SynGraph: A Dynamic Graph-LLM Synthesis Framework for Sparse Streaming User Sentiment Modeling

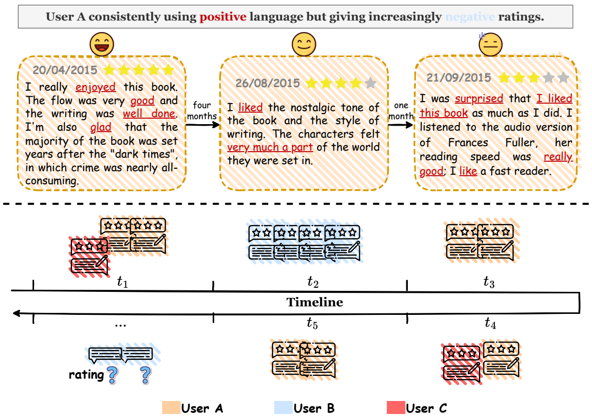
Xin Zhang[1], **Qiyu Wei**[1], **Yingjie Zhu**[2], **Linhai Zhang**[3], **Deyu Zhou**[4], **Sophia Ananiadou**[1]
[1]The University of Manchester [2]Harbin Institute of Technology
[3]King's College London [4]Southeast University
{xin.zhang-41, qiyu.wei}@postgrad.manchester.ac.uk
sophia.ananiadou@manchester.ac.uk, zhuyj@stu.hit.edu.cn
linhai.zhang@kcl.ac.uk, d.zhou@seu.edu.cn

Abstract

User reviews on e-commerce platforms exhibit dynamic sentiment patterns driven by temporal and contextual factors. Traditional sentiment analysis methods focus on static reviews, failing to capture the evolving temporal relationship between user sentiment rating and textual content. Sentiment analysis on streaming reviews addresses this limitation by modeling and predicting the temporal evolution of user sentiments. However, it suffers from data sparsity, manifesting in temporal, spatial, and combined forms. In this paper, we introduce SYNGRAPH, a novel framework designed to address data sparsity in sentiment analysis on streaming reviews. SYNGRAPH alleviates data sparsity by categorizing users into mid-tail, long-tail, and extreme scenarios and incorporating LLM-augmented enhancements within a dynamic graph-based structure. Experiments on real-world datasets demonstrate its effectiveness in addressing sparsity and improving sentiment modeling in streaming reviews.

1 Introduction

User reviews on e-commerce platforms provide valuable insights into customer preferences, product performance, and user satisfaction. These reviews reflect users' momentary sentiments and exhibit temporal dynamics as users interact with products over time. Traditional sentiment analysis methods, however, predominantly operate in static contexts, where individual reviews are treated as independent data points (Tang et al., 2015; Chen et al., 2016; Kim and Srivastava, 2007; Mittal et al., 2022; Zhang et al., 2025). This paradigm overlooks the inherent temporal dependencies and evolving nature of user sentiments, which are crucial for capturing the full spectrum of user behavior in real-world scenarios. For instance, as shown in Figure 1, a user's sentiment rating may initially be highly positive (e.g., "I really enjoyed this book") but gradually shift to neutral (e.g., "I was surprised that I liked this book as much as I did") in subsequent interactions. These transitions highlight the dynamic relationship between sentiment changes and textual cues, require models to learn sentiment change patterns from historical reviews, and leverage this understanding to predict future sentiment ratings.



I really enjoyed this book. The flow was very good and the writing was well done. I'm also glad that the majority of the book was set years after the "dark times", in which crime was nearly all-consuming.

I liked the nostalgic tone of the book and the style of writing. The characters felt very much a part of the world they were set in.

I was surprised that I liked this book as much as I did. I listened to the audio version of Frances Fuller, her reading speed was really good; I like a fast reader.

Figure 1: Sentiment Evolution in Streaming Reviews.

As a solution to these limitations, the task of "Sentiment Analysis on Streaming Reviews" has been proposed, aiming to model the temporal evolution of user sentiments across sequential time windows and predict user future sentiments (Zhang et al., 2023; Wu et al., 2023). This task emphasizes the interplay between numerical sentiment scores and their associated textual content over time. However, this task also introduces significant challenges, primarily due to the problem of data sparsity (Guo, 2013; Du et al., 2022; Zhang et al., 2024). Previous studies have commonly introduced graph structure information to supplement sparse user data in static scenarios (Zhou et al., 2023; Wang et al., 2023; Chen et al., 2022). Despite this, in streaming scenarios, the inclusion of graph structure information further complicates the classification and discussion of sparse users (Xu et al., 2024). As shown in Figure 1, some

Findings of the Association for Computational Linguistics: ACL 2025, pages 16338–16356
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

users leave reviews at irregular and widely spaced intervals, making it challenging for temporal models to capture sequential dependencies (User A: temporal sparsity). Others have limited social connections, resulting in a lack of neighbor information to support sentiment analysis (User B: spatial sparsity). Additionally, some users exhibit both temporal and spatial sparsity, further complicating predictive modeling (User C: combined sparsity). Existing methods fail to effectively address these sparsity issues, struggling to model temporal dependencies and spatial interactions.

