

Article

Repeat-Explore-Aware Grocery Next-Basket Recommendation with Time-Decayed LightGCN

Zahra Mansouri, Salim Lahmiri * and Rustam Vahidov

Department of Supply Chain and Business Technology Management, John Molson School of Business, Concordia University, Montreal, QC H3H 01A, Canada; zahra.mansouri@concordia.ca (Z.M.); rustam.vahidov@concordia.ca (R.V.)

* Correspondence: salim.lahmiri@concordia.ca

Abstract

Grocery shopping is highly repetitive, yet customers also introduce new products into their buying routine over time, creating a repeat-explore challenge for next-basket recommendation. This study examines this trade-off in offline grocery recommendation using the Dunnhumby Complete Journey and Ta-Feng datasets. The task is formulated at the household-product level and evaluated using overall, repeat-specific, and explore-specific Top-K metrics. We compare global and personal frequency baselines with GraphSAGE, GCN, LightGCN, and weighted and time-decayed graph variants, and propose two hybrid models: TL-HFE, which combines an exploitation-oriented LightGCN ranking over the learnable catalog with a history-filtered exploration head, and TL-PPR, which combines a non-parametric personal-popularity repeat head with a LightGCN-based exploration head and household-specific quota interleaving. On Dunnhumby, Personal Top-Frequency achieves Overall, Repeat, and Explore Recall@20 values of 0.20, 0.40, and 0.00, while TL-PPR achieves 0.15, 0.29, and 0.02. On Ta-Feng, TL-PPR achieves 0.19, 0.67, and 0.07, compared with 0.18, 0.80, and 0.03 for Personal Top-Frequency. Paired bootstrap tests confirm that TL-PPR significantly improves exploratory recommendation over Personal Top-Frequency, although repeat recovery remains stronger for the baseline. Overall, the findings show that grocery NBR should be evaluated through separate repeat and explore perspectives rather than aggregate accuracy alone, especially when product discovery is a practical objective.

Keywords: next-basket recommendation; grocery recommendation; recommender systems; repeat-explore trade-off; exploratory recommendation; LightGCN; graph neural networks; time-decayed recommendation; Bayesian personalized ranking; offline retail analytics

Academic Editor: Qiang Qu

Received: 8 May 2026

Revised: 17 June 2026

Accepted: 29 June 2026

Published: 30 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Retailers operate in highly competitive environments where understanding individual customer purchase behavior is difficult yet crucial, as it influences both customer outcomes and company decisions [1]. This challenge is particularly significant in offline grocery retail for two main reasons. First, grocery shopping is frequent, routine, and influenced by household-specific preferences, budgets, dietary needs, and habits. Second, grocery retailers must manage large product assortments, perishable items, inventory levels,

product turnover, and promotional strategies [2]. These combined factors make predicting customer demand both practically vital and operationally complex.

Recommender systems utilize a data-driven approach to personalize retail experiences. They analyze past customer-item interactions to suggest relevant products [3]. In grocery retail, these systems assist customers in managing extensive product selections and lowering search effort, while also helping retailers increase basket size, purchase frequency, target promotions more effectively, and better match demand [4,5]. However, unlike many online retail environments, offline grocery recommendations are more complex because they often rely primarily on transaction data rather than detailed behavioral signals such as clicks, browsing history, and search queries. As a result, grocery recommender systems must infer household preferences from purchase histories, basket contents, repeat-buying habits, and temporal patterns in shopping behavior [4,6].

In recommender systems research, next-basket recommendations (NBRs) are particularly relevant to grocery retail, as customers often purchase multiple products in a single shopping trip. Rather than predicting only one next item, NBR aims to predict the set of products that a customer is likely to buy in the next basket [7,8]. In this task, a customer's previous baskets are modeled as a sequential purchase history, and the next basket is used as the prediction target [5]. This differs from conventional sequential or session-based recommendation, where the objective is usually to predict the next individual item [3]. In NBR, the model must recommend multiple items that can be purchased together while treating items in the same basket as unordered observations. As a result, NBR models need to capture dependencies across baskets over time, item co-occurrence patterns, and household-level consumption behavior, making the task more complex than standard item-level recommendation [5,7,9,10].

A key behavioral challenge in grocery NBR is differentiating between repeat and exploratory purchasing. Repeat items are those that appeared in a household's previous transactions, while exploratory items are new to that household [5,7,11]. This distinction is especially vital in grocery retail, where shopping tends to be routine, repetitive, and low-involvement [4]. Unlike categories where customers actively search for new products, grocery shopping is often driven by habits, convenience, household needs, and established consumption patterns. Consequently, many future purchases are likely to be items customers have bought repeatedly before [4,12,13]. Research has shown that predicting repeat items is much easier than recommending exploratory ones because repeat items benefit from existing historical data, whereas exploratory items require inference without prior household-item interactions [5,7,10,13]. Therefore, the repetitive structure of grocery baskets can benefit history-based recommendation models, but it can also create repeat-biased recommendation behavior [14].

This repetitive structure poses a key evaluation challenge: relying solely on aggregate accuracy may not be enough to assess NBR models. A model can perform well overall by mainly promoting habitual purchases, yet still underperform on exploratory recommendations [5,7,11]. Li et al. [7] address this problem by developing metrics specific to repeat and explore recommendations, as well as analyzing the repetition-exploration ratio in the recommended basket. Their findings indicate that many perceived improvements in NBR are actually due to bias towards repeat recommendations. Consequently, NBR evaluation should include not only overall accuracy but also separate assessments of the effectiveness of repeat and exploratory recommendations.

Enhancing exploration also creates a wider trade-off between exploration and accuracy. Since exploratory items are more difficult to predict, increasing the proportion of novel recommendations might decrease overall accuracy [5]. Models focused mainly on accuracy tend to favor familiar, repeat items, while those that emphasize novelty, diversity, or exploration may compromise some predictive accuracy. Therefore, methods that

boost exploratory recommendation while limiting declines in both overall and repeat accuracy are particularly valuable for grocery NBR [5,15,16].

Various studies have tackled the repeat-explore issue by explicitly differentiating between the two processes. For example, RepeatNet proposed a repeat-explore framework for session-based recommendations by splitting the task into two modes: a repeat decoder that suggests items from the user's previous session and an explore decoder that recommends new items outside that history [11]. A neural switching mechanism is then used to estimate the probability of selecting each mode. Although RepeatNet was designed for next-item session prediction rather than grocery next-basket recommendation, it provides an important conceptual foundation for treating repetition and exploration as distinct recommendation behaviors. However, RepeatNet implicitly determines the repeat-explore trade-off via a learned mode-switching probability. In contrast, our proposed approach selects this trade-off explicitly through a validation-based composite score that jointly considers Overall, Repeat, and Explore Recall. This makes the model-selection criterion more transparent and better aligned with grocery next-basket recommendation, where optimizing only for repeated purchases may improve repeat accuracy while limiting the model's ability to recover exploratory items.

More recently, Liu et al. [12] examined NBR and found that optimizing for diversity or item fairness without considering repeat bias can unintentionally lead to a greater focus on repeat recommendations. Their research highlights the importance of evaluating objectives beyond accuracy in repeat-heavy NBR contexts.

The exploration component of this study relates to the broader exploration-exploitation literature in recommender systems. Exploitation-focused recommenders tend to repeatedly show users familiar or high-confidence items, forming a closed loop that restricts the discovery of broader user interests. For instance, Song et al. [17] introduce hierarchical contextual bandit methods that enable exploration across the entire item space, using online bandit learning to adapt recommendations based on user feedback over time. Conversely, this research addresses a similar challenge in an offline grocery NBR context, employing a history-filtered exploration head and household-specific quota interleaving.

The most closely related framework to this study is TREx, the Two-step Repetition–Exploration framework by Li et al. [5], which separates repetition and exploration in next-basket recommendation and controls their proportions via a repetition confidence threshold, primarily to study whether diversity and fairness can be improved while maintaining overall accuracy. This study builds on the same insight that repeat and explore items should be modeled and evaluated separately, but differs in both objective and methodology: rather than using exploration mainly as a plug-in component for beyond-accuracy optimization, we propose graph-based NBR models that learn household-item collaborative signals through binary, weighted, and time-decayed interaction graphs. In addition, our hybrid models combine graph-based exploration with household-specific repeat heads, and the repeat-explore trade-off is selected through a validation-based composite score over Overall, Repeat, and Explore Recall, making the trade-off explicit and aligned with predictive performance rather than fairness or diversity alone.

To address the complexity of NBR, earlier studies have employed various machine learning and deep learning approaches, including sequential [18], attention-based [19], and hybrid models [5], as well as techniques that examine basket dependencies and purchase timing [5,8]. Recently, deep learning recommender systems have become prominent in digital services due to their ability to capture intricate relationships among users, items, and context [20]. However, evaluations show that complex neural models do not always outperform simpler approaches—such as those based on purchase frequency and recency—especially when dealing with data involving repeated behaviors [8]. This underscores an important insight in grocery NBR: increasing model complexity does not neces-

sarily lead to better results. What truly matters is the model's capacity to recognize behavioral patterns—such as how often, how recently, and how likely customers are to repeat purchases—that impact future buying decisions.

Graph-based models are increasingly popular in deep learning recommender systems [21–23]. This is because the relationship between customers and products naturally fits a graph structure, with shoppers and items as nodes and interactions as connections. This setup enables graph neural networks to capture complex collaborative relationships [20,24,25]. Recent work has also combined GNN representations with reinforcement learning for interactive recommendation. For example, Li et al. [23] propose GraphDRL, a GNN-based deep reinforcement learning model for sparse interactive recommendation. However, GraphDRL differs from the present study because it focuses on interactive recommendation and exploration-exploitation decision-making through reinforcement learning, whereas this study addresses offline grocery next-basket recommendation using historical transaction data.

In graph neural network models, LightGCN stands out as a key graph-based collaborative filtering model for its simplified approach: it removes feature transformations and nonlinear activations, instead relying on linear neighbourhood propagation across the user-item interaction graph [26]. This makes LightGCN both effective and efficient, positioning it as a strong foundation for recommendation systems.

Standard GNN-based collaborative filtering models, such as LightGCN, face limitations in next-basket recommendation (NBR) because they usually represent historical user-item interactions as a static graph. In its original form, LightGCN learns embeddings through normalized neighborhood aggregation, but it does not explicitly distinguish recent purchases from older ones or preserve the chronological order of basket histories [26]. This is restrictive in grocery NBR, where frequency, recency, and routine behavior strongly influence future purchases [8]. Time-aware collaborative filtering studies support this concern [27,28]. Ding and Li [29] argue that recent interactions should have greater influence than older ones, while Wu et al. [30] propose a piecewise decay function to capture the changing influence of past ratings over time. Moreover, because LightGCN can be sensitive to noisy interactions [31], treating all past purchases equally may allow outdated or less relevant items to distort embedding propagation. Therefore, incorporating temporal weighting into the graph structure can make GNN-based NBR more responsive to evolving household preferences.

Unlike dynamic graph recommender models such as DGSR and DGEL [32,33], which model temporal information through sequential graph evolution or real-time embedding updates, this study incorporates temporal information directly at the graph construction stage. Specifically, temporal recency is embedded into the household-item edge weights before the graph-based recommendation models are trained.

Our approach is also related to the construction of weighted bipartite graphs. Liu and Wang [34] show that customer-product bipartite graphs can use edge weights to represent the frequency with which a customer purchases a product. Pande et al. [35] further demonstrate that edge weights can be incorporated into graph-based recommendation through weighted sampling, aggregation, random walks, and loss design. Importantly, SWAG constructs product-product graph weights from behavioral co-occurrences such as views, add-to-cart events, and purchases, and applies time decay to co-occurrences to capture item recency [35]. These studies support using weighted graph edges to represent interaction strength rather than treating all observed edges equally.

More directly related to temporal weighting, Wu et al. [36] transform implicit feedback data into a weighted user-item bipartite graph using an exponential time-decay function, where more recent interactions receive higher weights. However, their time-decayed graph is used for graph filtration and positive sample construction rather than as the di-

rect propagation graph for GNN-based recommendation. Similarly, Chanpuriya et al. [37] show in temporal graph learning that temporal proximity can be encoded through decay-based edge weights, although their method constructs a time-decayed line graph for temporal edge embedding rather than a household-item recommendation graph.

Moreover, Zhang et al. [38] combine exponential time-decayed edge weights with LightGCN-style propagation in a user-content bipartite graph, where interactions such as views, likes, comments, and shares are discounted over time to improve communication-effectiveness prediction. Huang et al. [39] also use temporal weighting in graph-based recommendation by constructing a Time Decay Heterogeneous Graph for query-oriented scientific paper recommendation, where a time-decay vector reduces the over-weighting of older papers in Random Walk with Restart (RWR) ranking. These studies support the broader value of temporal weighting in graph-based recommendation. However, they differ from the present work in task, graph structure, and objective: Zhang et al. focus on user-content diffusion, while Huang et al. focus on paper recommendation in a heterogeneous citation network. In contrast, this study constructs a household-item grocery purchase graph in which time-decayed edge weights aggregate repeated purchase events so that each edge reflects both purchase frequency and temporal recency before LightGCN-based repeat-explore next-basket recommendation.

Building on these ideas, this study constructs a time-decayed weighted household-item graph in which each edge captures both the frequency and temporal recency of a household's purchases. Specifically, the edge weight between a household and an item is obtained by summing the exponentially decayed contributions of all purchase events observed during the training period, so that more recent purchases contribute more strongly while repeated purchases still increase the overall interaction strength. This construction differs from binary graph representations, which collapse all historical purchases into a single unweighted edge, and from raw frequency weighting, which treats all purchases equally regardless of when they occurred. The resulting edge weight is therefore a continuous frequency-recency interaction strength measure, which is used as the input edge weight for the graph-based recommendation models examined in this study.

Another relevant direction in recent GNN recommendation research concerns the interpretation of ranked Top-K recommendation lists. Chen et al. [40] propose GREASE, a factual and counterfactual explanation framework for Top-K GNN-based recommendations, showing that recommendation explanations must account for ranked lists rather than only binary link prediction. This perspective is relevant to grocery NBR because the final Top-K basket may contain both repeat and exploratory items; therefore, this study evaluates recommendation lists using overall, repeat-specific, and explore-specific metrics.

Taken together, prior research suggests that grocery NBR requires models that address both behavioral and representational challenges. On the behavioral side, NBR models should distinguish repeat and exploratory recommendations, because strong aggregate accuracy may hide poor exploratory performance. On the representational side, graph-based models should account for interaction strength and temporal recency rather than treating all historical purchases as equally informative. However, limited attention has been given to LightGCN-based architectures specifically designed for grocery NBR that jointly incorporate time-decayed interaction graphs, repeat-explore separation, and household-specific exploration tendencies. This gap motivates the present study.

This study aims to examine whether incorporating interaction strength, temporal recency, and explicit repeat-explore separation can improve grocery next-basket recommendation. To achieve this, two hybrid LightGCN-based models are proposed: TL-HFE and TL-PPR. Both models are built on a household-item graph, where edges can be represented in three ways: binary edges, raw frequency-weighted edges, or time-decayed

weighted edges. While the proposed models do not explicitly model basket sequences using recurrent or Transformer-based architectures, the time-decayed graph variant embeds temporal information directly into the graph structure through recency-weighted household-item edges.

The initial model, TL-HFE, integrates a LightGCN repeat-ranking system with a history-based exploration component to suggest both past and new items. The second model, TL-PPR, combines a graph exploration method with a personal-popularity repeat system to better capture routine household repurchases. In both models, repeat and exploration rankings are generated separately and then merged using household-specific quota interleaving. This structure allows the final recommendation list to adjust the repeat-explore balance according to each household's recent purchasing pattern, rather than applying a uniform repeat-explore ratio across all households.

To further position this study, Table 1 compares representative related work in GNN-based recommendation, time-aware recommendation, repeat-explore recommendation, and exposure-control methods.

Table 1. Comparison of related work.

Study	Main Task	Core Method	Time-Aware?	Repeat/Explore-Aware?	User-Specific Allocation?	Limitation Relative to This Study
RepeatNet [11]	Session-based next-item recommendation	Repeat/explore decoders	Sequential session context	Yes	Learned switch	Not designed for offline grocery NBR or graph-based household-item modeling
TREx [5]	Next-basket recommendation	Two-step repetition-exploration framework	Limited/not graph-edge based	Yes	Yes/controlled balancing	Does not use time-decayed LightGCN edge construction
SUPER [41]	Popularity debiasing/exposure control	User-specific popularity exposure calibration	No	Not repeat/explore-specific	Yes	Controls popularity exposure, not household-level repeat/explore basket construction
This study	Offline grocery NBR	Time-aware LightGCN + repeat/explore hybrid ranking	Yes, through time-decayed edge weights	Yes	Yes, household-specific quota interleaving	Targets repeat-heavy grocery NBR with overall/repeat/explore evaluation

The main contribution of this study lies in adapting and integrating several complementary ideas into a repeat-explore-aware framework specifically designed for offline grocery next-basket recommendation. Prior studies have examined LightGCN-based collaborative filtering, time-aware recommendation, repeat-explore separation, and user-specific exposure control. However, limited work has examined these components jointly in offline grocery next-basket recommendation, particularly using graph edge constructions that explicitly encode interaction frequency and temporal recency.

