{l} \Big(\sum_{h=l+1}^{|\mathcal{S}^u|} \|\mathbf{x}(i^u_h) - \mathbf{x}(i^u_s)\mathbf{B}\|_F^2 \Big)$$

$$- (|\mathcal{S}^u| - l - 1)\|\mathbf{x}(i^u_s)\mathbf{B}\|_F^2 \Big\} + \tilde{\mathbf{B}}^S_1 \tag{20}$$

$$= \operatorname*{argmin}_{\mathbf{B}} \sum_{u=1}^{m} \sum_{l=1}^{|\mathcal{S}^u|-1} \sum_{s=1}^{l} \sum_{h=l+1}^{|\mathcal{S}^u|} \|\mathbf{x}(i^u_h) - \mathbf{x}(i^u_s)\mathbf{B}\|_F^2 + \tilde{\mathbf{B}}^S_2 \tag{21}$$

$$= \operatorname*{argmin}_{\mathbf{B}} \sum_{u=1}^{m} \sum_{s=1}^{|\mathcal{S}^u|-1} \sum_{h=s+1}^{|\mathcal{S}^u|} (h-s)\|\mathbf{x}(i^u_h) - \mathbf{x}(i^u_s)\mathbf{B}\|_F^2 + \tilde{\mathbf{B}}^S_2, \tag{22}$$

$$\text{where } \tilde{\mathbf{B}}^S_1 = \sum_{u=1}^{m} \sum_{l=1}^{|\mathcal{S}^u|-1} \|\mathbf{x}(i_{1:l})\mathbf{B}\|_F^2 - \sum_{u=1}^{m} \sum_{l=1}^{|\mathcal{S}^u|-1} \sum_{s=1}^{l} \|\mathbf{x}(i^u_s)\mathbf{B}\|_F^2,$$

$$\tilde{\mathbf{B}}^S_2 = \sum_{u=1}^{m} \sum_{l=1}^{|\mathcal{S}^u|-1} \|\mathbf{x}(i_{1:l})\mathbf{B}\|_F^2 - \sum_{u=1}^{m} \sum_{l=1}^{|\mathcal{S}^u|-1} \sum_{s=1}^{l} (|\mathcal{S}^u| - l)\|\mathbf{x}(i^u_s)\mathbf{B}\|_F^2.$$

Eq. (17) is the objective equation for the optimization problem with multi-target augmentation. For a user u, given items up to l-th in the sequence, it learns to restore items after $(l+1)$-th. To further analyze the learning between the source and target item, we first convert Eq. (17) into Eq. (18) which has only one source item. Then, we transform the equation using the following process.

- **Step1**: $\operatorname{argmin}_{\mathbf{D}} \|\mathbf{C} - (\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D}\|_F^2 = \operatorname{argmin}_{\mathbf{D}} \|\mathbf{C}\|_F^2 + \|(\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D}\|_F^2 - \mathbf{C}^\top(\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D} - ((\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D})^\top\mathbf{C}$.
- **Step2**: $\operatorname{argmin}_{\mathbf{D}} \|\mathbf{C}\|_F^2 + \|(\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D}\|_F^2 - \mathbf{C}^\top(\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D} - ((\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D})^\top\mathbf{C} = \operatorname{argmin}_{\mathbf{D}} \|\mathbf{C}\|_F^2 + \|(\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D}\|_F^2 - \mathbf{C}^\top(\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D} - ((\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D})^\top\mathbf{C} + \|\mathbf{C}\|_F^2$.
- **Step3**: $\operatorname{argmin}_{\mathbf{D}} 2\|\mathbf{C}\|_F^2 + \|(\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D}\|_F^2 - \mathbf{C}^\top(\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D} - ((\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D})^\top\mathbf{C} = \operatorname{argmin}_{\mathbf{D}} \big(\|\mathbf{C}\|_F^2 - \mathbf{C}^\top\mathbf{z}_1\mathbf{D} - \mathbf{D}^\top\mathbf{z}_1^\top\mathbf{C} + \|\mathbf{z}_1\mathbf{D}\|_F^2\big) + \big(\|\mathbf{C}\|_F^2 - \mathbf{C}^\top\mathbf{z}_2\mathbf{D} - \mathbf{D}^\top\mathbf{z}_2^\top\mathbf{C} + \|\mathbf{z}_2\mathbf{D}\|_F^2\big) + \big(\|(\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D}\|_F^2 - \|\mathbf{z}_1\mathbf{D}\|_F^2 - \|\mathbf{z}_2\mathbf{D}\|_F^2\big) = \operatorname{argmin}_{\mathbf{D}} \|\mathbf{C} - \mathbf{z}_1\mathbf{D}\|_F^2 + \|\mathbf{C} - \mathbf{z}_2\mathbf{D}\|_F^2 + \|(\mathbf{z}_1 + \mathbf{z}_2)\mathbf{D}\|_F^2 - \|\mathbf{z}_1\mathbf{D}\|_F^2 - \|\mathbf{z}_2\mathbf{D}\|_F^2$.