To tackle these challenges, we propose SYNGRAPH, a novel framework tailored for Sentiment Analysis on Streaming Reviews. SYNGRAPH operates within a dynamic graph structure and leverages LLM-augmented data synthesis to enhance sparse user representations. Specifically, SYNGRAPH employs a decompose-and-recompose strategy to categorize users into three scenarios—mid-tail, long-tail, and extreme scenarios—based on their temporal and spatial characteristics. For each scenario, SYNGRAPH integrates three core components: (a) *Local and global graph understanding*: Captures both micro-level (local) and macro-level (global) patterns of user-product interactions. (b) *High-order relation understanding*: Explores second-order and higher-order dependencies to enrich graph representations. (c) *LLM-augmented data synthesis*: Utilizes large language models to induce supplementary data for sparse users, enhancing representation robustness. Combining these components, SYNGRAPH effectively addresses the sparsity challenges in streaming reviews and provides a robust solution for learning dynamic sentiment patterns. Our main contributions can be summarized as follows:

- We introduce SYNGRAPH, a novel framework that integrates dynamic graph modeling with LLM-augmented data synthesis to address the problem of data sparsity in sentiment analysis on streaming reviews.

- We propose a decompose-and-recompose strategy to categorize users into different sparsity levels (mid-tail, long-tail, and extreme) and apply tailored profile-enhancing techniques to synthesize data for each category.

- We conduct extensive experiments on real-world datasets, demonstrating that SYNGRAPH significantly improves sentiment prediction accuracy while effectively mitigating the impact of temporal and spatial sparsity.

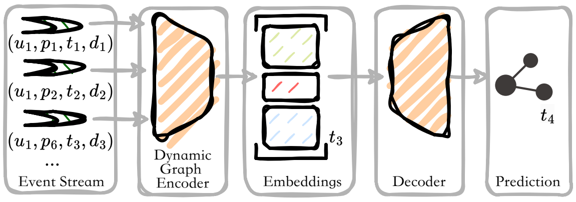


Figure 2: Streaming User Review Graph.

2 Problem Formulation

2.1 Continuous-Time Dynamic Graph

To model evolving user-product interactions, we represent the system as a Continuous-Time Dynamic Graph (CTDG; Zhu et al. (2022)). Formally, for a time span $T_{\text{span}} = [t_0, t_n]$, the dynamic graph is defined as $G_{T_{\text{span}}} = \left(G_{t_0}, O_{[t_1,t_n]}\right)$, where G_{t_0} is the initial graph at t_0 and $O_{[t_1,t_n]}$ is a sequence of timestamped events. Each event $o_i \in O_{[t_1,t_n]}$ represents either a node update (v_i, t_i) (adding a new user or product) or an edge update (u_i, p_i, t_i) (a user–product interaction).

2.2 Streaming User Review Graph

In the streaming review scenario, we observe a temporal sequence of reviews $E = \{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_T\}$, with each review $\mathcal{E}_i = (u_i, p_i, t_i, d_i, y_i)$ capturing the user u_i, product p_i, timestamp t_i, review content d_i, and sentiment y_i. We instantiate the CTDG framework to form our Streaming User Review Graph G_T, where users and products are nodes and each review creates a directed edge (u_i, p_i, t_i) annotated with $\{d_i, y_i\}$. In this graph, new reviews trigger node and edge updates, seamlessly mirroring the CTDG structure. As illustrated in Figure 2, interactions arrive as a continuous stream and are processed by a dynamic graph encoder that generates temporal embeddings reflecting evolving user preferences. These embeddings feed into a decoder that predicts future user sentiment at time t_4: $y = f\left(\mathcal{E}_t \,\middle|\, \{\mathcal{E}_1, \ldots, \mathcal{E}_{t-1}\}\right)$. By leveraging the CTDG structure, our approach naturally captures both temporal dependencies and structural evolution in user-product interactions, thereby improving sentiment prediction.

3 SYNGRAPH Framework

We propose SYNGRAPH, a three-stage framework that addresses data sparsity in Sentiment