The contributions of this study are summarized as follows:

1. **A time-aware, non-dynamic household-item graph construction for grocery NBR.**

This study constructs household-item edge weights by aggregating purchase events with exponential time decay, so that each edge reflects both purchase frequency and temporal recency before LightGCN propagation. The proposed construction does not introduce a new propagation rule, but adapts the LightGCN input graph to better represent the frequency-recency structure of grocery purchase histories.

2. **Two repeat-explore-aware hybrid ranking frameworks for final basket construction.**

This study proposes TL-HFE and TL-PPR as hybrid ranking frameworks that separate repeat-oriented and exploration-oriented recommendations before combining them into the final Top-K basket. TL-HFE uses a shared LightGCN embedding space to generate

an exploitation-oriented ranking and a history-filtered exploration ranking, while TL-PPR combines a personal-popularity repeat head with a LightGCN-based exploration head. Both models use household-specific repeat-explore allocation to better reflect differences in individual repurchasing behavior.

3. A repeat-explore-aware validation and evaluation protocol.

Because overall accuracy in grocery NBR can be dominated by repeat purchases, this study evaluates all models using overall, repeat-specific, and explore-specific metrics. In addition, model configurations are selected using a composite validation score that balances Overall Recall@20, Repeat Recall@20, and Explore Recall@20. This score is not proposed as a new recommendation metric or training objective, but as a transparent model-selection criterion to avoid choosing configurations that perform well mainly by recovering routine repurchases.

4. An empirical evaluation on two widely used grocery datasets with tuning, sensitivity, and ablation analysis.

Together, these contributions clarify that the novelty of this study lies in the combination of time-aware graph construction, repeat-explore hybrid ranking, household-specific quota-based basket construction, and repeat-explore-aware model selection in the context of grocery next-basket recommendation.

The empirical results show that grocery next-basket recommendation is strongly shaped by the repeat-explore trade-off. Repeat-oriented baselines, especially Personal Top-Frequency, remain highly competitive because grocery purchases are largely habitual. However, these methods provide limited support for exploratory recommendations. In contrast, graph-based models recover more new-to-household items but are generally weaker in repeat-item prediction. The proposed hybrid models, particularly TL-PPR, help bridge this gap by combining a strong personal-popularity repeat head with a LightGCN-based exploration component and household-specific quota interleaving. Overall, the findings suggest that explicitly modeling and evaluating repeat and exploratory behavior provides a more informative and practically useful framework for grocery next-basket recommendation than relying only on aggregate accuracy.

2. Materials and Methods

This section describes the dataset, task formulation, repeat-explore decomposition, evaluation framework, and recommendation models used in this study.

2.1. Dataset

This study uses two widely used offline grocery transaction datasets, the Dunnhumby Complete Journey retail dataset and the Ta-Feng grocery dataset, both of which have been commonly employed in grocery and next-basket recommendation studies [5,7,42]. The Dunnhumby Complete Journey dataset is a household-level grocery transaction database that records approximately two years of purchases from 2500 frequent-shopper households. The Ta-Feng dataset contains grocery transactions from 32,266 customers of a Chinese retail store over the period from 1 November 2000 to 28 February 2001.

Although the two datasets differ in their original structure, both were converted into a common product-level next-basket recommendation format. Users are represented by households, items by PRODUCT_ID, and baskets at the user-day level. Specifically, all products purchased by the same user on the same day are aggregated into a single basket. For Dunnhumby, this corresponds to a household-DAY basket, while for Ta-Feng, where no separate receipt or basket identifier is available, baskets are defined using the CUS-TOMER_ID $\times$ TRANSACTION_DT combination.

During preprocessing, records without usable product information were removed. In the Dunnhumby dataset, alcohol-related products, including products in the spirits department, were excluded to focus on grocery-relevant departments. Transactions with zero quantity were also removed, and the pricing and discount fields were checked to reduce data quality issues. For both datasets, additional filtering was applied following prior next-basket recommendation studies [5,13] to ensure sufficient user activity and product-level support. Specifically, product IDs were removed if they appeared in fewer than five user-day baskets or were purchased by fewer than two users. Users were removed if they had fewer than three distinct shopping days.

Table 2 summarizes the main dataset statistics before and after preprocessing. The preprocessed datasets were then used in the unified leave-last-day-out validation and test protocol described in the next sections.

Table 2. Summary of the grocery transaction datasets before and after preprocessing.

Dataset	Before Preprocessing			After Preprocessing		
	Number of Customers	Number of Product_ID	Number of Customer-Day Baskets ¹	Number of Customers	Number of Product_ID	Number of Customer-Day Baskets
Dunnhumby	2500	92,339	225,533	2487	25,902	199,805
Ta-Feng	32,266	23,812	119,578	14,012	13,892	94,354

¹ Customer-day baskets: all products bought by one customer on one shopping date.

2.2. Next-Basket Recommendation (NBR) Task Formulation

Let $H = \{h_1, h_2, \dots, h_n\}$ denote the set of households, and $I = \{i_1, i_2, \dots, i_m\}$ the set of items. The recommendation task is defined at the household-day level; therefore, each basket is represented as an unordered set of unique product IDs purchased by a user on a given shopping day.

For each household, we observe a temporally ordered sequence of daily baskets:

$$S_h = \{B_h^1, B_h^2, \dots, B_h^{t_h}, B_h^{t_h+1}\}, \quad (1)$$

where $B_h^t \subseteq I$ represents the set of items purchased by a household h on a day t . Because households have different numbers of observed shopping days, t_h is household-specific and denotes the index of the last basket/day in the training history of the household h . The following basket, $B_h^{t_h+1}$, is held out as the ground-truth next basket.

The objective of the NBR task is to predict which items will appear in the household's next basket, given the user's historical purchase sequence. Formally, the observed purchase history for a household h is defined as:

$$O_h = \{B_h^1, B_h^2, \dots, B_h^{t_h}\}, \quad (2)$$

and the ground-truth target basket, or the next basket to be predicted, is defined as:

$$T_h = B_h^{t_h+1}. \quad (3)$$

Given the observed training history O_h , a recommendation model f produces a ranked list of candidate items for the household h . In practice, we evaluate only the Top-K recommended items:

$$\hat{P}_h^K = f(O_h), \quad (4)$$

where $\hat{P}_h^K$ denotes the ordered Top-K list of items recommended to the household h .

2.3. Data Split

To support hyperparameter tuning while avoiding test-set leakage, this study adopts a unified leave-last-day-out validation and test protocol. For each user, the final observed

shopping day is held out as the target next basket, and all earlier shopping days are used as training history. Following prior next-basket recommendation work [5,9], users are randomly divided into two disjoint groups: a validation group, used for model selection, hyperparameter tuning, ablation analysis, and sensitivity analysis, and a test group, reserved only for final evaluation.

In the Dunnhumby dataset, this split results in 1244 validation users and 1243 test users. In the Ta-Feng dataset, this results in 7006 validation users and 7006 test users. In both datasets, each evaluated user has one held-out basket and at least two training shopping days.

The training catalog is defined as the set of product IDs observed in the training data, and recommendations are generated only from this catalog because products unseen during training cannot be learned by the models. For Dunnhumby, all validation and test products are covered by the training catalog. For Ta-Feng, four products in the validation set and four products in the test set are absent from the training catalog; these cold-start products are therefore excluded from the learnable ground-truth set during evaluation.

2.4. Repeat-Explore Decomposition

Following Li et al. [5], this study assumes a fixed learnable item universe, defined by the products observed in the training catalog. Although products may be introduced or discontinued over time in real retail settings, modeling product entry and exit is beyond the scope of this study. With this fixed-universe assumption, a household's repeat-item set grows as it makes more purchases, and the set of truly new candidate items decreases. To better assess recommendation performance, we break down both the target basket and the recommendation list into repeat and explore components, based on each household's purchase history.

Since previously we defined the observed training history for each household h as $O_h = \{B_h^1, B_h^2, \dots, B_h^{t_h}\}$, the set of unique items previously purchased by the household h during the training period is defined as the union of all items in previous baskets:

$$I_{h,t_h}^{rep} = \bigcup_{t=1}^{t_h} B_h^t, \quad (5)$$

This set represents the household-specific repeat-item space at the time of prediction. The corresponding exploration-item space is defined as the complement of the repeat-item space within the full item universe:

$$I_{h,t_h}^{expl} = I \setminus I_{h,t_h}^{rep}, \quad (6)$$

Therefore, an item is considered a repeat item for a household h if it has appeared at least once in that household's training history, and an item is considered exploratory if it has not previously appeared in that household's training history. Because households accumulate purchases over time, the repeat-item set grows monotonically, while the exploration-item set correspondingly shrinks. This reflects the fact that, as a household purchases more items, fewer items remain truly novel to that household.

Using these definitions, the held-out target basket T_h can be partitioned into a repeat and an explore subset. The repeat component of the target basket is defined as:

$$T_h^{rep} = T_h \cap I_{h,t_h}^{rep}, \quad (7)$$

and the explore component is defined as:

$$T_h^{expl} = T_h \cap I_{h,t_h}^{expl}, \quad (8)$$

Thus, the target basket can be decomposed as:

$$T_h = T_h^{rep} \cup T_h^{expl}, \quad T_h^{rep} \cap T_h^{expl} = \emptyset \quad (9)$$

This decomposition differentiates between two related yet distinct recommendation goals. The first is repetition, focusing on retrieving items already part of the household's purchase history. The second is exploration, aimed at identifying relevant new items for that household. This same distinction is used for the Top-K recommendation list, enabling separate evaluation of the model's performance on repeat versus exploratory recommendations. This differentiation is especially critical in grocery NBR, where overall accuracy can be heavily influenced by repeat purchases, potentially overshadowing the model's effectiveness on exploratory items.

In this study, the repeat-explore decomposition is applied separately for the validation and test evaluation groups. For every validation or test user, the repeat-item set is constructed only from that user's training history, (O_h) , and the held-out basket is decomposed into repeat and explore components relative to this fixed history. Therefore, validation and test target baskets are never used to define the user's repeat history. In addition, because recommendations are generated only from the training catalog, the item i universe used in the decomposition corresponds to the learnable catalog. Any cold-start products that are absent from the training catalog are excluded from the learnable ground-truth set before computing repeat and explore metrics.

2.5. Evaluation Metrics

Following Li et al. [5,7], this study evaluates recommendation performance from three perspectives: overall, repeat, and explore. This distinction is important in grocery next-basket recommendation because overall performance can be strongly influenced by habitual repurchases, while exploratory recommendations are typically more difficult. Therefore, model performance is assessed using metrics computed on the full held-out basket, as well as repeat- and explore-specific metrics computed on the decomposed target baskets and recommendation lists. The same metrics are used on the validation group for model selection, hyperparameter tuning, ablation analysis, and sensitivity analysis, while final performance is reported only on the held-out test group.

For each household h , the model produces a ranked Top-K recommendation list $\hat{P}_h^K$, which is compared with the held-out target basket T_h . Accordingly, the NBR problem is formulated as a Top-K item-ranking task at the product level, where K is a predefined cutoff. Prior NBR studies commonly evaluate recommendation performance at fixed basket sizes, particularly $K = 10$ and 20 [5,7]. This study reports results at $K = 10, 20$, and 30 to evaluate performance under compact and moderately broader recommendation lists. These cutoffs are also supported by the empirical distribution of held-out basket sizes in both datasets. In Dunnhumby, the final test baskets contain an average of 12.34 products and a median of 7 products; 63.65% of test households have held-out baskets with 10 or fewer products, 81.98% with 20 or fewer products, and 90.34% with 30 or fewer products. In Ta-Feng, held-out baskets are smaller, with an average of 6.62 products and a median of 5 products; 81.36% of test households have held-out baskets with 10 or fewer products, 95.55% with 20 or fewer products, and 98.99% with 30 or fewer products (Table 3). Therefore, $K = 10$ represents a compact recommendation setting close to the typical basket size, while $K = 20$ and 30 provide broader evaluation settings that cover most observed held-out baskets without producing unrealistically long grocery recommendation lists.

Table 3. Distribution-based justification for Top-K evaluation cutoffs.

Held-Out Basket Size Group	Dunnhumby		Ta-Feng	
	Number of Households	Percentage (%)	Number of Households	Percentage (%)
1–5	550	44.21	4035	57.59
6–10	242	19.45	1693	24.17
11–20	228	18.33	970	13.85
21–30	104	8.36	218	3.11
31–50	85	6.83	81	1.16
51+	34	2.81	9	0.13
Total	1243	100	7006	100

Let G denote an evaluation group, where $G \in \{G^{val}, G^{test}\}$. All metrics are computed at the household level and then averaged over the households in the corresponding evaluation group.

We use three ranking metrics commonly adopted in NBR research: Recall @ K, Normalized Discounted Cumulative Gain (NDCG@K), and Personalized Hit Ratio (PHR@K). Following prior NBR evaluation practice [7,10], we do not report Precision@K because it is more appropriate when the predicted basket size varies across households, whereas our evaluation adopts a fixed recommendation-list length. Under a fixed K, Precision@K is proportional to the number of hits in the Top-K list, while Recall@K normalizes these hits by the size of the held-out basket. Therefore, Recall@K is more suitable for grocery NBR, where basket sizes vary across households.

The overall Recall @ K for the household h is defined as:

$$Recall @ K (h) = \frac{|\hat{P}_h^K \cap T_h|}{|T_h|}. \quad (10)$$

The overall recall score for the evaluation group G is obtained by averaging Recall @ K (h) across all households in the relevant evaluation group.

NDCG@K (h) accounts for the rank positions of correctly recommended items and rewards models that place relevant items closer to the top of the list. For a household h , it is defined as:

$$NDCG@K (h) = \frac{\sum_{k=1}^K p_{h,k} / \log_2(k+1)}{\sum_{k=1}^{\min(K, |T_h|)} 1 / \log_2(k+1)}, \quad (11)$$

where $p_{h,k} = 1$ if the item ranked at position k in $\hat{P}_h^K$ belongs to T_h , and $p_{h,k} = 0$ if otherwise. The final NDCG@K score is computed by averaging household-level NDCG@K (h) values across all households in the relevant evaluation group.

PHR@K measures whether the model recommends at least one correct item for a household. At the household level, it is defined as:

$$PHR@K (h) = \varphi(\hat{P}_h^K, T_h), \quad (12)$$

where:

$$\varphi(\hat{P}_h^K, T_h) = \begin{cases} 1, & \text{if } \hat{P}_h^K \cap T_h \neq \emptyset \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

The final PHR@K is computed by averaging household-level PHR@K (h) values across all households in the relevant evaluation group.

To evaluate repeat and exploratory recommendations separately, the predicted Top-K list is decomposed using the household-specific repeat and explore item spaces defined in Section 2.4 while preserving the original ranking order:

$$\hat{P}_h^{K,rep} = \hat{P}_h^K \cap I_{h,t_h}^{rep}, \quad \hat{P}_h^{K,rep} = \{i \in \hat{P}_h^K : i \in I_{h,t_h}^{rep}\} \quad (14)$$

$$\hat{P}_h^{K,expl} = \hat{P}_h^K \cap I_{h,t_h}^{expl}, \quad \hat{P}_h^{K,expl} = \{i \in \hat{P}_h^K : i \in I_{h,t_h}^{expl}\} \quad (15)$$

As item sets, the Top-K recommendation list can be partitioned as follows:

$$\hat{P}_h^K = \hat{P}_h^{K,rep} \cup \hat{P}_h^{K,expl}, \quad \hat{P}_h^{K,rep} \cap \hat{P}_h^{K,expl} = \emptyset. \quad (16)$$

Repeat-specific metrics are computed only for households whose held-out basket contains at least one repeat item, while explore-specific metrics are computed only for households whose held-out basket contains at least one explore item. For evaluation group (G), the eligible repeat and explore household sets are defined as:

$$G^{rep} = \{h \in G : T_h^{rep} \neq \emptyset\}, \quad (17)$$

$$G^{expl} = \{h \in G : T_h^{expl} \neq \emptyset\}. \quad (18)$$

Repeat Recall@K and Explore Recall@K are then defined as:

$$Recall_{rep} @ K = \frac{1}{|G^{rep}|} \sum_{h \in G^{rep}} \frac{|\hat{P}_h^{K,rep} \cap T_h^{rep}|}{|T_h^{rep}|}, \quad (19)$$

$$Recall_{expl} @ K = \frac{1}{|G^{expl}|} \sum_{h \in G^{expl}} \frac{|\hat{P}_h^{K,expl} \cap T_h^{expl}|}{|T_h^{expl}|}. \quad (20)$$

Similarly, repeat and explore PHR@K are defined as:

$$PHR_{rep} @ K = \frac{1}{|G^{rep}|} \sum_{h \in G^{rep}} \varphi(\hat{P}_h^{K,rep}, T_h^{rep}), \quad (21)$$

$$PHR_{expl} @ K = \frac{1}{|G^{expl}|} \sum_{h \in G^{expl}} \varphi(\hat{P}_h^{K,expl}, T_h^{expl}). \quad (22)$$

Repeat and explore NDCG@K are computed using the original rank positions in the Top-K recommendation list, but relevance is judged only with respect to T_h^{rep} or T_h^{expl} , respectively. This means that repeat and explore items are not re-ranked after decomposition; instead, their positions in the original Top-K list are preserved. The resulting NDCG scores are averaged over, using G^{rep} or G^{expl} , respectively.