Finally, we derived Eq. (19) to consider only one source item. From this point on, the terms that are only affected by the source items are denoted by $\tilde{\mathbf{B}}^S_*$.[9] Then, we transform Eq. (19) into Eq. (20) which has only one target item.[10] In Eq. (21), l determines the range of s and h, and also determines the number of times the relationship between the source item and the target item is learned. To be more specific, the relationship between source item i^u_1 and target item $i^u_{|\mathcal{S}^u|}$ is learned when l has a value between 1 and $|\mathcal{S}^u|$, and the relationship between source item i^u_2 and target item $i^u_{|\mathcal{S}^u|-1}$ is learned when l has a value between 2 and $|\mathcal{S}^u| - 1$. This pattern helps us to know how many times the relationship between source item i^u_h and target item i^u_s is learned. So, we can convert Eq. (21) to Eq. (22) using this pattern.

In conclusion, the objective function of multi-target augmentation assigns weight according to the position gap (*i.e.*, $h - s$) in learning the relationship between the target and source item. This is a problem when applying temporal weights. In particular, weighting by position gap can lead linear model learning to be biased towards long sequences. Consequently, we need to design a new augmentation method that is well-suited to learning temporal information.

A.2 Single-Target Augmentation

The derivation of single-target augmentation is similar to that of multi-target augmentation. However, unlike multi-target augmentation, it is not weighted by position gap, meaning that it is a more suitable approach to inject temporal information into linear models.

$$\hat{\mathbf{B}}_{\text{single}} = \operatorname*{argmin}_{\mathbf{B}} \sum_{u'=1}^{m'} \|\mathbf{T}^{\text{single}}_{u',*} - \mathbf{S}_{u',*}\mathbf{B}\|_F^2 \tag{23}$$

$$= \operatorname*{argmin}_{\mathbf{B}} \sum_{u=1}^{m} \sum_{h=1}^{|\mathcal{S}^u|-1} \|\mathbf{x}(i^u_{l+1}) - \mathbf{B}\mathbf{x}(i^u_{1:l})\|_F^2 \tag{24}$$

$$= \operatorname*{argmin}_{\mathbf{B}} \sum_{u=1}^{m} \sum_{l=1}^{|\mathcal{S}^u|-1} \Big\{ \sum_{s=1}^{l} \Big(\|\mathbf{x}(i^u_{l+1}) - \mathbf{x}(i^u_s)\mathbf{B}\|_F^2$$

$$- \|\mathbf{x}(i^u_s)\mathbf{B}\|_F^2 \Big) + \|\mathbf{x}(i^u_{1:l})\mathbf{B}\|_F^2 \Big\} \tag{25}$$

$$= \operatorname*{argmin}_{\mathbf{B}} \sum_{u=1}^{m} \sum_{l=1}^{|\mathcal{S}^u|-1} \sum_{s=1}^{l} \|\mathbf{x}(i^u_{l+1}) - \mathbf{x}(i^u_s)\mathbf{B}\|_F^2 + \tilde{\mathbf{B}}^S_1 \tag{26}$$