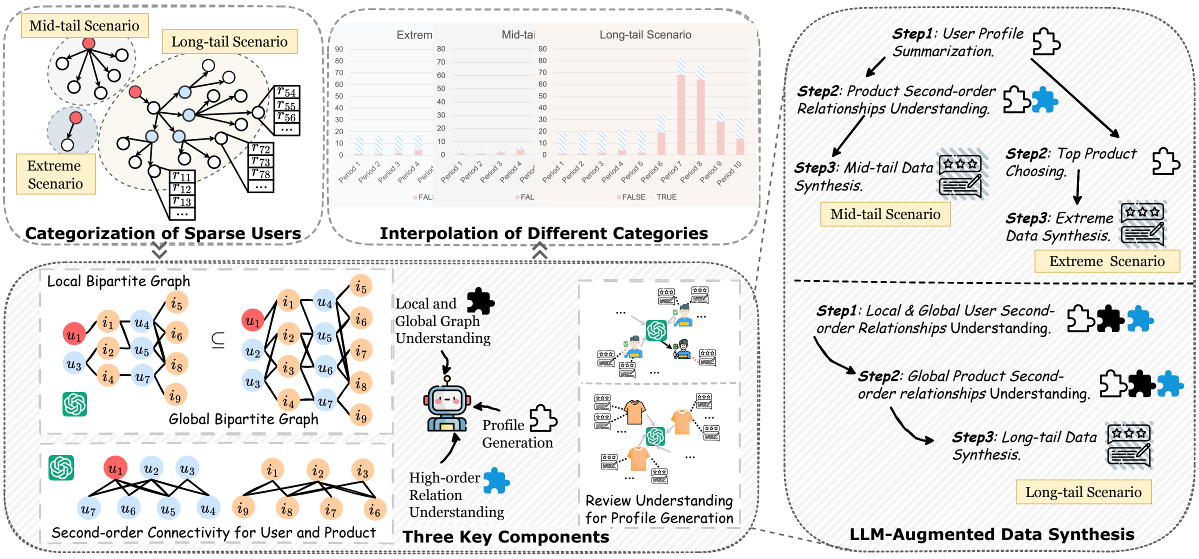

Mid-tail Scenario
Long-tail Scenario
Extreme Scenario
Categorization of Sparse Users
Extrem
Mid-tai
Long-tail Scenario
FALSE
TRUE
Interpolation of Different Categories
Step1: User Profile Summarization.
Step2: Product Second-order Relationships Understanding.
Step3: Mid-tail Data Synthesis.
Mid-tail Scenario
Step2: Top Product Choosing.
Step3: Extreme Data Synthesis.
Extreme Scenario
Step1: Local & Global User Second-order Relationships Understanding.
Step2: Global Product Second-order relationships Understanding.
Step3: Long-tail Data Synthesis.
Long-tail Scenario
LLM-Augmented Data Synthesis
Local Bipartite Graph
Global Bipartite Graph
Second-order Connectivity for User and Product
Local and Global Graph Understanding
Profile Generation
High-order Relation Understanding
Three Key Components
Review Understanding for Profile Generation

Figure 3: Overview of SynGraph, a three-stage framework for addressing data sparsity in sentiment analysis on streaming reviews. The three stages include: (1) Categorization of sparse users—identifying user types based on temporal and structural patterns; (2) LLM-augmented data Synthesis—enhancing user and product profiles using first- and second-order relationships; (3) Interpolation of different categories—integrating synthesized data to improve representation learning and sentiment prediction.

Analysis on Streaming Reviews by leveraging LLM-augmented dynamic graphs. The SynGraph framework consists of three key stages: (1) Categorization of sparse users—identifying user types based on their temporal and structural patterns; (2) LLM-augmented representation learning—enhancing user and product profiles using first- and second-order relationships; (3) Interpolation of different categories—integrating synthesized data to improve representation learning and sentiment prediction.

3.1 Categorization of Sparse Users

To better model user behaviors in streaming sentiment analysis, we classify users into three sparse categories:

(i) Mid-tail Users: These users actively contribute reviews within specific time frames but exhibit fluctuations in engagement across different intervals. To quantify this variability, we measure the variance in user interactions over a set of time intervals T: $\sigma^2_{I(u_m)} = \frac{1}{T}\sum_{t=1}^{T}\left(I_t(u_m) - \bar{I}(u_m)\right)^2$, where $I_t(u_m)$ represents the number of interactions by user u_m in interval t, and $\bar{I}(u_m)$ is the mean interaction count across all intervals. A higher variance indicates greater fluctuations, suggesting that the user's engagement pattern is inconsistent over time. *(ii) Long-tail Users:* Long-tail users contribute a limited number of reviews and have sparse first-order interactions, meaning they have few direct neighbors in the interaction graph. This limits the effectiveness of traditional neighbor-based analysis. However, these users often maintain a substantial number of second-order connections, allowing for indirect information propagation. We define first-order connectivity as: $C_1(u_l) = \sum_{v\in\mathcal{N}_1(u_l)} w(u_l, v)$, where $\mathcal{N}_1(u_l)$ is the set of direct neighbors of user u_l, and $w(u_l, v)$ denotes the interaction weight between u_l and neighbor v. A small $C_1(u_l)$ value indicates that user u_l has sparse immediate connections, whereas a high $C_2(u_l)$ value suggests that second-order neighbors provide rich contextual signals. *(iii) Extreme Users:* These users are highly sparse both in terms of review activity and connectivity within the interaction graph, making them challenging to model with traditional approaches. Their lack of both first- and second-order connections results in minimal information propagation. We define the overall connectivity score as: $C(u_e) = \sum_{v\in\mathcal{N}(u_e)} w(u_e, v)$, where $\mathcal{N}(u_e)$ denotes the set of all neighbors connected to u_e. A near-zero $C(u_e)$ indicates that the user is almost completely isolated in the network.