These decomposed metrics make it possible to determine whether performance gains come mainly from recovering habitual repurchases, improving exploratory recommendations, or both.

2.6. Recommendation Models

This study evaluates a set of benchmark and proposed recommendation models for offline grocery next-basket recommendation. The benchmark models include popularity-based baselines and graph-based collaborative filtering models, while the proposed models are repeat-exploration-aware hybrid models built on a LightGCN framework.

The models are organized in order of increasing complexity. First, two frequency-based baselines are introduced: a non-personalized global popularity model and a personalized household-level popularity model. Second, graph-based collaborative filtering models are presented, including GraphSAGE, GCN, and LightGCN, under different graph-construction strategies. Third, time-decayed graph variants are described to examine whether incorporating both purchase frequency and temporal recency improves recommendation performance. Finally, two hybrid models are introduced, which explicitly separate repeat-oriented and exploration-oriented recommendations before combining them into the final Top-K list.

2.6.1. Global Top-Frequency (G-TopFreq)

G-TopFreq is a non-personalized popularity baseline that recommends the same Top-K items to every household. This baseline is relevant in grocery recommendations because many purchases are concentrated among widely consumed staple products [5,43]. Item popularity is computed from the training data as the number of distinct households that purchased each item at least once [7]. Formally, the popularity score of an item i is defined as:

$$Pop(i) = \sum_{h \in H} 1(i \in I_{h,t_h}^{rep}), \quad (23)$$

where I_{h,t_h}^{rep} is the set of items previously purchased by household h during training. Items are ranked in descending order $Pop(i)$, and the same globally ranked list is assigned to all households.

2.6.2. Personal Top-Frequency (P-TopFreq)

P-TopFreq is a personalized, frequency-based baseline that recommends items based on each household's own purchase history [5,7]. Unlike G-TopFreq, which assigns the same globally popular items to all households, P-TopFreq ranks items separately for each household based on how frequently they appeared in that household's training baskets. Therefore, it represents a strong repeat-oriented baseline for grocery next-basket recommendation.

For the household h , the personal popularity score of item i is defined as:

$$Pop_h(i) = \sum_{t=1}^{t_h} 1(i \in B_h^t), \quad (24)$$

where $Pop_h(i)$ denotes the number of training basket-days in which household h purchased item i , and $1(i \in B_h^t)$ is an indicator function that equals 1 if item i appears in basket B_h^t , and 0 otherwise. Items with higher $Pop_h(i)$ values are ranked higher in the household-specific recommendation list. Ties are broken using global item popularity computed from the training data.

Because P-TopFreq only has personalized scores for items previously purchased by the household, it primarily recommends items from I_{h,t_h}^{rep} . If a household has fewer than K previously purchased items, the remaining positions in the Top-K list are filled with globally popular training items that have not already been selected. This ensures that each household receives a complete Top-K recommendation list while preserving the repeat-oriented nature of the baseline.

2.6.3. Graph-Based Collaborative Filtering Models

Graph-based collaborative filtering models represent historical household-item purchases as a bipartite interaction graph, where households and items are treated as two distinct node types, and observed purchases during the training period define the edges [21,22]. Let the household-item graph be defined as follows:

$$Graph = (H \cup I, E, W) \quad (25)$$

where H is the set of households, I is the set of product items, E is the set of observed household-item purchase edges, and W is the corresponding edge-weight matrix. An edge $(h, i) \in E$ exists if household h purchased item i at least once during the training period, that is, if $i \in I_{h,t_h}^{rep}$. Validation and test baskets are never used to construct the graph, ensuring that all learned representations are based only on historical purchase behaviour.

This study evaluates three graph-construction variants: binary edges, raw frequency-weighted edges, and time-decayed weighted edges. These variants allow us to examine whether graph-based recommendation benefits from moving beyond simple interaction existence toward interaction strength and temporal recency.

(A) Binary edges: In the binary graph, each unique observed household-item pair in the training history is represented as a unit-weight edge in the household-item bipartite graph. This representation is widely used in implicit-feedback recommendation, where observed interactions such as purchases are encoded as positive signals indicating that an interaction occurred [26,44]. Following graph-based collaborative filtering approaches that construct a user-item interaction graph from the interaction matrix, the binary edge set is defined as:

$$W_{h,i}^{binary} = 1 \text{ only if household } h \text{ purchased item } i \text{ at least once in training} \quad (26)$$

Thus, repeated purchases do not increase the edge weight in the binary graph. Household-item pairs with no observed purchase are not stored as zero-weight edges; they are absent from the sparse graph and may be treated as unobserved pairs during training. This representation captures only the existence of a purchase relationship and ignores how frequently or how recently the household purchased the item.

(B) Raw frequency-weighted edges: In the raw frequency-weighted graph, edge weights represent household-item interaction strength. Unlike the binary graph, this representation preserves how repeatedly a household purchased the same item during training period (Figure 1). Prior work on weighted bipartite graphs suggests that customer-product relationships may contain meaningful frequency information rather than only binary links. For example, Liu and Wang [34] define weighted customer-product bipartite graphs in which edge weights reflect the number of times a customer purchased a product. In GNN-based recommendation, Pande et al. [35] propose SWAG, a graph convolutional recommendation method for weighted product graphs, where edge weights between products are constructed using aggregate customer behavior. The weights reflect how strongly two products are related based on co-views, add-to-cart events, purchases, and recency-adjusted co-occurrence patterns. Similarly, Chen et al. [45] propose TGCN, which assigns weight 1 to observed user-item interaction edges, while user-tag and item-tag edges are weighted by tag annotation frequency. Therefore, the weights represent how strongly a user is associated with a tag and how strongly an item is associated with a tag. More recently, Priyam et al. [46] also emphasize that many recommender-system studies rely on binary user-item interaction encodings and propose EDGE-Rec, which directly models recommender-system graphs through weighted interaction matrices with real-valued edge weights representing interaction strength.

Following this logic, the raw frequency edge weight is defined as the number of distinct training basket-days on which household h purchased item i :

$$W_{h,i}^{freq} = \sum_{t \in \mathcal{T}} 1(i \in B_h^t) \quad (27)$$

where $1(i \in B_h^t)$ equals 1 if item i appears in household h 's basket on training day t , and 0 otherwise. This construction gives stronger graph influence to products that have been repeatedly purchased by the same household.

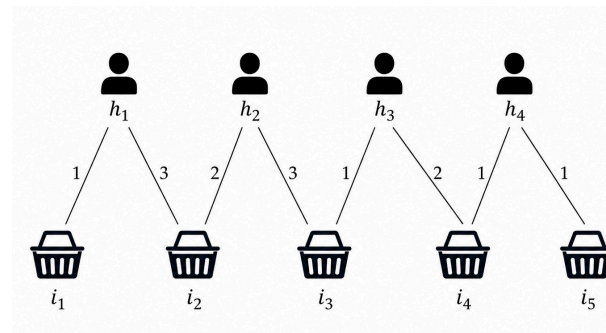


Figure 1. A weighted bipartite graph of the household-product network (the weight on the edge denotes the number of times that the customer has bought the product, households are shown with h and items with i).

(C) Time-decayed weighted edges: The third graph variant extends the raw frequency-weighted graph by incorporating temporal recency into the household-item edge weights. The binary graph captures only whether a household has purchased an item, and the raw frequency-weighted graph captures how often the household purchased the item. However, neither representation distinguishes recent purchases from older purchases. This distinction is important in grocery next-basket recommendation because household needs, replenishment patterns, and consumption preferences may change over time.

To address this limitation, each household-item purchase event is assigned an exponentially decaying contribution based on its age. Recent purchases therefore receive larger weights, while older purchases are retained with smaller weights rather than being discarded. This design preserves long-term household-item purchase signals while emphasizing interactions that are more likely to reflect current household preferences.

Following the general principle of time-aware collaborative filtering, Ding and Li [29] argue that interactions produced at different times should not be treated equally, because recent user behavior is more informative for predicting future preferences than older behavior. Recency-based recommender models commonly implement this idea using exponential decay functions, in which the influence of an interaction decreases as its timestamp moves farther away from the target recommendation time [47]. A similar exponential time decay strategy has also been used in implicit collaborative filtering to transform timestamped user-item interactions into a weighted user-item bipartite graph [36]. Motivated by these approaches, each household-item purchase event in this study is assigned a decayed contribution based on its temporal distance from the household's most recent training interaction.

Accordingly, we construct a time-decayed weighted household-item graph in which each purchase event contributes to the edge weight, but its contribution decreases as the purchase becomes older. For each household h and item i , let $\mathcal{T}_{hi}$ denotes the set of training days on which household h purchases item i . The final time-decayed edge weight between household h and item i is defined as:

$$w_{hi} = \sum_{t \in \mathcal{T}_{hi}} \exp(-\gamma \text{age}_{h,t}), \quad (28)$$

where $\text{age}_{h,t}$ is the time difference between the household h 's last observed training interaction and purchase day t . The decay rate is defined as:

$$\gamma = \frac{\ln(2)}{h_t}, \quad (29)$$

where h_l is the half-life parameter. The half-life represents the number of days required for the contribution of a purchase event to decrease to half of its original value. For example, when $h_l = 90$ days, the contribution of a purchase event is reduced by half every 90 days. Thus, recent purchases receive higher weights, while older purchases are retained with lower influence.

The resulting edge weight is a continuous interaction-strength value that jointly captures frequency and recency. If a household repeatedly purchases an item, the corresponding edge weight increases through the sum of multiple purchase events. However, recent purchase events contribute more strongly than older ones. This time-decayed weighted graph is then used as the input graph for the graph-based recommendation models examined in this study.

After defining the three edge-construction variants, namely binary, raw frequency-weighted, and time-decayed weighted edges, the resulting household-item graph is used as the input structure for the graph-based collaborative filtering models examined in this study. In all graph-based models, households and items are represented as nodes with trainable initial embeddings. Graph neural network propagation is then used to update these embeddings by aggregating information from neighboring nodes through the observed household-item purchase links.

Although the propagation mechanism differs across GraphSAGE, GCN, and LightGCN, the general principle is the same: household embeddings are updated using information from connected item nodes, and item embeddings are updated using information from connected household nodes. For the weighted graph variants, edge weights determine the relative strength of these neighboring signals during message passing. Therefore, edge weights indirectly affect recommendations by shaping the learned household and item embeddings. They do not directly change the final scoring function; rather, they influence the representations that are later used to compute household-item relevance scores.

After graph propagation, the final household and item embeddings are denoted by e_h and e_i , respectively. The relevance score between the household h and item i is computed using the dot product:

$$\hat{y}_{hi} = e_h^T e_i. \quad (30)$$

Items are ranked in descending order of $\hat{y}_{hi}$, and the Top-K items form the predicted basket $\hat{P}_h^K$. Previously purchased items are not removed during inference because repeat purchases are an essential component of grocery next-basket recommendation.

All graph-based models are trained using Bayesian Personalized Ranking (BPR), which encourages observed household-item interactions to receive higher scores than unobserved items:

$$\mathcal{L}_{BPR} = - \sum_{(h,i,j) \in \mathcal{D}} \log \sigma(\hat{y}_{hi} - \hat{y}_{hj}) + \lambda \|\Theta\|_2^2, \quad (31)$$

where $\mathcal{D}$ is the set of sampled training triplets, $i \in I_{h,t_h}^{rep}$ is a positive item from the household's training history, $j \notin I_{h,t_h}^{rep}$ is a sampled negative item, $\sigma(\cdot)$ is the sigmoid function, λ is the regularization coefficient, and Θ denotes the trainable model parameters.

Because the graph is a household-item bipartite graph, message passing occurs only through observed purchase links between households and products. This structure is suitable for collaborative filtering because it allows households and items to exchange information through historical purchase behavior. At the same time, bipartite propagation may also transmit noisy signals through highly popular products that connect many households. This potential limitation is discussed further in the limitations section.

GraphSAGE

GraphSAGE learns node representations by aggregating information from each node's local neighborhood [48,49]. In this study, GraphSAGE is applied to the household-item bipartite purchase graph constructed using the edge definitions described above. Therefore, the binary, raw frequency-weighted, and time-decayed weighted GraphSAGE variants share the same overall architecture and differ only in how neighbor contributions are weighted during aggregation.

In our implementation, GraphSAGE uses a weighted mean aggregation mechanism. For each node, the embeddings of neighboring nodes are aggregated according to the corresponding edge weights. The aggregated neighborhood representation is then combined with the node's current embedding and passed through a trainable transformation. This concatenate-and-transform structure allows GraphSAGE to preserve the node's own representation while incorporating collaborative signals from connected households and products.

Thus, the main difference among the GraphSAGE variants is not the scoring function or the training objective, but the way neighbor information is weighted during message passing. The binary version captures only whether a household purchased an item, the raw frequency version emphasizes repeatedly purchased items, and the time-decayed version emphasizes interactions that are both repeated and recent.

Graph Convolutional Network (GCN)

The Graph Convolutional Network (GCN) also learns household and item embeddings through neighborhood propagation, but it performs aggregation using a normalized adjacency matrix [26,50]. In this study, GCN is applied to the graph-construction variants defined earlier. Therefore, the binary, raw frequency-weighted, and time-decayed weighted GCN variants share the same model architecture and differ only in the edge weights used to construct the graph adjacency matrix.

Let R_w denote the weighted household-item interaction matrix, where each entry reflects the constructed edge weight between household h and item i . For the binary graph, R_w contains unit values for observed household-item interactions. For the raw frequency graph, R_w contains purchase-frequency weights. For the time-decayed graph, R_w contains frequency-and-recency adjusted weights. The corresponding bipartite adjacency matrix is defined as:

$$A_{hi} = \begin{bmatrix} 0 & R_w \\ R_w^\top & 0 \end{bmatrix}. \quad (32)$$

where the upper-right block represents household-to-item edges, and the lower-left block represents item-to-household edges.

To stabilize propagation and allow each node to retain part of its own representation, self-loops are added before normalization:

$$\tilde{A} = A + \alpha I \quad (33)$$

where I is the identity matrix and α is the self-loop weight. The adjacency matrix is then symmetrically normalized as:

$$\bar{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \quad (34)$$

where $\tilde{D}$ is the degree matrix of $\tilde{A}$.

Starting from trainable household and item embeddings, each GCN layer applies a learnable linear transformation and propagates information through the normalized adjacency matrix. ReLU activation is applied after intermediate layers, while the final layer is left linear to produce the final household and item embeddings used for recommendation.

Through this normalized adjacency matrix, edge weights control how strongly information is propagated between households and items. In the binary variant, all observed purchase links have equal initial strength. In the raw frequency-weighted variant, repeatedly purchased items exert stronger influence. In the time-decayed weighted variant, interactions that are both frequent and recent contribute more strongly to the learned embeddings.

LightGCN

LightGCN is a simplified graph collaborative filtering model designed specifically for recommendation tasks. Classical GCNs usually combine three operations in each layer: neighborhood aggregation, feature transformation, and nonlinear activation. However, in collaborative filtering, the household-item graph typically does not contain rich node attributes. Instead, households and items are mainly represented by trainable ID embeddings. Therefore, additional feature transformations and nonlinear activations may increase model complexity without necessarily improving recommendation performance. LightGCN addresses this issue by simplifying graph convolution for recommendation: it removes feature transformation matrices and nonlinear activation functions and retains only neighborhood aggregation over the user-item interaction graph [26,51].