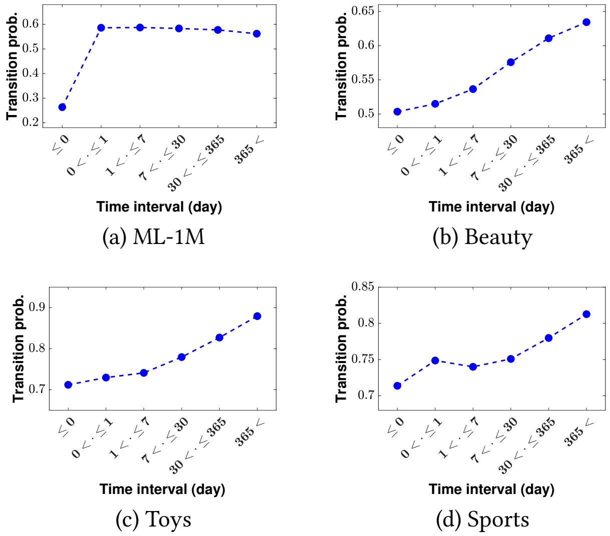
$$= \operatorname*{argmin}_{\mathbf{B}} \sum_{u=1}^{m} \sum_{s=1}^{|\mathcal{S}^u|-1} \sum_{h=s+1}^{|\mathcal{S}^u|} \|\mathbf{x}(i^u_h) - \mathbf{x}(i^u_s)\mathbf{B}\|_F^2 + \tilde{\mathbf{B}}^S_1. \tag{27}$$

[9] Single- and multi-target augmentation have the same estimate of the concentration matrix $(\mathbf{S}^\top\mathbf{S})^{-1}$.

[10] It is derived through a similar process of converting Eq. (17) to Eq. (18).

Table 8: Pearson correlation coefficients between inverse average time interval matrix and the learned item-item weight matrix $\hat{\mathbf{B}}$ for SLIST and TALE on five datasets.

Model	ML-1M	Beauty	Toys	Sports	Yelp
SLIST [5]	0.0130	0.0524	0.0718	0.0325	0.0910
TALE	**0.0375**	**0.3944**	**0.5602**	**0.5027**	**0.3885**



(a) ML-1M

(b) Beauty

(c) Toys

(d) Sports

Figure 6: Attribute transition probability by group of time interval on ML-1M, Beauty, Toys, and Sports. The x-axis is the time interval between two consecutive items, and the y-axis is the transition probability of genre/category.

B Detailed Experimental Setup

B.1 Hyperparameter Settings

Table 7 suggests the best hyperparameters of TALE. For Yelp, we only utilize a combination of TALE and SLIS and set α to 0.9. In [5], $\alpha \in [0, 1]$ is used to control the importance between SLIS and SLIT; if α is set to zero, only SLIT is used.

C Additional Experimental Results

This section contains the following experiments.

- Appendix C.1 verifies the existence of user preference drifts.
- Appendix C.3 shows the debiasing effect of TALE.
- Appendix C.4 demonstrates how well the learned item-to-item weight matrix $\hat{\mathbf{B}}$ incorporates temporal information (*i.e.*, user preference drifts over time).
- Appendix C.5 shows that the efficiency comparison between TALE and seven SR models except for efficient SR models (*i.e.*, LinRec [20], LRURec [39], and SLIST [5]).
- Appendix C.2, C.6–C.8 present the experimental results on the remaining three datasets (*i.e.*, Toys, Sports, and Yelp) that we were unable to include in the main text due to space limits.

C.1 Presence of User Preference Drifts

Figure 6 shows the transition probability of the item's attribute by time interval group. For attribute information, we utilize the attributes of each dataset. For ML-1M, the genre that best characterizes the movie was used as the attribute. For Beauty, Toys, and Sports attributes, we utilized item categories as representative attributes. Given a user sequence $i_1 \longrightarrow i_2 \longrightarrow i_3$, if the genres of i_1 and i_2 are horror and comedy, then one attribute transition has occurred. If the genres of i_2 and i_3 are the same, then no attribute change has occurred, and the user sequence has experienced one attribute transition within the user sequence. In this manner, the attribute transition probability is calculated for the entire user sequence and averaged by the time interval groups. For example, if the time interval of an item transition is 3 days, it falls in the range $1 < \cdot \leq 7$, and if the two items interacted at the same time, it belongs to the range ≤ 0. Figure 6 indicates that as the time interval increases, the probability of attribute transition increases in the four datasets. This result indirectly means that user preference changes over time. However, traditional linear SR models (*e.g.*, SLIST [5]) cannot reflect user preferences that change over time because they consider the time interval between successive items with equal weight. To address this issue, our proposed TALE utilizes temporal information to effectively capture user preference drifts.