As illustrated in the top-left part of Figure 3 ("Categorization of Sparse Users"), these three categories of sparse users exhibit distinct distributions. In the figure, the three types of sparse users are highlighted in red, the first-order neighbors (products) in white, and the second-order homogeneous neighbors (users) in blue, with each node containing a subset of relevant reviews.

3.2 LLM-Augmented Data Synthesis

In this section, we introduce an LLM-augmented Data Synthesis strategy, using LLM to synthesize user and product representations while incorporating structural neighbor information from dynamic graphs. This stage consists of two main parts: Key Components for LLM-augmented Data Synthesis and LLM-augmented Data Synthesis.

Key Components for LLM-Augmented Data Synthesis: To effectively handle data sparsity in streaming sentiment analysis, we introduce three essential components that enable LLMs to synthesize high-quality augmented data.

(1) Local-Global Graph Understanding: In our framework, the user profile is enhanced by supplementing it with contextual information derived from both local and global neighbor data. Specifically, the local information is obtained from the immediate neighbors that reflect recent and direct interactions, while the global information is collected from a broader set of neighbors, capturing long-term trends and overall network context. Our approach employs LLM to interpret and enrich the user's profile based on these local and global cues.

(2) High-order Relation Understanding: Instead of relying solely on direct connections, we extract second-order relationships to improve user-product profiles. Given a user u with neighbors $\mathcal{N}(u)$, we define their second-order neighborhood as: $\mathcal{N}^2(u) = \Big(\bigcup_{v \in \mathcal{N}(u)} \mathcal{N}(v)\Big) \setminus \{u\}$.

(3) Profile Generation: For sparse users and products lacking explicit profiles, we utilize LLMs to synthesize synthetic profile based on existing reviews. These profiles can be directly extracted from user-generated content or inferred through second-order neighborhood information within the interaction graph. Formally, we define the profile generation function as: $\mathcal{P}(x) = \text{LLM}\big(\mathcal{R}(x), P_x\big)$, where $\mathcal{R}(x)$ is the selected review set for entity x (which can be a user or a product), and P_x is the corresponding guiding prompt. In our implementation, we use:P_u for generating user profiles, P_{prod} for generating product profiles, and P_{data} for synthesizing the final review data.

LLM-Augmented Data Synthesis: After obtaining the three key components, we combine these three methods and use them in three sparse categories.

(a) Mid-tail Data Synthesis: Mid-tail users have moderate interactions, allowing profile generation from historical reviews and second-order product relationships. Formally, the profile of a mid-tail user u_m is generated as: $\mathcal{P}(u_m) = \text{LLM}\big(\mathcal{R}(u_m), P_u\big)$. To incorporate second-order product relationships, we refine the product profile set as: $\mathcal{P}_{\text{set}}(u_m) = \text{LLM}\big(P_{\text{prod}}, \mathcal{N}^2(\mathcal{N}(u_m))\big)$, where P_{prod} is the prompt used for generating product profiles. The final synthesized data is then constructed by integrating both user and product profiles:

$$\mathcal{D}_M = \text{LLM}\big(P_{\text{data}}, \mathcal{P}(u_m), \mathcal{P}_{\text{set}}(u_m)\big). \quad (1)$$

(b) Long-tail Data Synthesis: Long-tail users have significantly fewer interactions, necessitating the use of neighbor-based augmentation. Their profiles are constructed using both local and global second-order relationships: $\mathcal{P}(u_l) = \text{LLM}\big(\mathcal{N}^2_{\text{local}}(u_l), \mathcal{N}^2_{\text{global}}(u_l), \mathcal{R}_{\text{chosen}}(u_l), P_u\big)$, where $\mathcal{N}^2_{\text{local}}(u_l)$ and $\mathcal{N}^2_{\text{global}}(u_l)$ denote the second-order neighbors from the local and global graphs, respectively. Similarly, product profiles for long-tail users are defined as: $\mathcal{P}_{\text{set}}(u_l) = \text{LLM}\big(P_{\text{prod}}, \mathcal{N}^2(\mathcal{N}(u_l))\big)$. The synthesized data for long-tail users is obtained by:

$$\mathcal{D}_L = \text{LLM}\big(P_{\text{data}}, \mathcal{P}(u_l), \mathcal{P}_{\text{set}}(u_l)\big). \quad (2)$$

(c) Extreme Data Synthesis: For extreme users with minimal or no interactions, we introduce pseudo-data synthesis based on high-rated products. The user profile is synthesized as: $\mathcal{P}(u_e) = \text{Profile}(u_e)$, where $\text{Profile}(u_e)$ denotes a predefined or externally obtained profile for the extreme user u_e. Popular products are incorporated as proxies: $\mathcal{P}_{\text{set}}(u_e) = \text{LLM}\Big(\mathcal{R}_{\text{chosen}}(P_{\text{high}}), P_{\text{prod}}\Big)$, where P_{high} denotes a selected set of high-rated products and $\mathcal{R}_{\text{chosen}}(P_{\text{high}})$ is the corresponding review set. The final synthesized data for extreme users is given by:

$$\mathcal{D}_E = \text{LLM}\big(P_{\text{data}}, \mathcal{P}(u_e), \mathcal{P}_{\text{set}}(u_e)\big). \quad (3)$$

By leveraging LLM-based profile generation, second-order relationship modeling, and local-global graph understanding, SYNGRAPH effectively synthesizes data for sparse users, enhancing model stability across varying sparsity levels.