In this study, LightGCN is applied to the same household-item bipartite graph variants defined earlier: binary, raw frequency-weighted, and time-decayed weighted graphs. These variants share the same LightGCN propagation mechanism and differ only in the edge weights used to construct the normalized adjacency matrix. In the binary variant, all observed household-item edges contribute equally. In the raw frequency-weighted variant, repeatedly purchased household-item pairs receive stronger influence. In the time-decayed weighted variant, purchase interactions that are both frequent and recent contribute more strongly during propagation, while older interactions remain in the graph with lower influence.

For all graph variants, the adjacency matrix is normalized using symmetric degree normalization. Let w_{hi}^{edge} denote the edge weight between household h and item i . In the binary graph, $w_{hi}^{edge} = 1$ for observed interactions, while in the raw frequency variant, w_{hi}^{edge} is the number of training basket-days in which household h purchased item i . Under the time-decayed variant, w_{hi}^{edge} is the accumulated exponentially decayed purchase evidence:

$$w_{hi}^{edge} = \sum_{t \in \mathcal{T}_{hi}} \exp(-\gamma \text{age}_{h,t}), \quad (35)$$

where $\mathcal{T}_{hi}$ denotes the set of training days on which household h purchased item i , γ is the decay rate, and $\text{age}_{h,t}$ is the age of the purchase event relative to the household's last observed training day.

The weighted degree of household h is defined as $d_h = \sum_i w_{hi}^{edge}$, and the weighted degree of item i is defined as $d_i = \sum_h w_{hi}^{edge}$. Therefore, the normalized adjacency entry used for propagation between household h and item i is proportional to:

$$\frac{w_{hi}^{edge}}{\sqrt{d_h d_i}} \quad (36)$$

This normalization means that larger edge weights increase the relative influence of a household-item connection during propagation, while degree normalization controls for highly connected households and popular items.

Starting from trainable household and item embeddings, LightGCN propagates embeddings directly over the normalized bipartite adjacency matrix:

$$E^{l+1} = \bar{A} E^l \quad (37)$$

where E^l denotes the household and item embeddings after layer l , and $\bar{A}$ is the symmetrically normalized household-item adjacency matrix constructed from the selected graph variant. Unlike standard GCN, LightGCN does not apply a learnable transformation matrix or nonlinear activation function at each layer.

In this implementation, LightGCN does not add explicit self-loops to the adjacency matrix. Instead, the initial embeddings are preserved through layer-wise averaging. The final embedding is obtained by averaging the initial embedding and the embeddings learned from all propagation layers:

$$E_{final} = \left(\frac{1}{L+1} \right) \sum_{l=0}^L E^l \quad (38)$$

where L is the number of propagation layers. This averaging combines information from the original trainable embeddings, direct household-item connections, and higher-order collaborative signals obtained through multi-layer propagation.

2.6.4. Proposed Repeat-Exploration-Aware Hybrid Models

The standard graph-based models described above generate recommendations through a single ranking function. Although this approach can capture collaborative filtering signals, it may produce recommendation lists dominated by repeat items, especially in grocery next-basket recommendation, where many purchases are habitual and recurring. To address this issue, this study proposes two repeat-exploration-aware hybrid models that explicitly separate repeat-oriented and exploration-oriented recommendations at inference time. Both hybrid models are built on a LightGCN encoder trained with uniform BPR on a household-item graph. The graph input can be constructed using the edge variants defined earlier. Thus, the selected edge type determines how household-item interactions influence graph propagation, while the time decay half-life is used when the time-decayed weighted graph variant is selected. After training, the models generate two candidate rankings: a repeat-oriented ranking and a history-filtered exploration ranking. These two rankings are finally combined through a household-specific quota-based interleaving strategy, allowing the final Top-K basket to preserve likely repurchases while reserving controlled space for new-to-household items.

Hybrid Model A: Two-Head LightGCN with History-Filtered Explore Head (TL-HFE)

Hybrid Model A (Figure 2), denoted as TL-HFE, is designed to reduce repeat-item dominance in grocery next-basket recommendation while preserving the predictive strength of graph-based collaborative filtering. Grocery baskets contain many habitual repurchases, so a single ranking function may mainly recommend items that the household has already purchased. Instead of using a single ranking list for all candidate items, TL-HFE first learns household and item embeddings using the LightGCN framework described earlier, with the selected household-item graph representation serving as the input structure. After training, the model separates recommendation generation into two inference-time heads. The first head is an unconstrained repeat-oriented ranking head that uses the learned LightGCN scores to preserve strong collaborative and habitual purchase signals. The second head is a history-filtered exploration head that ranks only items not previously purchased by the household, thereby creating a candidate list for new-to-household recommendations. These two ranked lists are then combined through a household-specific quota-based interleaving strategy, so that the final Top-K basket keeps likely repeat purchases while reserving controlled space for exploratory items. Therefore, TL-HFE does not introduce a separate encoder or additional training objective for repeat and

exploration; its main contribution is the two-head inference design built on the shared LightGCN embedding space.

Step 1. Construct the household-item graph input

TL-HFE first constructs the household-item bipartite graph using only the training purchase history. The detailed graph-construction procedure is defined in Section 2.6.3; therefore, this step does not repeat the full binary, raw frequency, and time-decayed edge definitions. TL-HFE is evaluated under the same graph-input variants defined earlier: binary, raw frequency-weighted, and time-decayed weighted edges. These variants do not change the two-head inference structure of TL-HFE. Instead, they only change the edge weights used to construct the normalized LightGCN propagation matrix.

In the time-decayed version of TL-HFE, each household-item edge represents the accumulated purchase strength between a household and an item, with larger contributions assigned to purchases that are more recent and/or repeated more often. Thus, the graph preserves long-term purchase history while giving stronger influence to interactions that better reflect the household's current preferences. Comparing the binary, raw frequency, and time-decayed variants allows the study to examine whether frequency and temporal recency improve the shared embedding space used by both the repeat-oriented and history-filtered exploration heads. The repeat and exploration distinction is introduced later, at inference time, through the two ranking heads.

Step 2. Learn shared household and item embeddings using LightGCN

After constructing the household-item graph input, TL-HFE trains a single LightGCN encoder to learn household and item embeddings. The encoder follows the LightGCN propagation procedure described in Section LightGCN; therefore, this step does not introduce a new graph architecture. Starting from trainable household and item ID embeddings, the model propagates information through the normalized household-item adjacency matrix. When the time-decayed graph is used, this propagation is guided by edge weights that reflect both purchase frequency and temporal recency.

The final household and item embeddings are obtained by combining the initial embeddings with the embeddings learned across the propagation layers. These final embeddings represent both direct household-item purchase history and higher-order collaborative signals from the graph. TL-HFE is trained using the BPR objective defined earlier, with uniform positive-item sampling from observed training interactions. Thus, edge weights affect the model through LightGCN message passing, rather than through weighted positive sampling during training.

Importantly, TL-HFE does not train separate encoders for repeat and exploration recommendation. Both the repeat-oriented head and the exploration head use the same shared LightGCN embedding space. The separation between repeat and exploration is introduced only after training, when the learned embeddings are used to generate two different inference-time rankings.

Step 3. Generate the repeat-oriented ranking head (Exploitation)

After training the shared LightGCN encoder, TL-HFE first generates a repeat-oriented candidate ranking. This head uses the final household and item embeddings learned in Step 2 and scores each candidate item using the standard LightGCN dot-product relevance score:

$$\hat{y}_{hi} = e_h^T e_i \quad (39)$$

where e_h denotes the final embedding of household h , and e_i denotes the final embedding of item i . Items are then ranked in descending order of this score:

$$R_h^{TTL-HFE} = \text{Rank}_{i \in I}(\hat{y}_{hi}) \quad (40)$$

This head is repeat-oriented because it preserves the exploitation behavior of the LightGCN model. However, it is not restricted only to items previously purchased by the household. Instead, it ranks the full learnable item catalog. In grocery recommendation, items that have been purchased frequently or recently by the household often receive high scores because their signals are strongly represented in the graph embeddings. Therefore, this ranking naturally tends to recover likely repurchases while still allowing any item in the training catalog to appear if its collaborative score is high. The resulting ranked list is used as the repeat-oriented candidate pool in the final interleaving step.

Step 4. Generate the history-filtered exploration ranking head (Discovery)

The second inference-time head in TL-HFE is the exploration head. This head is designed to produce a discovery-oriented candidate list by recommending items that have not appeared in the household's training history. It uses the same final household and item embeddings learned by the shared LightGCN encoder, and the relevance score is again computed using the dot product: $\hat{y}_{hi} = e_h^T e_i$.

The difference from the repeat-oriented head is the candidate space. Only items that are new to the household h are eligible for the exploration ranking. This definition focuses on household-level exploration rather than global item novelty, meaning that an item can be exploratory for a household even if it is popular in the overall dataset.

The exploration candidate list is generated by ranking only the eligible new-to-household items according to the LightGCN relevance score:

$$E_h^{TTL-HFE} = \text{Rank}_{i \in I_{h,t_h}^{expl}} (\hat{y}_{hi}) \quad (41)$$

where I_{h,t_h}^{expl} denotes the exploration candidate set for household h .

Thus, the exploration head applies history exclusion at inference time. Items that the household has already purchased are removed from this candidate list, while items that are new to the household remain eligible. The resulting ranked list is used as the exploration-oriented candidate pool in the final interleaving step.

Step 5. Estimate Household-Specific Repeat-Explore Quotas

After generating the repeat-oriented and exploration-oriented rankings, TL-HFE constructs the final Top-K recommendation list using a household-specific quota-based interleaving strategy. The purpose of this step is to personalize the repeat-explore balance rather than applying the same fixed proportion to all households.

Recent popularity-debiasing research has emphasized that recommendation exposure should not be controlled only at the aggregate level, but should instead be calibrated to user-specific behavioral tendencies. For example, SUPER separates items into popular and tail subsets, estimates each user's inclination toward popular content, and blends the final recommendation list according to this user-level exposure profile [41]. Similarly, TL-HFE uses household-specific purchase behavior to determine how much of the final basket should be allocated to exploratory recommendations.

To estimate this household-specific tendency, we compute a household-specific exploration rate from recent training behavior. Let W denote the recent observation window. In this study, W is treated as a validation-selected hyperparameter rather than a fixed universal value. For each household h , let I_h^{win} denote the set of unique items purchased during the most recent W -day training window, and let I_h^{past} denote the set of unique items purchased before that window. Items purchased in the recent window but not observed in the earlier history are treated as recently exploratory items; therefore, the household-specific exploration rate is then defined as:

$$r_h = \frac{|I_h^{win} \setminus I_h^{past}|}{|I_h^{win}|} = \frac{|I_h^{recent_explore}|}{|I_h^{win}|} \quad (42)$$

This value measures the extent to which household h has recently introduced new items into its basket. For example, if a household purchased four unique items in the recent window, $I_h^{win} = \{milk, bread, shrimp, avocado\}$, and two of these items, $\{milk, bread\}$, had appeared in its earlier history, then the recently exploratory set is $I_h^{explore} = \{shrimp, avocado\}$. In this case, $r_h = \frac{2}{4} = 0.5$, meaning that 50% of the household's recent basket items were new relative to its past behavior. A larger r_h indicates a more exploration-oriented household, while a smaller value indicates a more repeat-oriented household.

To prevent extreme allocations, the exploration rate is clipped within validation-selected lower and upper bounds:

$$\tilde{r}_h = clip(r_h, r_{min}, r_{max}) \quad (43)$$

where r_{min} and r_{max} are selected using the validation set. The clipped exploration rate is then converted into the number of exploration positions in the final Top-K recommendation list:

$$q_h^E = round(K\tilde{r}_h) \quad (44)$$

The remaining positions are assigned to the repeat-oriented ranking:

$$q_h^R = K - q_h^E \quad (45)$$

For example, if $K = 10$ and $r_h = 0.3$, then, $\tilde{r}_h = 0.3$, $q_h^E = round(10 \times 0.3) = 3$, and $q_h^R = 10 - 3 = 7$. Thus, three positions are allocated to exploratory items and seven positions are allocated to repeat-oriented items. The exact values of W , r_{min} and r_{max} are not fixed manually; they are selected during validation and reported in the hyperparameter-tuning results.

Step 6. Generate the final Top-K basket through quota-based interleaving

The final Top-K recommendation list is constructed by interleaving items from the repeat and exploration rankings according to the household-specific quotas defined in Step 5. This step is applied only at inference time and does not introduce an additional learning objective. The interleaving module preserves the internal ranking order of each candidate list while controlling how many items are selected from each head.

Specifically, q_h^E positions are allocated to the exploration ranking, while q_h^R positions are allocated to the repeat ranking. During merging, duplicate items are removed so that each item appears at most once in the final recommendation list. If one ranking does not contain enough eligible candidates to satisfy its quota, the remaining positions are backfilled from the other ranking. This ensures that the final output contains K recommended items whenever enough candidates are available.

Formally, the final TL-HFE recommendation list for household h is denoted as:

$$\hat{P}_h^K = Interleave(R_h^{TTL-HFE}, E_h^{TTL-HFE}; q_h^R, q_h^E), \quad (46)$$

where $R_h^{TTL-HFE}$ is the repeat-oriented ranking, $E_h^{TTL-HFE}$ is the exploration-oriented ranking, q_h^R is the repeat quota, and q_h^E is the exploration quota.

This quota-based interleaving strategy allows TL-HFE to preserve strong repeat-oriented recommendations while reserving controlled space for new-to-household items. As a result, the final recommendation list reduces repeat dominance without discarding the collaborative signals learned by the shared LightGCN encoder. The final predicted basket $\hat{P}_h^K$ is then evaluated using the overall, repeat, and explore metrics defined earlier.

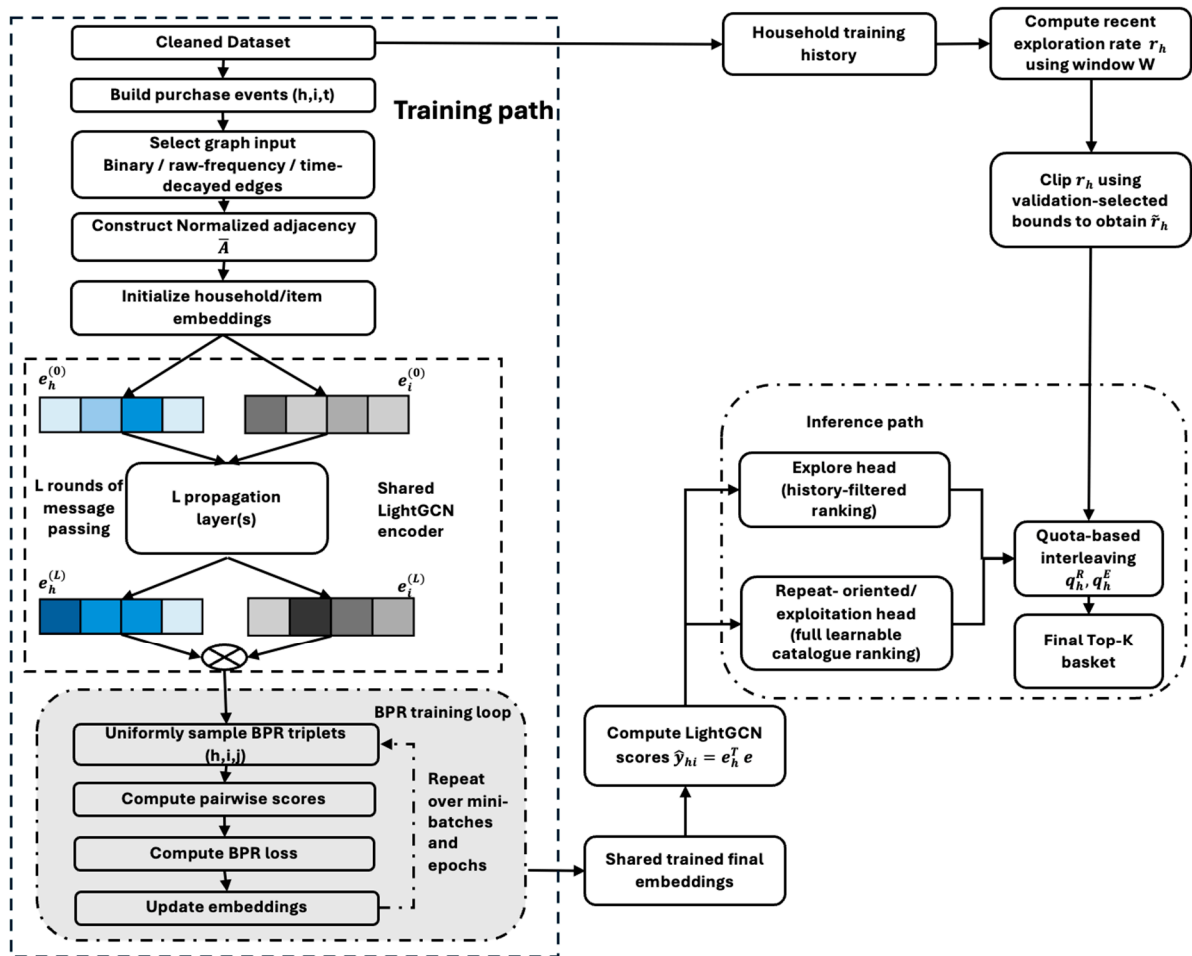


Figure 2. Hybrid Model A: Two-Head LightGCN with History-Filtered Explore Head (TL-HFE).