C.2 Co-occurrences over Time Intervals

Figure 7 depicts the co-occurrence over average time intervals on Toys, Sports, and Yelp. They also show similar trends to ML-1M and Beauty in Figure 2. Note that Yelp shows a relatively uniform distribution compared to the other datasets because it is a place review dataset and sequential order is less important than others.

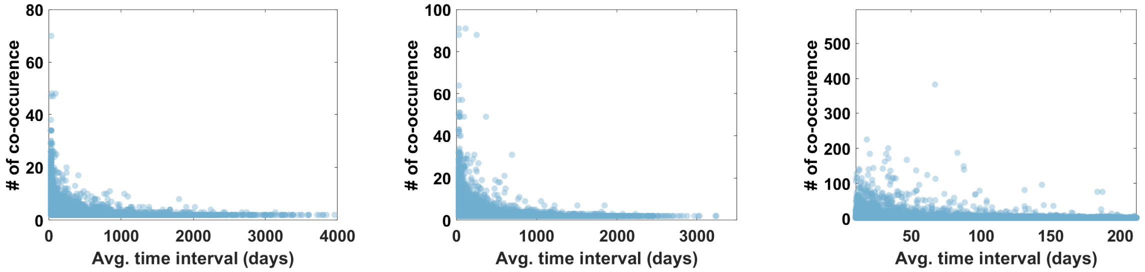
C.3 Effect of Trend-aware Normalization

Figure 8 depicts the average model prediction score of each item according to item popularity. Before describing the results, we introduce the two properties of the ideal distribution (if the model is perfectly debiased): (i) Uniform distribution. (ii) All weights with a value of 1.

We found the intriguing observations. (i) Compared to TALE, SLIST has a more skewed distribution where a few popular items have high scores, meaning that SLIST has more popularity bias. Meanwhile, TALE has a more uniform distribution because it mitigates the popularity bias. (ii) Applying trend-aware normalization elevates the performance of tail and head items by slightly increasing the weights of the overall items, as evidenced in Table 5.

C.4 Analysis of Temporal Information

Table 8 indicates how SLIST and TALE reflect temporal information by using the Pearson correlation coefficient. Inspired by the observation that the probability of user preference drifts is proportional to the average time interval (Refer to Figure 6), we analyze the correlation between the average time interval of items and the learned item-item weight matrix $\hat{\mathbf{B}}$. Specifically, we first compute the average time interval between consecutive items. These values constitute the average time interval matrix of the same size as the item-item matrix. Since shorter average time intervals are generally

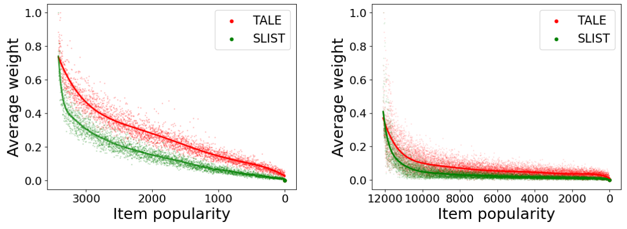


(a) Toys (b) Sports (c) Yelp

Figure 7: Co-occurrences between two consecutive items over average time interval on Toys, Sports, and Yelp.

Table 10: Tail and Head performance comparison on Toys, Sports, and Yelp. 'Norm.' denotes the existing normalization method, *i.e.*, Eq. (4). Each metric is measured by NDCG@5.

Dataset	Toys		Sports		Yelp	
Model	Tail	Head	Tail	Head	Tail	Head
SASRec	0.0287	0.0431	0.0088	0.0263	0.0309	0.0330
BSARec	0.0306	0.0514	0.0094	0.0297	0.0343	0.0359
TiSASRec	0.0303	0.0487	0.0074	0.0281	0.0290	0.0343
SLIST	0.0376	0.0755	0.0100	0.0486	0.0371	**0.0380**
SLIST+Norm.	0.0415	0.0767	**0.0146**	0.0466	0.0402	0.0333
TALE	**0.0428**	**0.0818**	0.0133	**0.0512**	**0.0415**	0.0367



(a) ML-1M (b) Beauty

Figure 8: Correlation between item popularity and the average weight for each item on ML-1M and Beauty. The x-axis is the item ID sorted by popularity, and the y-axis is the average weight for each item.