Category	Normal		Mid-tail		Long-tail		Extreme		Interpolated Review Count		
	Count	Proportion	Count	Proportion	Count	Proportion	Count	Proportion	Mid-tail	Long-tail	Extreme
Magazine_Subscriptions	183	(52.59%)	8	(2.30%)	154	(44.25%)	3	(0.86%)	67	1287	26
Appliances	11	(23.40%)	2	(4.26%)	19	(40.36%)	15	(31.91%)	15	158	126
Gift_Cards	203	(44.42%)	45	(9.85%)	209	(45.73%)	0	(0.00%)	358	1753	0

Table 1: **Left:** User distribution across different sparsity levels. **Right:** Number of interpolated reviews.

3.3 Interpolation of Different Categories

After generating enriched representations for each user category, we apply data interpolation across different user groups to improve sentiment modeling and ensure data balance. To quantify the interpolation need for a given user u, we define the interpolation factor as: $I_f(u) = \frac{1}{T}\sum_{t=1}^{T}\mathbf{1}_{\{I_t(u)=0\}}$, where $\mathbf{1}_{\{I_t(u)=0\}}$ is an indicator function that returns 1 if the user u has no interactions at time step t, ensuring that interpolation is applied where necessary. Using our interpolation position search method, we identify the missing data points for each category and apply interpolation accordingly. The total number of interpolated interactions for a given category C is computed as:

$$I_{\text{total}}(C) = \sum_{u \in C}\sum_{t=1}^{T}\mathbf{1}_{\{I_t(u)=0\}}, \tag{4}$$

where C represents a user category (mid-tail, long-tail, or extreme). The interpolated data is then incorporated into the training pipeline, ensuring that each user maintains a minimum data availability threshold of 10 interactions.

4 Experiments

4.1 Experiments Setup

Dataset statistics. We utilize three datasets from the Amazon dataset collection (Ni et al., 2019), specifically Magazine_Subscriptions, Appliances, and Gift_Cards, chosen for their relatively smaller data sizes. To preserve the integrity of the original data distribution, we retain them in their raw form without additional preprocessing. Users in each dataset are categorized into mid-tail, long-tail, and extreme users based on the definitions established in this paper. (The detailed user categorization process is provided in the appendix.) The distribution of users across different sparsity levels is presented in Table 1.

Baselines. We compare our approach against two categories of baseline models: (1) Text-based models: BiLSTM+Att, BERT-Sequence (Devlin et al., 2019). (2) User and product-based models: CHIM (Amplayo, 2019), IUPC (Lyu et al., 2020), NGSAM (Zhou et al., 2021), and DC-DGNN (Zhang et al., 2023). Among these, DC-DGNN is specifically designed for continuous dynamic graph learning on streaming data. Models marked with * denote those trained on datasets augmented with the three types of synthesized data synthesized via the SynGraph framework.

Implementation details. For user and product embeddings, all models utilize a 128-dimensional representation. The batch size is set to 8, and the learning rate is $3e-5$, with training conducted for 2 epochs. Sentiment analysis is formulated as a classification problem, and model performance is evaluated using seven metrics: Accuracy, Precision, Recall, F1-score, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). The dataset is split into training and test sets at a $9:1$ ratio. Table 1 also presents the statistics of interpolated review counts across different categories.

4.2 Main Results

The performance on Magazine_Subscriptions, Appliances, and Gift_Cards datasets, illustrating the impact of incorporating synthesized data into sentiment analysis of streaming user reviews, are presented in Table 2. The key observations are as follows: (1) Across all three datasets, with a total of 18 experimental configurations spanning six different models, we observe that 15 out of 18 settings exhibit a significant improvement in performance when leveraging the SynGraph framework. This result underscores the effectiveness of SynGraph in enhancing sentiment analysis in streaming settings. (2) Among all evaluated models, DC-DGNN and its augmented variant DC-DGNN* consistently achieve the best performance, both before and after incorporating synthesized data. This finding highlights the critical role of dynamic user and product modeling in sentiment analysis for streaming reviews. (3) A comparative analysis of test set results before and after incorporating synthesized data