Hybrid Model B: Two-Head LightGCN with Personal-Popularity Repeat Head (TL-PPR)

Hybrid Model B (Figure 3), denoted as TL-PPR, is the second repeat-exploration-aware hybrid model proposed in this study. TL-PPR follows the same overall framework as TL-HFE: it trains a shared LightGCN encoder on the household-item graph, generates a history-filtered LightGCN exploration ranking, estimates household-specific repeat-explore quotas, and constructs the final Top-K recommendation list through quota-based interleaving. Therefore, this section focuses only on the main difference between TL-HFE and TL-PPR: the construction of the repeat-oriented head.

In TL-HFE, the repeat-oriented ranking is generated using LightGCN relevance scores over the item catalog. In contrast, TL-PPR replaces this graph-based repeat head with a non-parametric personal-popularity repeat head. This design is motivated by the repetitive nature of grocery shopping, where many next-basket purchases are driven by a household’s own historical purchase behavior. Instead of using the learned LightGCN score for repeat recommendation, TL-PPR directly ranks previously purchased items according to household-level purchase frequency.

Let I_{h,t_h}^{rep} denote the set of items previously purchased by household h in the training period. TL-PPR restricts the repeat candidate set to these historical items only. For each repeat candidate $i \in I_{h,t_h}^{rep}$, the personal-popularity repeat score is defined as:

$$s_{hi}^{PPR} = \sum_{t=1}^{t_h} 1(i \in B_h^t), \quad (47)$$

where s_{hi}^{PPR} denote the number of training basket-days in which household h purchased item i , and $1 (i \in B_h^t)$ is an indicator function that equals 1 if item i appears in basket B_h^t , and 0 otherwise. The count is computed at the household-day level, so an item contributes at most once per household basket-day.

The repeat-oriented ranking is then obtained by sorting the household's historical items according to this personal-popularity score:

$$R_h^{TL-PPR} = \text{Rank}_{i \in I_{h,t_h}^{rep}} (s_{hi}^{PPR}) \quad (48)$$

Items with higher household-level purchase frequency are ranked higher. In the case of ties, global raw frequency is used only as a tie-breaking criterion among items already purchased by the household. It does not introduce new items into the repeat list. Thus, the repeat head remains a pure repeat-oriented component.

Importantly, the TL-PPR repeat head is independent of the selected LightGCN graph edge type. The graph input used by the shared LightGCN encoder may be binary, raw frequency-weighted, or time-decayed weighted, and these edge variants affect the learned embeddings and the history-filtered exploration ranking. However, the personal-popularity repeat head itself is fixed as a raw frequency household-level ranking. Therefore, time-decayed edge weights influence TL-PPR through the LightGCN representation and exploration head, but not directly through the repeat head score.

This repeat head is non-parametric because it does not introduce additional trainable parameters, a separate encoder, or an auxiliary loss function. Instead, it directly ranks likely repurchases using household-level frequency and recency signals. This makes TL-PPR different from TL-HFE: TL-HFE uses the LightGCN score for the repeat-oriented ranking, whereas TL-PPR uses household-specific personal-popularity evidence for the repeat ranking.

The exploration ranking in TL-PPR is the same history-filtered LightGCN exploration ranking defined for TL-HFE. The trained LightGCN embeddings are used to score candidate items, while items already purchased by the household are excluded from the exploration candidate set. The final TL-PPR recommendation list is then produced using the same household-specific quota-based interleaving strategy:

$$\hat{P}_h^K = \text{Interleave} (R_h^{TL-PPR}, E_h^{TL-HFE}; q_h^R, q_h^E), \quad (49)$$

where R_h^{TL-PPR} is the personal-popularity repeat ranking, E_h^{TL-HFE} denotes the shared history-filtered LightGCN exploration ranking, and q_h^R and q_h^E are the household-specific repeat and exploration quotas, respectively.

Overall, TL-PPR separates the source of repeat and exploration evidence. Repeat recommendations are generated from direct household-level purchase frequency, while exploration recommendations are generated from the shared LightGCN embedding space. This allows the model to preserve strong habitual repurchase signals while reserving controlled space for new-to-household items.

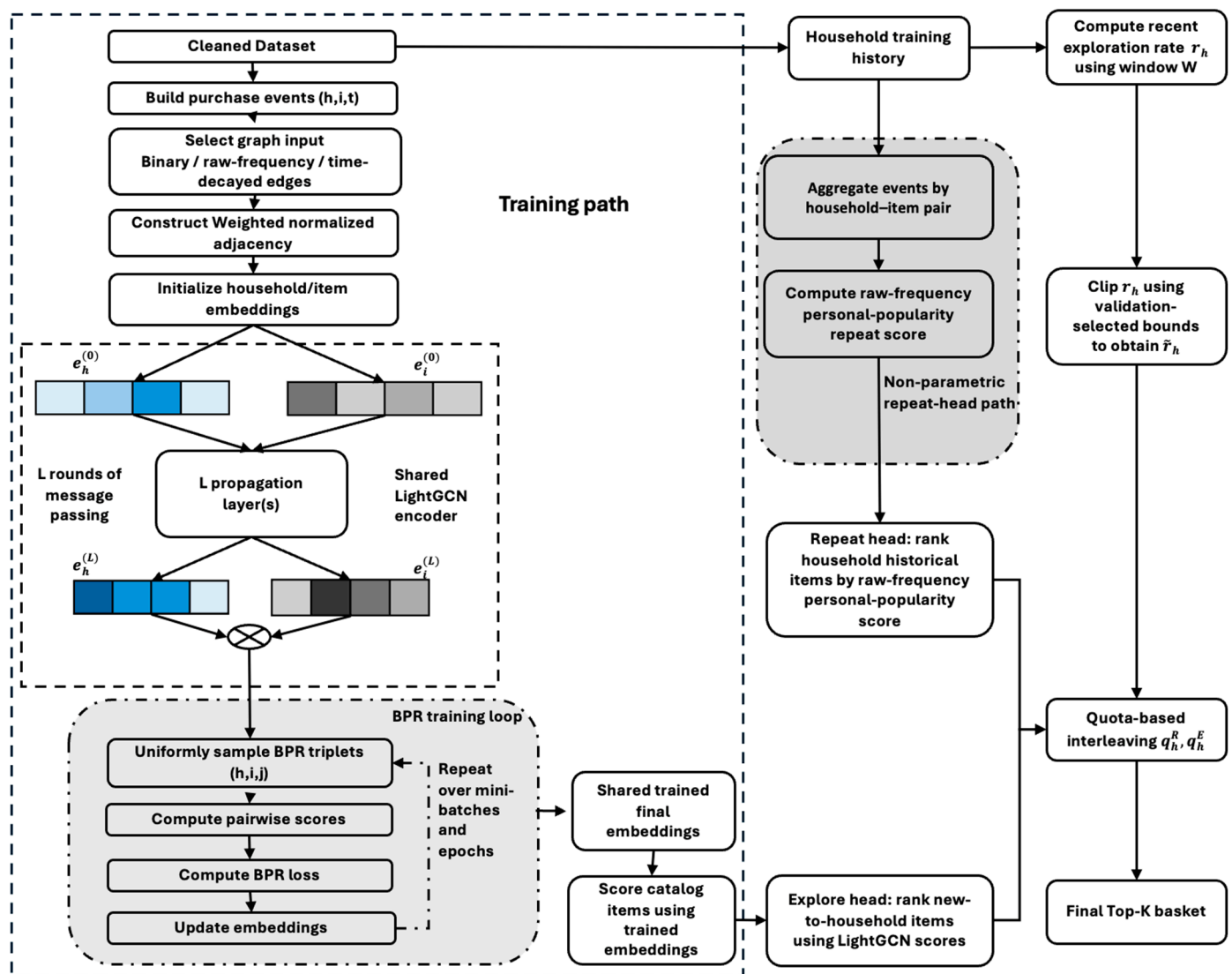


Figure 3. Hybrid Model B: Two-Head LightGCN with Personal-Popularity Repeat Head (TL-PPR).

2.7. Composite Validation Score for Model Selection

Because grocery next-basket recommendation involves a trade-off between routine repurchase accuracy and exploratory recommendation, model selection should not be based only on Overall Recall. Prior NBR research shows that aggregate accuracy can be dominated by repeat items, while performance on exploratory items may remain weak [7]. Therefore, in addition to reporting overall, repeat-specific, and explore-specific metrics separately, this study uses a composite validation score to select the final model configuration.

The composite score is not introduced as a new recommendation metric or as a training objective. Rather, it is a validation-based model-selection criterion used to compare candidate configurations under the repeat-explore objective of this study. This follows the general multi-criteria decision-making idea that alternatives can be ranked according to multiple criteria with explicit importance weights [52,53]. In this setting, the alternatives are candidate configurations, and the criteria are normalized Overall Recall@20, Repeat Recall@20, and Explore Recall@20.

Selecting candidate configurations using only one validation metric would bias model selection toward one side of the repeat-explore trade-off. For example, prioritizing Overall Recall or Repeat Recall alone may favor configurations that mainly reproduce frequently purchased items, whereas prioritizing Explore Recall alone may favor configura-

tions with stronger exploratory behavior but weaker repeat-purchase accuracy. Therefore, the composite validation score provides a transparent way to balance these competing objectives during hyperparameter selection.

For each candidate configuration C , validation performance is measured using Overall Recall@20, Repeat Recall@20, and Explore Recall@20. To place the three metrics on a comparable scale, each metric is normalized by the best validation value achieved by any candidate configuration:

$$\tilde{R}_{m,C} = \frac{R_{m,C}}{\max_{C' \in \mathcal{C}} R_{m,C'} + \epsilon}, m \in \{Overall, Repeat, Explore\} \quad (50)$$

where $\mathcal{C}$ is the set of validation candidates and ϵ is a small constant used to avoid division by zero. The validation composite score of candidate C is then defined as:

$$U(C) = \alpha \tilde{R}_{Overall,C} + \beta \tilde{R}_{Repeat,C} + \gamma \tilde{R}_{Explore,C} \quad (51)$$

where $\alpha + \beta + \gamma = 1$. The main model-selection profile is explore-aware, with weights $\alpha = 0.35$, $\beta = 0.25$, and $\gamma = 0.40$. This profile gives the largest weight to Explore Recall while still requiring the selected model to preserve competitive overall and repeat performance. The weights are not claimed to be universal. They operationalize the selection preference of this study: improving exploratory recall while still preserving overall and repeat-item accuracy.

To avoid hiding trade-offs, the composite score is used only to select among candidate configurations on the validation set. After the best configuration is selected on the validation group, it is frozen and evaluated once on the disjoint test group at $K = 10, 20$, and 30 .

3. Hyperparameter Tuning

As described earlier, this study adopts a day-level leave-last-day-out evaluation protocol. For each household, the final observed shopping day is held out as the target next basket, while all previous shopping days are used to construct the training history and, where applicable, the household-product interaction graph. Following [5,9], and to support hyperparameter tuning without contaminating final test performance, the held-out households are divided into two disjoint groups: 50% are assigned to the validation set, and the remaining 50% are assigned to the test set. The validation households' final baskets are used only for model selection and hyperparameter tuning, whereas the test households' final baskets are reserved for final performance reporting. Importantly, no final-day basket is included in training.

For each trainable model, a predefined hyperparameter search space is evaluated on the validation set (Table 4). The deterministic baselines, Global Top-Frequency and Personal Top-Frequency, do not require hyperparameter tuning because their scores are directly computed from training-set purchase frequencies. For graph-based models, the validation grid includes alternative edge representations: binary, raw frequency, and time-decayed edges. When time-decayed edges are used, the half-life parameter is selected from 30, 60, and 90 days. The architectural search space includes embedding dimensions of 64 and 128 and LightGCN propagation depths of one, two, and three layers. For the hybrid models, the validation grid also includes recent-history windows and quota-clipping ranges, which control the repeat-explore allocation.

Model selection is performed using the explore-aware composite validation score defined in Section 2.7. This score jointly considers Overall Recall@20, Repeat Recall@20, and Explore Recall@20, rather than selecting the configuration that maximizes only one metric. After the best configuration for each model is selected on the validation households, all hyperparameters are fixed and the selected configuration is evaluated once on the held-

out test households at $K = 10, 20$, and 30 . This procedure ensures that model selection and final performance reporting are conducted on disjoint household groups.

For both Dunnhumby and Ta-Feng, the selected TL-PPR configuration uses time-decayed graph edges with a 30-day half-life. This indicates that recent purchase behavior is more informative than older interactions and that combining frequency with recency is more effective than using binary or raw frequency graph edges alone. Both datasets also select a 90-day recent-history window, suggesting that the quota mechanism benefits from a broader recent-history view, while the graph edge weights themselves still rely more strongly on short-term temporal decay.

The main dataset-specific differences appear in the quota-clipping range and LightGCN architecture. For Dunnhumby, TL-PPR selects a clipping range of 0.10 – 0.50 , 64-dimensional embeddings, and one LightGCN propagation layer. This suggests that a relatively shallow graph representation is sufficient and that a slightly wider exploration allowance is useful for maintaining the repeat-explore balance. For Ta-Feng, TL-PPR selects a more conservative clipping range of 0.05 – 0.40 , 128-dimensional embeddings, and three LightGCN layers. This indicates that Ta-Feng benefits from a stronger graph-based exploration component, while still preserving repeat accuracy through a tighter quota range.

Overall, the validation-grid results show that TL-PPR consistently favors time-decayed graph construction and short temporal half-life across both datasets, confirming the importance of recency in grocery next-basket recommendation. At the same time, the selected quota and architecture differ by dataset, reflecting differences in repeat-purchase strength and graph structure. Thus, the selected TL-PPR configurations are not arbitrary; they represent validation-based trade-offs between overall accuracy, repeat recovery, and exploratory recommendation.

Table 4. Hyperparameter tuning.

Model-Evaluated Configuration		Selected Configuration— Dunnhumby	Selected Configuration—Ta-Feng
Graph SAGE	hidden dimension: 64, 128; learning rate: 0.001, 0.0005; regularization: 0.0001, 0.00001; epochs: 5; steps per epoch: 50; batch size: 512.	Time-decayed edges; half-life = 90; embedding dimension = 64; hidden dimension = 128; layers = 2; learning rate = 0.0005; regularization = 0.00001.	Raw frequency edges; embedding dimension = 64; hidden dimension = 64; layers = 1; learning rate = 0.001; regularization = 0.0001.
GCN	hidden dimension: 64, 128; self-loop weight = 0.1; learning rate: 0.001, 0.0005; regularization: 0.0001, 0.00001; epochs: 5; steps per epoch: 200; batch size: 512.	Time-decayed edges; half-life = 60; embedding dimension = 128; hidden dimension = 64; layers = 3; learning rate = 0.001; regularization = 0.0001.	Time-decayed edges; half-life = 60; embedding dimension = 128; hidden dimension = 64; layers = 1; self-loop weight = 0.1; learning rate = 0.0005; regularization = 0.0001.
LightGCN	learning rate: 0.001, 0.0005; regularization: 0.0001, 0.00001; epochs: 5; training steps: 50; batch size: 4096.	Time-decayed edges; half-life = 30; embedding dimension = 128; layers = 1; learning rate = 0.001; regularization = 0.00001.	Time-decayed edges; half-life = 30; embedding dimension = 128; layers = 1; learning rate = 0.001; regularization = 0.00001.
TL-HFE	recent-history window: 30, 60, 90 days; quota clipping bounds: (0.05, 0.40), (0.10, 0.50), (0.05, 0.60); epochs: 5; training steps: 50; batch size: 4096; learning rate = 0.001; regularization = 0.0001.	Time-decayed edges; half-life = 30; recent-history window = 90; clipping bounds = (0.10, 0.50); embedding dimension = 64; layers = 1.	Time-decayed edges; half-life = 30; recent-history window = 90; clipping bounds = (0.05, 0.40); embedding dimension = 128; layers = 1.
TL-PPR	recent-history window: 30, 60, 90 days; quota clipping bounds: (0.05, 0.40), (0.10, 0.50), (0.05, 0.60); epochs:	Time-decayed edges; half-life = 30; recent-history window = 90; clipping	Time-decayed edges; half-life = 30; recent-history window = 90; clip-

5; training steps: 50; batch size: 4096; bounds = (0.1, 0.5); embedding dimension = 0.001; regularization = mension = 64; layers = 1. 0.0001.

ping bounds = (0.05, 0.40); embedding dimension = 128; layers = 3.