Table 9: Efficiency comparison for TALE and seven SR models on ML-1M, Beauty, and Yelp. 'Train' and 'Eval' mean the runtime (seconds) for training and evaluation, respectively. For neural models, the runtime was measured on GPU, and for linear models on CPU.

Dataset	ML-1M		Beauty		Yelp	
Model	Train	Eval	Train	Eval	Train	Eval
SASRec [17]	1,475	11	128	42	252	43
DuoRec [23]	6,278	11	1,308	42	2,182	43
FEARec [8]	20,301	48	529	172	539	246
BSARec [25]	2,537	12	89	42	309	55
TiSASRec [19]	5,370	12	2,093	43	1,662	57
TCPSRec [30]	7,853	11	421	42	1,336	43
TiCoSeRec [7]	10,512	11	322	42	5,794	43
TALE	**179**	**0.2**	**5**	**10**	**13**	**21**

more highly correlated, we take the element-wise inverse of the average time interval matrix. Then, we compute the Pearson correlation coefficient between this matrix and the weight matrix $\hat{\mathbf{B}}$. The larger the coefficient, the better $\hat{\mathbf{B}}$ reflects the temporal information. In Table 8, TALE has higher coefficients than SLIST in all datasets, indicating that it effectively models user preference drifts into the item-item weight matrix.

C.5 Efficiency Comparison

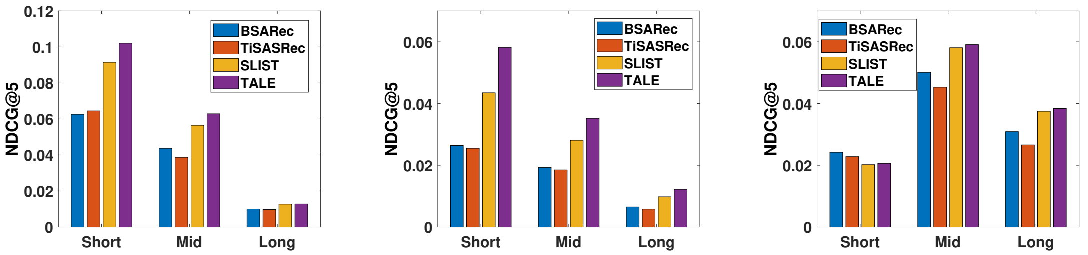
Table 9 shows the training and evaluation time of TALE and the comparison models on three datasets. TALE has the fastest training time, 8.2x, 25.6x, and 19.4x faster than SASRec for ML-1M, Beauty, and Yelp, respectively. Since the training time was measured for a single model training, when hyperparameter tuning is performed, the neural models take a much longer training time than TALE. Among the neural models, SASRec and BSARec have the shortest training times, while the contrastive learning-based models (*i.e.*, DuoRec, TCPSRec, and TiCoSeRec) have longer training times because of the data augmentation and the computation of contrastive loss. Looking at the efficiency of LRURec in Table 3, it has a long evaluation time compared to other neural models, which is caused by the serial processing of RNN.

C.6 Tail and Head Performance

Table 10 shows Tail and Head performance on Toys, Sports, and Yelp. In all three datasets, we can see that TALE outperforms other models on tail items, alleviating popularity bias. The existing normalization (*i.e.*, Norm.) also shows superior Tail performance, indicating the debiasing effect of normalization.