Method	**Evaluation Metrics**							*RMSE Reduction*
	Accuracy (↑)	Precision (↑)	Recall (↑)	F1 (↑)	MSE (↓)	RMSE (↓)	MAE (↓)	
Dataset: Magazine_Subscriptions								
BiLSTM+Att	0.6910	0.4040	0.4054	0.4019	1.5021	1.2256	0.5837	-
BiLSTM+Att*	0.6831	0.5966	0.3667	0.3976	1.3862	1.1774	0.5628	↓3.93%
Bert-Sequence	0.6953	0.2589	0.3049	0.2791	1.2918	1.1366	0.5451	-
Bert-Sequence*	0.7035	0.4479	0.5278	0.4792	0.5160	0.7183	0.3654	↓36.80%
NGSAM	-	-	-	-	1.2867	1.1343	0.7772	-
NGSAM*	-	-	-	-	0.9433	0.9712	0.6077	↓14.38%
CHIM	0.6084	0.3549	0.3002	0.3073	1.5392	1.2406	0.6831	-
CHIM*	0.6995	0.2453	0.2768	0.2553	1.8798	1.3711	0.6557	↑10.52%
IUPC	0.7039	0.2671	0.3499	0.3016	0.9442	0.9717	0.4635	-
IUPC*	0.7420	0.4800	0.5283	0.4979	0.5721	0.7564	0.3349	↓22.16%
DC-DGNN	0.7554	0.4290	0.4107	0.4016	0.7768	0.8814	0.3820	-
DC-DGNN*	**0.7983**	**0.6879**	**0.5853**	**0.5385**	**0.4206**	**0.6485**	**0.2575**	↓26.42%
Dataset: Appliances								
BiLSTM+Att	0.5000	0.1250	0.2500	0.1667	1.3250	1.1511	0.7250	-
BiLSTM+Att*	0.7250	0.6230	0.5417	0.5605	0.3500	0.5916	0.3000	↓48.61%
Bert-Sequence	0.7143	0.2381	0.3333	0.2778	1.0476	1.0235	0.4762	-
Bert-Sequence*	0.6042	0.2062	0.2404	0.2178	0.9583	0.9789	0.5417	↓4.36%
NGSAM	-	-	-	-	0.7007	0.8371	0.6010	-
NGSAM*	-	-	-	-	0.5069	0.7120	0.4763	↓14.94%
CHIM	0.5500	0.2244	0.2414	0.2230	0.9500	0.9747	0.6000	-
CHIM*	0.6000	0.2468	0.2614	0.2453	0.5500	0.7416	0.4500	↓23.92%
IUPC	0.7143	0.2381	0.3333	0.2778	1.0476	1.0235	0.4762	-
IUPC*	0.6042	0.2510	0.3212	0.2483	0.9583	0.9789	0.5417	↓4.36%
DC-DGNN	0.7143	0.2381	0.3333	0.2778	1.0476	1.0235	0.4762	-
DC-DGNN*	**0.8571**	**0.5441**	**0.5833**	**0.5625**	**0.6667**	**0.8165**	**0.2857**	↓20.22%
Dataset: Gift_Cards								
BiLSTM+Att	0.7439	0.2690	0.2034	0.1778	0.5213	0.7220	0.3273	-
BiLSTM+Att*	0.7749	0.3066	0.2631	0.2727	0.5201	0.7211	0.3079	↓0.12%
Bert-Sequence	0.8754	0.2189	0.2500	0.2334	0.3064	0.5535	0.1717	-
Bert-Sequence*	0.8443	0.4417	0.4053	0.4201	0.2909	0.5393	0.1905	↓2.57%
NGSAM	-	-	-	-	0.4152	0.6444	0.3463	-
NGSAM*	-	-	-	-	0.2544	0.5044	0.3116	↓21.73%
CHIM	0.8754	0.2189	0.2500	0.2334	0.3064	0.5535	0.1717	-
CHIM*	0.7671	0.3698	0.3346	0.3475	0.6080	0.7798	0.3312	↑40.89%
DC-DGNN	0.8754	0.2189	0.2500	0.2334	0.3064	0.5535	0.1717	-
DC-DGNN*	**0.8956**	**0.4384**	**0.5000**	**0.4671**	**0.1145**	**0.3383**	**0.1077**	↓38.88%

Table 2: Performance comparison of sentiment analysis models on streaming user reviews across three real-world Amazon datasets. ↓ indicates that lower values are better, while ↑ indicates that higher values are better. Models marked with * denote those trained on datasets augmented with the three types of synthesized data via the SYNGRAPH framework, and these rows are highlighted in blue. *RMSE Reduction* represents performance improvement, where ↓ indicates an performance improvement after data augmentation, while ↑ indicates a performance drop.

reveals a steady performance improvement after augmentation. This phenomenon suggests that the model demonstrates improved generalization and robustness, validating the ability of SYNGRAPH to synthesize diverse and effective user data.