4. Results

All experiments were implemented in Python 3.12.13 and executed in a Google Colab/Jupyter Notebook environment on a Linux platform using an NVIDIA A100-SXM4-40GB GPU with CUDA 12.8. The main software libraries used for data preprocessing, graph construction, model training, and evaluation included NumPy 2.0.2, Pandas 2.2.2, SciPy 1.16.3, Scikit-learn 1.6.1, and PyTorch 2.11.0 + cu128. To improve reproducibility, random seeds were fixed for data splitting, negative sampling, model initialization, and training procedures.

Table 5 reports the performance of the baseline and proposed models on the Dunnhumby and Ta-Feng datasets. Bold values in the Table 5 indicate the best performance for each evaluation criterion. Across both datasets, PHR@K and Recall@K generally increase as K becomes larger, because longer recommendation lists provide more opportunities to include relevant held-out products. The results also confirm that grocery next-basket recommendation is strongly repeat-driven: models that rely directly on household-level purchase history achieve strong repeat performance, while exploration remains more difficult.

Table 5. Model performance across Top-K recommendation list sizes. (For GraphSAGE, GCN, LightGCN, TL-HFE, and TL-PPR, the reported results correspond to the validation-selected configurations shown in Table 4.)

Dataset	Models	K	Overall			Repeat			Explore		
			PHR@K	Recall@K	NDCG@K	PHR@K	Recall@K	NDCG@K	PHR@K	Recall@K	NDCG@K
Dunnhumby	Model 1: Global Top-Frequency	10	0.37	0.04	0.08	0.40	0.09	0.10	0.06	0.01	0.01
		20	0.41	0.05	0.07	0.44	0.11	0.10	0.09	0.01	0.01
		30	0.50	0.08	0.08	0.54	0.15	0.12	0.13	0.02	0.01
	Model 2: Personal Top-Frequency	10	0.58	0.14	0.20	0.69	0.28	0.28	0.00	0.00	0.00
		20	0.66	0.20	0.19	0.78	0.40	0.30	0.01	0.00	0.00
		30	0.70	0.23	0.20	0.82	0.47	0.33	0.01	0.00	0.00
	Model 3: GraphSAGE	10	0.32	0.04	0.06	0.35	0.07	0.07	0.05	0.01	0.01
		20	0.45	0.06	0.06	0.48	0.12	0.09	0.10	0.02	0.01
		30	0.48	0.08	0.06	0.52	0.14	0.10	0.13	0.02	0.01
	Model 4: Weighted Graph Convolutional Network (GCN)	10	0.43	0.06	0.09	0.48	0.12	0.13	0.06	0.01	0.01
		20	0.50	0.08	0.09	0.55	0.16	0.13	0.11	0.02	0.01
		30	0.54	0.10	0.09	0.60	0.19	0.14	0.15	0.03	0.02
	Model 5: LightGCN	10	0.43	0.06	0.10	0.48	0.12	0.13	0.06	0.01	0.01
		20	0.50	0.09	0.09	0.55	0.16	0.14	0.12	0.02	0.01
		30	0.54	0.11	0.10	0.60	0.20	0.15	0.15	0.03	0.02
	Model 6: TL-HFE	10	0.44	0.06	0.07	0.49	0.13	0.10	0.06	0.01	0.01
		20	0.51	0.09	0.08	0.57	0.19	0.11	0.11	0.02	0.01
		30	0.56	0.12	0.08	0.63	0.23	0.13	0.14	0.02	0.01
	Model 7: TL-PPR	10	0.52	0.10	0.12	0.60	0.20	0.16	0.06	0.01	0.01
		20	0.60	0.15	0.12	0.69	0.29	0.18	0.11	0.02	0.01
		30	0.65	0.19	0.13	0.74	0.35	0.20	0.14	0.02	0.01
Ta-Feng	Model 1: Global Top-Frequency	10	0.23	0.07	0.08	0.18	0.12	0.10	0.16	0.05	0.05
		20	0.33	0.11	0.09	0.25	0.16	0.11	0.25	0.08	0.06
		30	0.39	0.12	0.10	0.29	0.18	0.12	0.30	0.10	0.07
		10	0.41	0.13	0.12	0.77	0.62	0.41	0.02	0.01	0.01
		20	0.51	0.18	0.13	0.90	0.80	0.47	0.07	0.03	0.01

Model 2: Personal Top-Frequency	30	0.57	0.21	0.14	0.95	0.89	0.49	0.13	0.05	0.02
Model 3: GraphSAGE	10	0.22	0.07	0.08	0.17	0.11	0.10	0.16	0.05	0.05
	20	0.32	0.11	0.09	0.22	0.15	0.11	0.24	0.08	0.06
	30	0.36	0.12	0.09	0.25	0.16	0.11	0.28	0.10	0.07
Model 4: Graph Convolutional Network (GCN)	10	0.23	0.07	0.08	0.18	0.12	0.10	0.16	0.05	0.05
	20	0.33	0.11	0.09	0.26	0.16	0.11	0.25	0.08	0.06
	30	0.39	0.12	0.10	0.30	0.19	0.12	0.29	0.10	0.07
Model 5: LightGCN	10	0.26	0.09	0.09	0.22	0.16	0.12	0.17	0.06	0.05
	20	0.33	0.12	0.09	0.29	0.21	0.13	0.23	0.08	0.06
	30	0.39	0.14	0.10	0.35	0.25	0.14	0.28	0.10	0.06
Model 6: TL- HFE	10	0.38	0.13	0.10	0.54	0.39	0.20	0.14	0.05	0.05
	20	0.45	0.15	0.11	0.63	0.47	0.22	0.20	0.07	0.06
	30	0.49	0.17	0.11	0.65	0.49	0.23	0.25	0.09	0.06
Model 7: TL- PPR	10	0.42	0.14	0.12	0.65	0.49	0.25	0.14	0.05	0.05
	20	0.52	0.19	0.13	0.81	0.67	0.30	0.20	0.07	0.06
	30	0.58	0.21	0.14	0.89	0.78	0.33	0.23	0.08	0.06

4.1. Dunnhumby Results

For the Dunnhumby dataset, Personal Top-Frequency is the strongest model in terms of overall and repeat accuracy. At $K = 20$, it achieves Overall Recall@20 = 0.20 and Repeat Recall@20 = 0.40, outperforming all graph-based and hybrid models. However, its Explore Recall@20 is approximately zero, showing that its strong performance is almost entirely driven by repeated products.

The graph-based models provide more balanced but lower overall performance. GraphSAGE, GCN, and LightGCN achieve Overall Recall@20 values between 0.06 and 0.09, while their Explore Recall@20 values are around 0.02. This indicates that graph-based collaborative signals can recover some exploratory items, but they do not match the repeat-recovery strength of the personal-frequency baseline.

The proposed hybrid models improve over the single-head graph models in overall and repeat recommendations. TL-HFE obtains Overall Recall@20 = 0.09 and Repeat Recall@20 = 0.19, while TL-PPR further improves these values to Overall Recall@20 = 0.15 and Repeat Recall@20 = 0.29. Nevertheless, TL-PPR does not surpass Personal Top-Frequency on Dunnhumby. Its main advantage is that it combines stronger repeat recovery than graph-only models with non-zero exploration, whereas Personal Top-Frequency remains almost purely repeat-oriented. Therefore, on Dunnhumby, TL-PPR should be interpreted as a competitive hybrid alternative rather than the best-performing model overall.

4.2. Ta-Feng Results

For the Ta-Feng dataset, repeat behavior is even stronger. Personal Top-Frequency achieves very high repeat performance, with Repeat Recall@20 = 0.80 and Repeat NDCG@20 = 0.47. However, its Explore Recall@20 remains low at 0.03, confirming that personal-frequency recommendation mainly captures repeated purchases and has limited ability to recommend new-to-customer products.

Among the graph-based baselines, GraphSAGE, GCN, and LightGCN show similar overall performance. At $K = 20$, their Overall Recall@20 ranges from 0.11 to 0.12, while Explore Recall@20 is around 0.08. These models perform better than Personal Top-Frequency on exploration, but they are much weaker on repeat recommendation.

The proposed hybrid models improve overall and repeat performance compared with the graph-only models. TL-HFE achieves Overall Recall@20 = 0.15 and Repeat Recall@20 = 0.47. TL-PPR performs best in terms of Overall PHR@20 and Overall Recall@20,

reaching Overall PHR@20 = 0.52 and Overall Recall@20 = 0.19. This slightly exceeds Personal Top-Frequency, which obtains Overall PHR@20 = 0.51 and Overall Recall@20 = 0.18. TL-PPR also achieves strong repeat performance, with Repeat Recall@20 = 0.67, although it remains below Personal Top-Frequency.

For exploration, TL-PPR does not dominate the graph-based baselines. Its Explore Recall@20 = 0.07, while Global Top-Frequency, GraphSAGE, GCN, and LightGCN reach approximately 0.08. This means that TL-PPR's advantage on Ta-Feng comes mainly from combining strong repeat recovery with acceptable exploration, not from achieving the highest exploration accuracy. Therefore, TL-PPR is best described as the strongest overall hybrid model on Ta-Feng, but not as the best pure exploration model.

4.3. Cross-Dataset Interpretation of PHR, NDCG, and the Repeat-Explore Trade-Off

The PHR and NDCG results provide additional insight beyond Recall. PHR@K shows whether a model retrieves at least one relevant product for a household, while NDCG@K reflects whether relevant products are ranked near the top of the recommendation list. In both datasets, PHR generally increases with K, as longer lists make it more likely that at least one held-out product is recommended. However, NDCG changes less sharply because it rewards correct items more when they appear in higher-ranked positions. On Dunnhumby, Personal Top-Frequency achieves the strongest Overall and Repeat PHR/NDCG values, confirming that it ranks habitual repurchases very effectively, but its Explore PHR and Explore NDCG remain close to zero. On Ta-Feng, TL-PPR achieves the highest Overall PHR@20 and matches or slightly exceeds the strongest overall Recall performance, while Personal Top-Frequency still dominates Repeat NDCG. Across both datasets, explore-specific NDCG values remain low, indicating that exploratory products are not only harder to retrieve but also harder to place near the top of the ranked list. These patterns reinforce the main conclusion that grocery NBR performance depends on both hit coverage and ranking quality, and that repeat-oriented and exploration-oriented behavior should be interpreted separately.

Overall, the results show a clear repeat-explore trade-off. Personal Top-Frequency is the strongest repeat-oriented baseline, especially on Dunnhumby, but it provides very limited exploratory recommendations. In contrast, graph-based models achieve better exploration but are weaker at recovering repeat purchases. The proposed hybrid models, especially TL-PPR, reduce this gap by combining a repeat-oriented personal-popularity head with a graph-based exploration component. Although TL-PPR does not always achieve the highest Explore Recall, its exploratory performance remains close to the graph-based baselines, while its Overall Recall is substantially higher. For example, on Ta-Feng at K = 20, TL-PPR obtains slightly lower Explore Recall than the graph-based baselines, but achieves higher Overall Recall by substantially improving repeat-item recovery. Therefore, the main advantage of TL-PPR is not maximizing pure exploration, but providing a more practical repeat-explore compromise in which overall next-basket accuracy improves while exploratory performance remains competitive.

5. Sensitivity Analysis

Table 6 reports the validation-based sensitivity analysis for the TL-PPR model on the Dunnhumby and Ta-Feng datasets. The reported values are grouped averages over the TL-PPR validation grid. For each design factor, including quota-clipping range, graph edge type, temporal half-life, and recent-history window, all configurations sharing the same factor level were grouped together, and the mean Overall Recall@20, Repeat Recall@20, Explore Recall@20, and Explore-aware Composite Validation Score@20 were computed. Therefore, these values should be interpreted as validation-level sensitivity patterns rather than final test-set results or one-factor-at-a-time controlled experiments.

Since TL-PPR uses a fixed raw frequency personal-popularity repeat head, changes in graph edge construction mainly affect the LightGCN-based exploration head. The sensitivity results are reported as marginal averages over the validation grid. Therefore, the best-performing marginal factor level does not necessarily have to match the factor level selected in the best joint configuration.

Table 6. Sensitivity analysis.

Dataset	Hyperparameter	Setting	Overall Recall@20	Repeat Recall@20	Explore Recall@20	Explore-Aware Composite Validation Score@20
Dunhumby	Clip range	[0.05, 0.40]	0.1606	0.3021	0.0137	0.8422
		[0.10, 0.50]	0.1526	0.2819	0.0159	0.8486
		[0.05, 0.60]	0.1476	0.2665	0.0173	0.8496
	Edge type	Time-decayed	0.1539	0.2835	0.0163	0.8586
		Binary	0.1535	0.2835	0.0153	0.8395
		Raw frequency	0.1529	0.2835	0.0142	0.8188
	Half-life	30 days	0.1545	0.2835	0.0174	0.8801
		60 days	0.1537	0.2835	0.0161	0.8544
		90 days	0.1534	0.2835	0.0154	0.8412
	W last days	30 days	0.1562	0.2877	0.0153	0.8491
		60 days	0.1528	0.2827	0.0156	0.8444
		90 days	0.1517	0.2802	0.016	0.847
Ta-Feng	Clip range	[0.05, 0.40]	0.1906	0.6678	0.0686	0.9357
		[0.10, 0.50]	0.1832	0.6179	0.071	0.9158
		[0.05, 0.60]	0.1747	0.5584	0.0735	0.891
	Edge type	Time-decayed	0.1836	0.6147	0.0719	0.9201
		Binary	0.1815	0.6147	0.0695	0.9044
		Raw frequency	0.1818	0.6147	0.0698	0.9064
	Half-life	30 days	0.1848	0.6147	0.0735	0.9302
		60 days	0.1832	0.6147	0.0714	0.9166
		90 days	0.1828	0.6147	0.0709	0.9134
	W last days	30 days	0.1829	0.6153	0.0709	0.9141
		60 days	0.1828	0.6144	0.0711	0.9142
		90 days	0.1828	0.6144	0.0711	0.9142

5.1. Interpretation for Dunhumby

For quota clipping, the results show a clear repeat-explore trade-off. The narrowest range, (0.05, 0.40), gives the highest Overall Recall@20 and Repeat Recall@20, with values of 0.1606 and 0.3021, respectively. However, it has the lowest Explore Recall@20, equal to 0.0137. As the clipping range becomes wider, repeat-oriented performance decreases while exploration improves. The widest range, (0.05, 0.60), gives the highest Explore Recall@20, 0.0173, and the highest mean Explore-aware Composite Validation Score@20, 0.8496. This indicates that allowing more flexible household-specific exploration improves the ability of TL-PPR to recommend previously unseen items, although this comes at the cost of lower repeat recall.

For edge type, time-decayed graph edges perform best. They achieve the highest Explore Recall@20, 0.0163, and the highest Explore-aware Composite Validation Score@20, 0.8586. Binary edges perform slightly lower, while raw frequency edges obtain the weakest exploration performance and composite score. This supports the role of time-decayed graph construction in improving the LightGCN exploration head.

For half-life, the best value is 30 days. It gives the highest Overall Recall@20, Explore Recall@20, and Explore-aware Composite Validation Score@20, with values of 0.1545,

0.0174, and 0.8801, respectively. Longer half-lives of 60 and 90 days reduce exploration performance and the composite score. This suggests that recent purchase behavior is more informative than longer-term interaction history for the time-decayed exploration graph.

For the recent-history window, $W = 30$ gives the highest Overall Recall@20 and Repeat Recall@20, whereas $W = 90$ gives the highest Explore Recall@20. The differences are moderate, but the pattern suggests that shorter windows favor repeat accuracy, while longer windows provide slightly more support for exploration. Overall, the Dunnhumby results show that TL-PPR benefits from time-decayed edges, short temporal half-life, and a moderately flexible repeat-explore allocation mechanism.

5.2. Interpretation for Ta-Feng

The narrowest range for quota clipping, (0.05, 0.40), gives the highest Overall Recall@20 and Repeat Recall@20, with values of 0.1906 and 0.6678. However, the widest range, (0.05, 0.60), gives the highest Explore Recall@20, equal to 0.0735. This means that increasing the allowed exploration quota improves exploration, but it reduces repeat and overall accuracy. For Ta-Feng, the best mean Composite Validation Score is still achieved by (0.05, 0.40), so the model benefits from a more repeat-conservative allocation.

For edge type, the time-decayed graph performs best. It gives the highest Overall Recall@20, Explore Recall@20, and Composite Validation Score @20. The differences are modest, but the ordering is consistent: time-decayed edges achieve the highest Overall Recall@20, Explore Recall@20, and explore-aware utility, followed by raw frequency and binary edges. This supports our argument that time-decayed LightGCN edges are useful for the exploration head.