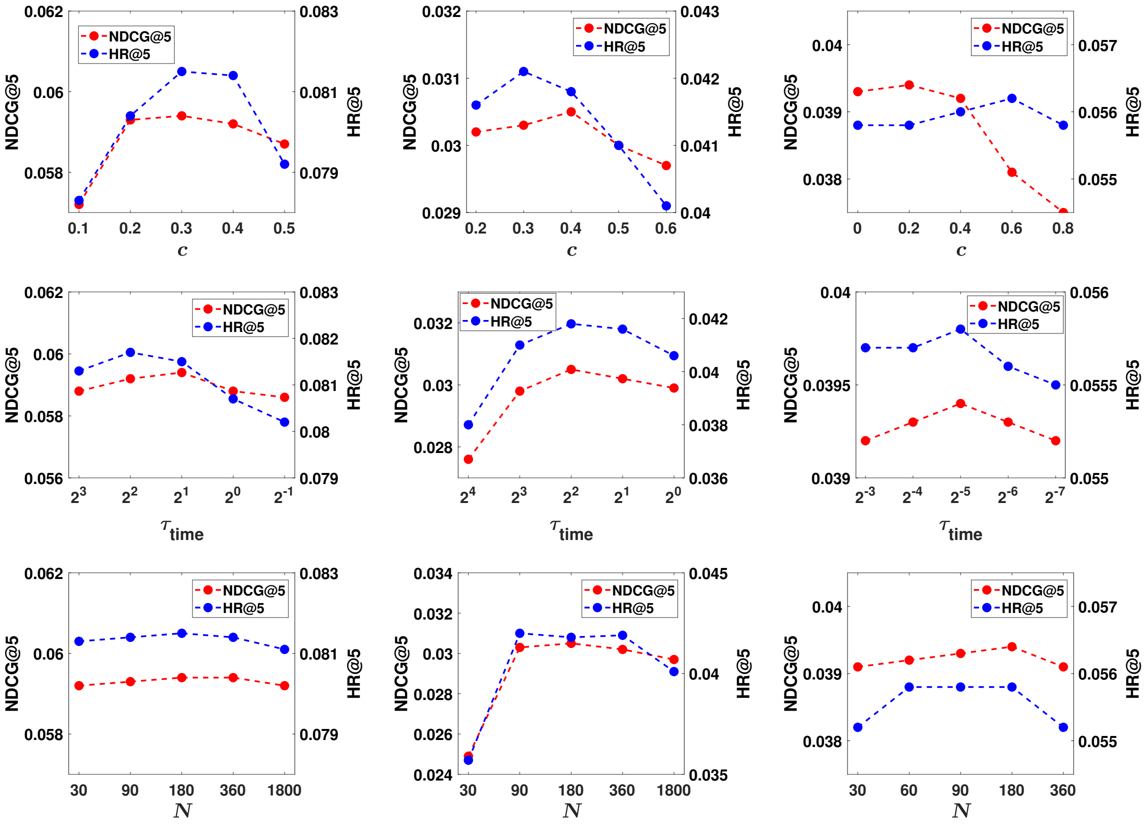
C.7 Performance over Time Intervals

Figure 9 illustrates the performance of the three datasets over the three groups as time intervals (*i.e.*, Short, Mid, and Long). On Toys and Sports, TALE excels in all groups, similar to the Beauty dataset, while the Yelp dataset performs well in Mid and Long. The consistent performance in Mid and Long suggests that TALE successfully captures the relation between items with long-time intervals. Meanwhile, the performance of the Yelp dataset for Short is lower than for Mid and Long because users do not consume similar items in a row, which is a characteristic of the Yelp dataset. For example, a user



(a) Toys (b) Sports (c) Yelp

Figure 9: Accuracy comparison by the time interval group on Toys, Sports, and Yelp. Each metric is NDCG@5.



(a) Toys (b) Sports (c) Yelp

Figure 10: Performance of TALE over the three hyperparamters, *i.e.*, c, τ_{time}, and N, on Toys, Sports, and Yelp.

visiting a Chinese restaurant does not immediately visit another Chinese restaurant.

C.8 Hyperparameter Sensitivity

Figure 10 shows the performance of TALE on Toys, Sports, and Yelp according to the three hyperparameters (*i.e.*, c, τ_{time}, and N). Based on the dataset statistics, we divide them into three groups. *Group1*: (Beauty, Toys, and Sports), *Group2*: Yelp, and *Group3*: ML-1M. The average time interval and optimal τ_{time} become smaller in the order of Group 1, 2, and 3. This is because it is natural to give a weaker time decay for shorter average time intervals (The numerator and denominator become similar in scale.). On Toys, Sports, and Yelp, optimal values of c are 0.3, 0.4, and 0.2, respectively, indicating that long-time dependency varied depending on the dataset characteristics. The three datasets achieve optimal performance when the window size N for trend-aware normalization is 180.