4.3 Ablation Study

To validate the effectiveness of each proposed component, we conducted ablation experiments on DC-DGNN to assess the efficiency of data synthesis for each category, denoted as -M, -L, and -E for mid-tail, long-tail, and extreme users, respectively. As shown in Table 3, we found that combining data from all three categories generally resulted in the best performance across various datasets, such as Appliances and Gift_Cards. In some cases, using only one type of supplementation led to the optimal outcome, as observed in Magazine_Subscriptions. This is because the

Method	**Evaluation Metrics**						
	Accuracy (↑)	Precision (↑)	Recall (↑)	F1 (↑)	MSE (↓)	RMSE (↓)	MAE (↓)
				Dataset: Magazine_Subscriptions			
DC-DGNN*	0.7983	0.6879	0.5853	0.5385	0.4206	0.6485	0.2575
DC-DGNN-M	**0.8155**	0.6785	**0.6044**	**0.6225**	0.4292	0.6551	0.2489
DC-DGNN-L	0.7725	**0.6826**	0.5029	0.4661	0.5365	0.7324	0.2961
DC-DGNN-E	0.8112	0.5715	0.5521	0.5484	**0.4077**	**0.6385**	**0.2446**
				Dataset: Appliances			
DC-DGNN*	**0.8571**	**0.5441**	**0.5833**	**0.5625**	**0.6667**	**0.8165**	**0.2857**
DC-DGNN-M	0.7143	0.2381	0.3333	0.2778	1.0476	1.0235	0.4762
DC-DGNN-L	0.6667	0.2593	0.3111	0.2828	0.6190	0.7868	0.4286
DC-DGNN-E	0.7143	0.2381	0.3333	0.2778	1.0476	1.0235	0.4762
				Dataset: Gift_Cards			
DC-DGNN*	**0.8956**	**0.4384**	**0.5000**	**0.4671**	**0.1145**	**0.3383**	**0.1077**
DC-DGNN-M	0.8754	0.2234	0.2500	0.2359	0.1448	0.3805	0.1313
DC-DGNN-L	0.8754	0.2189	0.2500	0.2334	0.3064	0.5535	0.1717

Table 3: Results of interpolating sparse user data across different categories. DC-DGNN* represents the results obtained by inducing data from Mid-tail, Long-tail, and Extreme user categories. DC-DGNN-M refers to the results obtained by inducing data from only Mid-tail users, DC-DGNN-L from only Long-tail users, and DC-DGNN-E from only Extreme users.

datasets considered in this study are small-scale datasets, and introducing more data could introduce additional noise, potentially leading to a decrease in predictive performance. Similarly, there was no change in performance in the -M and -E cases of the Appliances dataset, likely due to the small number of synthesized data introduced. This is reasonable, as attempting to improve performance by introducing only a few data points, as shown in Table 1, is also unlikely. As for the -L case of the Gift_Cards dataset, overfitting still occurred, likely due to the severe imbalance of the original labels in this dataset, with proportions corresponding to labels 5, 4, 3, 2, and 1 being [0.9258, 0.0519, 0.0111, 0.0074, 0.0037] respectively. Introducing a large amount of similar data under the long-tail scenario exacerbated this imbalance. However, it is worth mentioning that the overfitting phenomenon during training on the Gift_Cards dataset was mitigated to some extent when combining the synthesized data from all three categories.

4.4 Evaluation of Synthesized Data

To assess the quality of the LLM-synthesized data, we conducted an evaluation using the latest GPT-4[1] model across four key metrics: (1) Language Style Similarity (LSS): Measures the alignment between the synthesized review and the user's writing style. (2) Rating Habit Similarity (RHS): Evaluates the consistency between the synthesized rating and the user's historical rating patterns. (3) Sentiment Similarity (SS): Assesses whether the sentiment of the synthesized review aligns with product reviews. (4) Aspect Similarity (AS): Determines whether the synthesized review focuses on relevant product attributes. Figure 4 presents the results across different sparsity categories and datasets. We observe that mid-tail users exhibit the highest consistency across all metrics, while extreme sparsity scenarios result in lower LSS and RHS scores due to limited user history. Even in sparse cases, SS and AS scores remain stable, indicating that the model effectively maintains

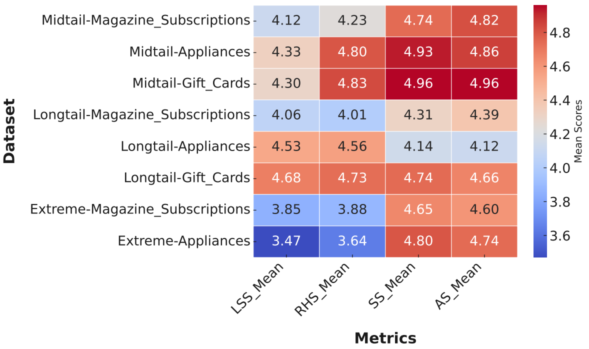


Figure 4: GPT-4 evaluation of synthesized reviews across different sparsity categories.