For half-life, the best value is 30 days. It gives the highest Overall Recall@20, Explore Recall@20, and explore-aware Composite Validation Score. Longer half-lives, 60 and 90 days, slightly reduce performance. This suggests that for Ta-Feng, recent purchase behavior is more useful than longer-term history in the time-decayed graph.

For W last days, the differences are very small. $W = 30$ gives slightly higher repeat recall, while $W = 60$ and $W = 90$ give slightly higher explore recall and Composite Validation Score. This means the recent-history window is less sensitive than clip range, edge type, or half-life.

6. Ablation Study

To examine the contribution of the main TL-PPR components, ablation studies were conducted on the validation splits of both the Dunnhumby and Ta-Feng datasets. Each ablation removes or replaces one component while keeping the selected full configuration fixed. The analysis focuses on the personal-popularity repeat head, LightGCN exploration head, history filter, and repeat-explore quota mechanism (Table 7).

Bold values in Table 7 indicate the best performance for each evaluation criterion in each dataset. The results show that the repeat and exploration heads play complementary roles. In both datasets, the repeat-only variant achieves the highest Repeat Recall@20 but produces zero Explore Recall@20, while the explore-only variant improves Explore Recall@20 but loses all repeat performance. This confirms that TL-PPR is not intended to maximize only one behavior; it is designed to balance repeat and exploration recommendations.

The time-decayed graph consistently improves the exploration component. In Dunnhumby, Explore Recall@20 decreases from 0.0210 to 0.0172 with raw frequency edges and to 0.0162 with binary edges. Similarly, in Ta-Feng, it decreases from 0.0741 to 0.0688 and 0.0687. Repeat Recall@20 remains unchanged in these edge ablations because the repeat head is fixed as raw frequency personal popularity and does not depend on the LightGCN edge type.

Replacing the LightGCN exploration head with global-popularity exploration reduces Explore Recall@20 in both datasets, showing that the graph-based exploration head contributes beyond recommending popular unseen items. The history filter is also important: removing it reduces Overall Recall@20 in both datasets, especially in Dunnhumby, where Explore Recall@20 drops from 0.0210 to 0.0131. In contrast, the fixed global quota gives results very close to the household-specific quota, suggesting that the quota mechanism is not the main source of performance variation in these validation settings.

Overall, the ablation study shows that TL-PPR's main contribution comes from combining a strong personal-popularity repeat head with a time-decayed LightGCN exploration head. The effect is especially clear for exploration, while the dataset-specific difference is that Dunnhumby remains more repeat-dominant: the repeat-only variant achieves higher Overall Recall@20 than the full model, indicating that TL-PPR sacrifices some repeat accuracy to improve exploration. In Ta-Feng, the full model provides a stronger overall repeat-explore balance.

Table 7. Ablation study.

Dataset	Variant	Ablated/Changed Component	Overall Recall@20	Repeat Recall@20	Explore Recall@20
Dunnhumby	Full TL-PPR	Selected full model: raw frequency repeat head + time-decayed weighted LightGCN exploration head	0.154	0.279	0.021
	Raw frequency edge	Replace time-decayed graph edges with raw frequency edges	0.152	0.279	0.017
	Binary edge	Replace time-decayed graph edges with binary edges	0.151	0.279	0.016
	Repeat only	Remove the LightGCN exploration head	0.189	0.377	0.000
	Explore only	Remove the personal-popularity repeat head	0.018	0.000	0.029
	Fixed global quota	Replace household-specific quota with one global average quota	0.152	0.276	0.021
	No history filter	Allow the exploration head to recommend previously purchased items	0.136	0.252	0.013
	Global-popularity explore head	Replace the LightGCN exploration head with globally popular unseen items	0.151	0.279	0.018
Ta-Feng	Full TL-PPR	Selected full model: raw frequency repeat head + time-decayed weighted LightGCN exploration head	0.195	0.668	0.074
	Raw frequency edge	Replace time-decayed graph edges with raw frequency edges	0.191	0.668	0.069
	Binary edge	Replace time-decayed graph edges with binary edges	0.191	0.668	0.069
	Repeat only	Remove the LightGCN exploration head	0.156	0.807	0.000
	Explore only	Remove the personal-popularity repeat head	0.076	0.000	0.091
	Fixed global quota	Replace household-specific quota with one global average quota	0.195	0.668	0.074
	No history filter	Allow the exploration head to recommend previously purchased items	0.176	0.564	0.073
	Global-popularity explore head	Replace the LightGCN exploration head with globally popular unseen items	0.188	0.668	0.066

7. Significance Test

Statistical inference is an important component of recommender-system evaluation because average performance metrics alone do not show whether observed differences between models reflect reliable improvements or random variation across users. Ihemelandu and Ekstrand [54,55] describe statistical inference as a missing but necessary part of recommender-system evaluation, and Ihemelandu [56] further identifies the statistical significance of observed improvements as one of the key components of Top-N recommendation evaluation. This is particularly relevant because Top-N recommendation experiments often involve sparse relevance information, popularity bias, and large user samples, which can affect both metric values and their interpretation. Similarly, Ekstrand, Carterette, and Diaz [57] argue that recommender-system evaluation should move beyond point estimates by considering uncertainty and distributional variation in performance.

Several statistical tests have been used to compare ranking and recommendation models, including the paired t-test, randomization test, bootstrap test, Wilcoxon signed-rank test [58], and sign test. In this study, a paired bootstrap significance test is selected because all models are evaluated on the same final test households, creating a naturally paired comparison at the household level. For each household, the difference in Recall@K between two models is computed, and bootstrap resampling is used to estimate whether the mean difference is statistically reliable. This design controls for household-level variation in recommendation difficulty and avoids strong distributional assumptions about the distribution of Recall differences. The bootstrap procedure follows the general framework of Efron and Tibshirani [59]. Its use is also supported by prior ranking-system evaluation studies: Smucker, Allan, and Carterette [60] compare multiple significance tests in information retrieval and show that bootstrap, randomization, and paired t-tests often lead to similar conclusions, while Ihemelandu and Ekstrand [55] find that these tests behave more consistently than sign and Wilcoxon tests in large-scale search and top-N recommendation experiments. Therefore, the paired bootstrap test provides an appropriate method for assessing whether the observed differences between the proposed models and benchmark models are statistically reliable.

In this study, Recall@K is first computed separately for each model and each eligible household. The test is conducted for $K = 10, 20$, and 30 , and for the Overall, Repeat, and Explore evaluation scopes (Table 8). For each pairwise model comparison, household-level differences are computed as the Recall@K of model A minus the Recall@K of model B. These paired household-level differences are then resampled with replacement 10,000 times to approximate the sampling distribution of the mean difference. The 95% confidence interval is obtained from the bootstrap distribution, and a two-sided p -value is calculated to evaluate whether the observed mean difference is significantly different from zero. The final significance tables report the mean Recall@K of each model, the mean difference, the 95% confidence interval, the two-sided p -value, the significance level, and the direction of the difference. Because large recommendation experiments can make even small differences statistically significant, the interpretation considers both the p -value and the size and direction of the mean difference.

Table 8. Paired bootstrap significance test results for TL-PPR at Recall@20 on the final test households.

Dataset	Comparison	Overall Δ Recall@20	Repeat Δ Recall@20	Explore Δ Recall@20
Dunnhumby	TL-PPR – Global Popularity	+0.0998 [0.0874, 0.1124] ***	+0.1785 [0.1585, 0.1990] ***	+0.0031 [−0.0011, 0.0078] ns
	TL-PPR – Personal Popularity	−0.0418 [−0.0499, −0.0338] ***	−0.1109 [−0.1239, −0.0984] ***	+0.0164 [0.0117, 0.0215] ***
	TL-PPR – TL-HFE	+0.0603 [0.0498, 0.0711] ***	+0.0955 [0.0772, 0.1141] ***	0.0000 [0.0000, 0.0000] ns
Ta-Feng	TL-PPR – Global Popularity	+0.0786 [0.0740, 0.0835] ***	+0.5115 [0.4966, 0.5260] ***	−0.0137 [−0.0155, −0.0120] ***

TL-PPR – Personal Popularity	+0.0058 [0.0023, 0.0094] ***	−0.1344 [−0.1436, −0.1251] ***	+0.0352 [0.0321, 0.0386] ***
TL-PPR – TL-HFE	+0.0310 [0.0273, 0.0349] ***	+0.1987 [0.1851, 0.2122] ***	−0.0016 [−0.0043, 0.0010] ns

Note: *** ($p < 0.001$); ns = not significant.

Values report the mean paired difference in Recall@20 between TL-PPR and the comparison model, with 95% bootstrap confidence intervals in brackets. Positive values indicate that TL-PPR performs better. For the Dunnhumby dataset, Overall Recall is computed over 1243 test households, Repeat Recall over 1047 households, and Explore Recall over 1073 households. For the Ta-Feng dataset, Overall Recall is computed over 7006 test households, Repeat Recall over 3566 households, and Explore Recall over 6643 households. Significance levels: *** ($p < 0.001$), ** ($p < 0.01$), * ($p < 0.05$), ns = not significant.

For Dunnhumby, TL-PPR significantly outperforms Global Popularity in Overall and Repeat Recall@20, while the Explore Recall@20 difference is not statistically significant. Compared with Personal Popularity, TL-PPR performs significantly worse in Overall and Repeat Recall@20, confirming that the pure personal-popularity baseline remains stronger for repeat-oriented recommendation on this dataset. However, TL-PPR significantly improves Explore Recall@20 over Personal Popularity, showing that the LightGCN exploration head contributes to recommending previously unseen products. Compared with TL-HFE, TL-PPR significantly improves Overall and Repeat Recall@20, while Explore Recall@20 is statistically unchanged. Therefore, on Dunnhumby, the main benefit of TL-PPR is not superior repeat accuracy over Personal Popularity, but rather a more balanced repeat-explore trade-off than a purely repeat-based model.

For Ta-Feng, TL-PPR significantly improves Overall Recall@20 compared with all three benchmark models. The gain over Personal Popularity is statistically significant but small in magnitude, indicating that TL-PPR provides only a modest overall improvement over a strong repeat-oriented baseline. The Repeat Recall@20 results show that TL-PPR strongly outperforms Global Popularity and TL-HFE, but remains significantly below Personal Popularity, which is expected because Personal Popularity directly ranks each household's historically purchased products. For Explore Recall@20, TL-PPR significantly outperforms Personal Popularity, confirming that the exploration head improves the recommendation of unseen products compared with a purely repeat-based method. However, TL-PPR performs significantly worse than Global Popularity in Explore Recall@20, and its difference from TL-HFE is not statistically significant at $K = 20$. Overall, the Ta-Feng results show that TL-PPR improves overall performance mainly by combining strong repeat recommendation with some exploration ability, rather than by achieving the best standalone exploration performance.

Taken together, the paired bootstrap results provide a more nuanced interpretation of TL-PPR. Across both datasets, TL-PPR consistently improves Explore Recall@20 over Personal Popularity, demonstrating the value of adding a graph-based exploration head to a repeat-oriented recommendation structure. However, Personal Popularity remains stronger for Repeat Recall, especially in households with stable purchase histories. Therefore, TL-PPR should be interpreted as a hybrid model that improves the repeat-explore balance and overall recommendation utility, rather than as a model that uniformly dominates every baseline in every evaluation scope.

8. Discussion and Conclusions

This study examined grocery next-basket recommendation as a repeat-explore trade-off rather than only an aggregate accuracy problem. Using the Dunnhumby and Ta-Feng datasets, the results show that repeat-oriented methods remain very strong in grocery recommendation because many future purchases are habitual repurchases [7,8]. In particu-

lar, Personal Top-Frequency remains a difficult baseline to outperform for repeat-item prediction. This confirms that household-level purchase frequency is a strong predictor in grocery baskets and that more complex graph-based models do not automatically dominate simple personalized baselines.

The results also highlight an important practical trade-off. Increasing exploratory recommendations can reduce repeat and overall accuracy because positions allocated to new-to-household products are no longer available for high-probability repeat purchases. Therefore, the proposed hybrid models should not be interpreted as universal replacements for repeat-oriented baselines. If the objective is purely short-term accuracy, automatic replenishment, or routine basket completion, Personal Top-Frequency or other repeat-focused methods may remain preferable.

At the same time, relying only on repeat-oriented recommendations provides limited support for exploratory recommendations. TL-HFE and TL-PPR address this limitation by separating repeat-oriented and exploration-oriented ranking and combining them through household-specific quota interleaving. Among the proposed models, TL-PPR provides the most useful overall balance because it combines a strong personal-popularity repeat head with a LightGCN-based exploration head. The paired bootstrap tests show that TL-PPR consistently improves Explore Recall@20 over Personal Top-Frequency, although Personal Top-Frequency remains stronger for Repeat Recall. Consistent with prior work, a recommendation method can be considered successful when it improves beyond-accuracy performance while maintaining a comparable level of accuracy [5,61]. Based on this criterion, TL-PPR can be considered the best-balanced model in this study because it substantially improves exploratory recommendation compared with the strongest accuracy-oriented baseline, while preserving competitive overall accuracy and strong repeat performance.

The findings also clarify the role of time-decayed graph construction. Encoding frequency and recency in household-item edge weights helps the graph-based component better represent purchase strength than binary interaction edges alone. However, the benefit of time-decayed edges depends on the dataset and model configuration. This suggests that temporal weighting is useful, but it should be selected and validated carefully rather than treated as universally optimal. The sensitivity and ablation analyses support this interpretation by showing that graph edge type, recent-history window, quota clipping, and history filtering all affect the final repeat-explore balance.

From a managerial perspective, the choice of model should depend on the retailer's objective. For routine basket completion, replenishment, or short-term accuracy, a repeat-oriented baseline such as Personal Top-Frequency may be sufficient and even preferable. However, many retail applications require more than routine repurchase prediction. Retailers may also want to expose households to relevant new products, support cross-selling, promote seasonal or strategic items, reduce over-concentration on habitual products, or increase discovery within large assortments. In these settings, TL-HFE and TL-PPR are more appropriate because they explicitly reserve controlled space for exploratory recommendations. In these settings, TL-HFE and TL-PPR are more appropriate because both models explicitly separate repeat-oriented and exploration-oriented rankings and reserve controlled space for exploratory recommendations through household-specific quota interleaving. The main difference between the two models lies in the repeat-oriented component. TL-HFE represents a fully graph-based hybrid framework because both its exploitation-oriented ranking and exploration-oriented ranking are generated from the shared LightGCN embedding space. In contrast, TL-PPR replaces the graph-based repeat-oriented head with a household-level personal-popularity repeat head while retaining the same LightGCN-based exploration head. Therefore, TL-PPR is more suitable when the application requires a safer compromise between repeat accuracy and exploration, be-

cause its personal-popularity repeat head preserves strong habitual repurchase signals while the LightGCN exploration head introduces a controlled number of new-to-household products.

This study has several limitations. First, the proposed models are evaluated offline, so the results do not capture customer reactions, click behavior, substitution effects, or business outcomes in a live retail environment. Second, the time decay function uses validation-selected half-life values, but grocery categories may have different replenishment cycles; future research could examine adaptive or category-specific decay rates. Third, the models rely only on transaction histories and do not include prices, promotions, seasonality, product attributes, or household demographics, all of which may improve exploratory recommendations. Fourth, future research could incorporate additional beyond-accuracy metrics, such as diversity, novelty, calibration, and fairness [7,12]. Finally, although time-decayed weighting can reduce the influence of outdated interactions, standard bipartite graph propagation may still transmit noisy signals through highly popular products. Recent studies show that the benefit of GNNs depends on the relationship between graph structure, neighborhood patterns, and node distinguishability, and that real-world graphs may combine heterogeneity and heterophily in ways that challenge standard message-passing models [62,63]. Future work could therefore compare the proposed framework with sequence-based, Transformer-based, and heterophily-aware or heterogeneous GNN models.

Overall, the study shows that grocery next-basket recommendation should be evaluated through overall, repeat-specific, and explore-specific performance. The main contribution is not a new graph propagation rule, but a repeat-explore-aware framework that combines time-aware graph construction, hybrid ranking, household-specific quota interleaving, and validation-based model selection. The results demonstrate that explicitly modeling the repeat-explore trade-off provides a more informative and practically useful approach than optimizing aggregate accuracy alone.

Author Contributions: Conceptualization, Z.M.; methodology, Z.M.; validation, Z.M., S.L., R.V.; formal analysis, Z.M.; investigation, Z.M.; data curation, Z.M.; writing—original draft preparation, Z.M.; writing—review and editing, Z.M., S.L., R.V.; visualization, Z.M.; supervision, S.L., R.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: This study uses the Dunnhumby Complete Journey retail dataset freely available at <https://www.dunnhumby.com/source-files/> (accessed on 30 January 2026), and the Ta-Feng dataset freely available at <https://www.kaggle.com/datasets/chiranjivdas09/ta-feng-grocery-dataset/data> (accessed on 1 June 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NBR	Next-Basket Recommendation
GNN	Graph Neural Network
GCN	Graph Convolutional Network
LightGCN	Light Graph Convolutional Network
GraphSAGE	Graph Sample and Aggregate
BPR	Bayesian Personalized Ranking
PHR	Personalized Hit Ratio
NDCG	Normalized Discounted Cumulative Gain
Top-K	Top-K recommendation list

G-TopFreq	Global Top-Frequency
P-TopFreq	Personal Top-Frequency
TL-HFE	Two-Head LightGCN with History-Filtered Explore Head
TL-PPR	Two-Head LightGCN with Personal-Popularity Repeat Head
TREx	Two-step Repetition-Exploration framework
SKU	Stock-Keeping Unit

References

1. Liu, X.; Chen, X. Who Decides: The Consumer or the Retailer? An LLM-Assisted Bayesian Framework for Modeling Purchase Decisions with Retailer Influence. *J. Retail. Consum. Serv.* **2026**, *89*, 104645. <https://doi.org/10.1016/j.jretconser.2025.104645>.
2. Roy, A.; Sana, S.S.; Chaudhuri, K. Optimal Pricing of Competing Retailers under Uncertain Demand—a Two Layer Supply Chain Model. *Ann. Oper. Res.* **2018**, *260*, 481–500. <https://doi.org/10.1007/s10479-015-1996-0>.
3. Ricci, F.; Rokach, L.; Shapira, B. (Eds.) *Recommender Systems Handbook*; Springer: New York, NY, USA, 2022.
4. Kim, J.K.; Moon, H.S.; An, B.J.; Choi, I.Y. A Grocery Recommendation for Off-Line Shoppers. *Online Inf. Rev.* **2018**, *42*, 468–481. <https://doi.org/10.1108/OIR-04-2016-0104>.
5. Li, M.; Liu, Y.; Jullien, S.; Ariannezhad, M.; Yates, A.; Aliannejadi, M.; De Rijke, M. Are We Really Achieving Better Beyond-Accuracy Performance in Next Basket Recommendation? In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington, DC, USA, 14–18 July 2024*; ACM: New York, NY, USA, 2024; pp. 924–934.
6. Wang, B.; Ye, F.; Xu, J. A Personalized Recommendation Algorithm Based on the User’s Implicit Feedback in E-Commerce. *Future Internet* **2018**, *10*, 117. <https://doi.org/10.3390/fi10120117>.
7. Li, M.; Jullien, S.; Ariannezhad, M.; De Rijke, M. A Next Basket Recommendation Reality Check. *ACM Trans. Inf. Syst.* **2023**, *41*, 1–29 <https://doi.org/10.1145/3587153>.
8. Shao, Z.; Wang, S.; Zhang, Q.; Lu, W.; Li, Z.; Peng, X. A Systematical Evaluation for Next-Basket Recommendation Algorithms. In *Proceedings of the 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), Shenzhen, China, 13–16 October 2022*; pp. 1–10.
9. Faggioli, G.; Polato, M.; Aiolfi, F. Recency Aware Collaborative Filtering for Next Basket Recommendation. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, Genoa, Italy, 14–17 July 2020*; ACM: New York, NY, USA, 2020; pp. 80–87.
10. Hu, H.; He, X.; Gao, J.; Zhang, Z.-L. 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, 25–30 July 2020*; ACM: New York, NY, USA, 2020; pp. 1071–1080.
11. Ren, P.; Chen, Z.; Li, J.; Ren, Z.; Ma, J.; Rijke, M. de RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-Based Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*; PKP Publishing Services Network: Burnaby, BC, Canada, 2019; Volume 33, pp. 4806–4813.
12. Liu, Y.; Li, M.; Aliannejadi, M.; Rijke, M. de Repeat-Bias-Aware Optimization of Beyond-Accuracy Metrics for Next Basket Recommendation. In *Proceedings of the European Conference on Information Retrieval*; Springer Nature: Cham, Switzerland, 2025; pp. 214–229.
13. Ariannezhad, M.; Jullien, S.; Li, M.; Fang, M.; Schelter, S.; De Rijke, M. ReCANet: A Repeat Consumption-Aware Neural Network for Next Basket Recommendation in Grocery Shopping. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022*; ACM: New York, NY, USA, 2022; pp. 1240–1250.
14. Chizari, N.; Tajfar, K.; Moreno-García, M. Measuring Inter-Bias Effects and Fairness-Accuracy Trade-Offs in GNN-Based Recommender Systems. *Future Internet* **2025**, *17*, 461. <https://doi.org/10.3390/fi17100461>.
15. Jannach, D.; De Souza, P.; Moreira, G.; Oldridge, E. Why Are Deep Learning Models Not Consistently Winning Recommender Systems Competitions Yet?: A Position Paper. In *Proceedings of the Recommender Systems Challenge 2020, Virtual Event, Brazil, 26 September 2020*; ACM: New York, NY, USA, 2020; pp. 44–49.
16. Zhao, M.; Wu, L.; Liang, Y.; Chen, L.; Zhang, J.; Deng, Q.; Wang, K.; Shen, X.; Lv, T.; Wu, R. Investigating Accuracy-Novelty Performance for Graph-Based Collaborative Filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*; ACM: New York, NY, USA, 2022; pp. 50–59.

17. Song, Y.; Lian, J.; Sun, S.; Huang, H.; Li, Y.; Jin, H.; Xie, X. Show Me the Whole World: Towards Entire Item Space Exploration for Interactive Personalized Recommendations. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*; ACM: New York, NY, USA, 2022; pp. 947–956.
18. Chen, Y.; Li, J.; Liu, C.; Li, C.; Anderle, M.; McAuley, J.; Xiong, C. Modeling Dynamic Attributes for Next Basket Recommendation. *arXiv* **2021**, arXiv:2109.11654.
19. Lashinin, O.; Krasilnikov, D.; Milogradskii, A.; Ananyeva, M. SAFERec: Self-Attention and Frequency Enriched Model for Next Basket Recommendation. In *Advances in Information Retrieval*; Springer: Cham, Switzerland, 2024.
20. Moreno-García, M.N.; Prieta, F.D.L. Deep Learning in Recommender Systems. *Future Internet* **2026**, *18*, 131. <https://doi.org/10.3390/fi18030131>.
21. Wu, S.; Sun, F.; Zhang, W.; Xie, X.; Cui, B. Graph Neural Networks in Recommender Systems: A Survey. *ACM Comput. Surv.* **2023**, *55*, 1–37. <https://doi.org/10.1145/3535101>.
22. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph Neural Networks: A Review of Methods and Applications. *AI Open* **2020**, *1*, 57–81. <https://doi.org/10.1016/j.aiopen.2021.01.001>.
23. Li, W.; Song, X.; Tu, Y. GraphDRL: GNN-Based Deep Reinforcement Learning for Interactive Recommendation with Sparse Data. *Expert Syst. Appl.* **2025**, *273*, 126832. <https://doi.org/10.1016/j.eswa.2025.126832>.
24. Kim, M.; Oh, J.; Do, J.; Lee, S. Debiasing Neighbor Aggregation for Graph Neural Network in Recommender Systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, Atlanta, GA, USA, 17–21 October 2022; pp. 4128–4132.
25. Liu, T.; Liu, B. Next Basket Recommendation Based on Graph Attention Network and Transformer. *J. Phys. Conf. Ser.* **2022**, *2303*, 012023. <https://doi.org/10.1088/1742-6596/2303/1/012023>.
26. He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; Wang, M. 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*, Xi'an, China, 25–30 July 2020.
27. Wang, X.; Ming, W.; Fu, Z.; Ma, X. Enhancing Recommendation Diversity and Accuracy with Product Paths and Time Decay Mechanisms. *PLoS ONE* **2026**, *21*, e0343638. <https://doi.org/10.1371/journal.pone.0343638>.
28. Behera, G.; Nain, N. Collaborative Filtering with Temporal Features for Movie Recommendation System. *Procedia Comput. Sci.* **2023**, *218*, 1366–1373. <https://doi.org/10.1016/j.procs.2023.01.115>.
29. Ding, Y.; Li, X. Time Weight Collaborative Filtering. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, Bremen, Germany, 31 October–5 November 2005; ACM: New York, NY, USA, 2005; pp. 485–492.
30. Wu, P.; Yeung, C.H.; Liu, W.; Jin, C.; Zhang, Y.-C. Time-Aware Collaborative Filtering with the Piecewise Decay Function. *arXiv* **2010**, arXiv:1010.3988.
31. He, X.; Wei, T.; He, J. Robust Basket Recommendation via Noise-Tolerated Graph Contrastive Learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, Birmingham, UK, 21–25 October 2023; ACM: New York, NY, USA, 2023; pp. 709–719.
32. Zhang, M.; Wu, S.; Yu, X.; Liu, Q.; Wang, L. Dynamic Graph Neural Networks for Sequential Recommendation. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 4741–4753.
33. Tang, H.; Wu, S.; Xu, G.; Li, Q. Dynamic Graph Evolution Learning for Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Taipei, Taiwan, 23–27 July 2023; ACM: New York, NY, USA, 2023; pp. 1589–1598.
34. Liu, X.; Wang, X. Cohesive Subgraph Identification in Weighted Bipartite Graphs. *Appl. Sci.* **2021**, *11*, 9051. <https://doi.org/10.3390/app11199051>.
35. Pande, A.; Ni, K.; Kini, V. SWAG: Item Recommendations Using Convolutions on Weighted Graphs. In *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 9–12 December 2019; pp. 2903–2912.
36. Wu, J.; Wu, Z.; Li, X.; Li, R.-H.; Wang, G. TFPS: A Temporal Filtration-Enhanced Positive Sample Set Construction Method for Implicit Collaborative Filtering. *arXiv* **2026**, arXiv:2602.22521.
37. Chanpuriya, S.; Rossi, R.A.; Kim, S.; Yu, T.; Hoffswell, J.; Lipka, N.; Guo, S.; Musco, C. Direct Embedding of Temporal Network Edges via Time-Decayed Line Graphs. *arXiv* **2022**, arXiv:2210.00032.
38. Zhang, Z.; Chen, Y.; Zhang, J. Stable Diffusion and LightGCN Based AIGC Modeling for Generating and Optimizing Communication Effectiveness of Guangdong Hong Kong Macao City Brand Visual Symbols. In *Proceedings of the 2025 International Conference on Artificial Intelligence, Virtual Reality and Interaction Design*, Dongguan, China, 17–19 October 2025; ACM: New York, NY, USA, 2025; pp. 680–686.

39. Huang, Z.; Tang, D.; Zhao, R.; Rao, W. A Scientific Paper Recommendation Method Using the Time Decay Heterogeneous Graph. *Scientometrics* **2024**, *129*, 1589–1613. <https://doi.org/10.1007/s11192-024-04933-4>.
40. Chen, Z.; Huang, J.; Silvestri, F.; Zhang, Y.; Ahn, H.; Tolomei, G. Joint Factual and Counterfactual Explanations for Top-k GNN-Based Recommendations. *ACM Trans. Recomm. Syst.* **2026**, *4*, 1–26. <https://doi.org/10.1145/3731683>.
41. Yavru, I.B.; Yalcin, E.; Bilge, A.; Gunduz, S.Y. SUPER: Smart User-Centric Popularity Exposure Reduction for Fair and Diverse Recommendations. *IEEE Access* **2026**, *14*, 37582–37605. <https://doi.org/10.1109/ACCESS.2026.3671645>.
42. Kolesnikov, S.; Lashinin, O.; Pechatov, M.; Kosov, A. Next Period Recommendation Reality Check. *arXiv* **2021**, arXiv:2110.05589.
43. Hu, Y.; Koren, Y.; Volinsky, C. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008*; IEEE: New York, NY, USA, 2009; pp. 263–272.
44. Wang, X.; He, X.; Wang, M.; Feng, F.; Chua, T.-S. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019*; pp. 165–174.
45. Chen, B.; Guo, W.; Tang, R.; Xin, X.; Ding, Y.; He, X.; Wang, D. TGCN: Tag Graph Convolutional Network for Tag-Aware Recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, Ireland, 19–23 October 2020*; ACM: New York, NY, USA, 2020; pp. 155–164.
46. Priyam, U.; Shah, H.; Botta, E. EDGE-Rec: Efficient and Data-Guided Edge Diffusion For Recommender Systems Graphs. *arXiv* **2024**, arXiv:2409.14689.
47. Sánchez-Moreno, D.; Zheng, Y.; Moreno-García, M.N. Time-Aware Music Recommender Systems: Modeling the Evolution of Implicit User Preferences and User Listening Habits in A Collaborative Filtering Approach. *Appl. Sci.* **2020**, *10*, 5324. <https://doi.org/10.3390/app10155324>.
48. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; pp. 1025–1035.
49. El Alaoui, D.; Riffi, J.; Sabri, A.; Aghoutane, B.; Yahyaouy, A.; Tairi, H. Deep GraphSAGE-Based Recommendation System: Jumping Knowledge Connections with Ordinal Aggregation Network. *Neural Comput. Appl.* **2022**, *34*, 11679–11690. <https://doi.org/10.1007/s00521-022-07059-x>.
50. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907.
51. Wang, R.; Lou, J.; Jiang, Y. LightGCAN: A Lightweight Graph Convolutional Attention Network for User Preference Modeling and Personalized Recommendation. *Expert Syst. Appl.* **2023**, *232*, 120741. <https://doi.org/10.1016/j.eswa.2023.120741>.
52. Triantaphyllou, E. *Multi-Criteria Decision Making Methods: A Comparative Study*; Applied Optimization; Springer: Boston, MA, USA, 2000; Volume 44.
53. Hafez, M.M.; Redondo, R.P.D.; Vilas, A.F.; Pazó, H.O. Multi-Criteria Recommendation Systems to Foster Online Grocery. *Sensors* **2021**, *21*, 3747. <https://doi.org/10.3390/s21113747>.
54. Ihemelandu, N.; Ekstrand, M.D. Statistical Inference: The Missing Piece of RecSys Experiment Reliability Discourse. *arXiv* **2021**, arXiv:2109.06424.
55. Ihemelandu, N.; Ekstrand, M.D. Inference at Scale Significance Testing for Large Search and Recommendation Experiments. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, Taipei, Taiwan, 23–27 July 2023*; pp. 2087–2091.
56. Ihemelandu, N. Best Practices for Top-N Recommendation Evaluation: Candidate Set Sampling and Statistical Inference Techniques. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17–21 October 2022*; ACM: New York, NY, USA, 2022; pp. 5120–5123.
57. Ekstrand, M.D.; Carterette, B.; Diaz, F. Distributionally-Informed Recommender System Evaluation. *ACM Trans. Recomm. Syst.* **2024**, *2*, 1–27. <https://doi.org/10.1145/3613455>.
58. Anelli, V.W.; Bellogin, A.; Ferrara, A.; Malitesta, D.; Merra, F.A.; Pomo, C.; Donini, F.M.; Di Noia, T. Elliot: A Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, 11–15 July 2021*; ACM: New York, NY, USA, 2021; pp. 2405–2414.
59. Efron, B.; Tibshirani, R. *An Introduction to the Bootstrap*; Monographs on Statistics and Applied Probability; Chapman & Hall: Boca Raton, FL, USA, 1998.
60. Smucker, M.D.; Allan, J.; Carterette, B. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, Lisbon, Portugal, 6–10 November 2007*; ACM: New York, NY, USA, 2007; pp. 623–632.

61. Zhao, Y.; Wang, Y.; Liu, Y.; Cheng, X.; Aggarwal, C.; Derr, T. Fairness and Diversity in Recommender Systems: A Survey. *ACM Trans. Intell. Syst. Technol.* **2025**, *16*, 1–28. <https://doi.org/10.1145/3664928>.
62. Luan, S.; Hua, C.; Xu, M.; Lu, Q.; Zhu, J.; Chang, X.-W.; Fu, J.; Leskovec, J.; Precup, D. When Do Graph Neural Networks Help with Node Classification? Investigating the Impact of Homophily Principle on Node Distinguishability. In Proceedings of the Advances in Neural Information Processing Systems 36, New Orleans, LA, USA, 10–16 December 2023.
63. Lin, J.; Guo, X.; Zhang, S.; Zhu, Y.; Shun, J. When Heterophily Meets Heterogeneity: Challenges and a New Large-Scale Graph Benchmark. In Proceedings of the KDD'25: The 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Toronto, ON, Canada, 3–7 August 2025.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.