[1] https://openai.com/gpt-4

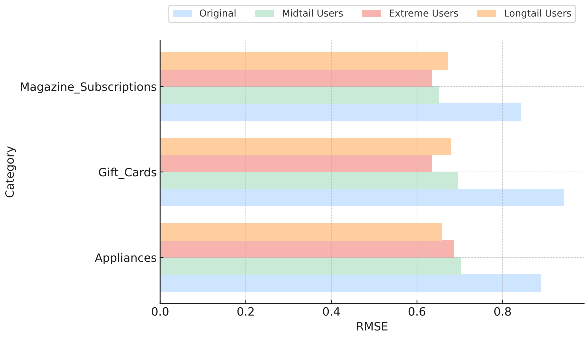


Figure 5: Comparison of vocabulary richness between LLM-synthesized and original data across different sparsity categories.

product-level coherence.

4.5 Vocabulary Richness Analysis

To evaluate the quality of the LLM-Synthesized data, we employ NLTK[2] to compute the overall average vocabulary richness across different sparsity categories. We then compare these values with those of the original dataset, as shown in Figure 5. Our analysis reveals that the LLM-synthesized text exhibits a vocabulary richness pattern consistent with prior findings (Li et al., 2023c), suggesting a potential limitation in lexical diversity. Specifically, across all sparsity categories and datasets, the vocabulary richness of the LLM-synthesized text is consistently lower than that of the original dataset. Furthermore, the richness levels remain relatively stable across different categories of synthesized data, indicating that while the LLM effectively synthesizes supplementary content, it may introduce a degree of lexical homogeneity.

5 Related Work

Data sparsity is a persistent challenge in e-commerce recommendation, where traditional methods enhance preference representations by leveraging user attributes (Volkovs et al., 2017; Zhu et al., 2020), integrating social networks (Sedhain et al., 2014; Du et al., 2022), or transferring knowledge across domains (Hu et al., 2018; Li and Tuzhilin, 2020; Gao et al., 2023; Zhu et al., 2021). Few-shot learning techniques further mitigate sparsity by leveraging semantic associations in graph-structured data (Wang et al., 2018; Zhou et al., 2023; Wang et al., 2023; Chen et al., 2022) or employing meta-learning to enhance adaptability across domains (Wu and Zhou, 2023; Lu et al., 2020; Lee et al., 2019). However, these approaches treat sparsity as a static issue, failing to model the temporal evolution of user interactions and the structural complexity of dynamic streaming data.

Recent advances in LLMs offer new opportunities to address sparsity by leveraging their reasoning abilities and extensive pretrained knowledge. Prompt engineering has been explored for mathematical QA (Yu et al., 2024), chain-of-thought reasoning (Liang et al., 2023), symbolic data generation (Ye et al., 2023), and hybrid human–LLM annotation strategies (Staab et al., 2024; Li et al., 2023a,b). In recommendation systems, LLMs have been employed to enrich sparse user data by integrating first-order neighbor information (Wei et al., 2023) or inferring user profiles from textual descriptions (Sun et al., 2023). However, these methods often overlook the structural complexity of graphs, limiting their effectiveness in leveraging higher-order relationships. Moreover, temporal dependencies are frequently ignored, leading to suboptimal modeling of evolving user preferences in streaming environments.

Despite these advancements, an effective framework that jointly models temporal dynamics, high-order structural dependencies, and LLM-based augmentation remains underexplored. Our work fills this gap by introducing a unified approach that systematically integrates these factors, offering a more comprehensive solution to the challenges in sentiment analysis on streaming user reviews.

6 Conclusion

We propose SynGraph, a dynamic graph-based framework that mitigates data sparsity in sentiment analysis on streaming reviews. By integrating LLM-augmented enhancements, it effectively models local and global structures, high-order relationships, and supplementary data. It adapts to different sparsity scenarios through a flexible decomposition and recombination mechanism. Experiments on three real-world Amazon datasets demonstrate its effectiveness in improving sentiment analysis on streaming reviews under sparse conditions.

Acknowledgement

We would like to thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by the British Heart Foun-

[2] https://www.nltk.org/

dation Manchester Research Excellence Award (RE/24/130017). We also acknowledge the CSF3 at the University of Manchester for providing GPU resources. Xin is supported by the UoM-CSC Joint Scholarship.

Limitations

While our data synthesis approach effectively mitigates user data sparsity and demonstrates strong performance, several aspects warrant further investigation:

- **Neighbor Selection Strategy:** To balance efficiency and computational cost, we adopt a random sampling strategy when selecting next-hop neighbors. While this method proves effective in our study, particularly for datasets with limited sample sizes, its impact may vary in large-scale scenarios where sampling variance becomes more pronounced. Future work could explore more principled selection mechanisms that optimize for both efficiency and representational quality.

- **LLM Understanding of Graph Structures:** Our framework integrates local and global graph structures to enhance representation learning. However, we do not explicitly analyze how LLMs differentiate between these two types of graphs or the extent to which each contributes to the final representations. A deeper investigation into LLMs' ability to process and leverage hierarchical graph structures could further improve the robustness of graph-based data augmentation.

- **Mitigating Hallucination in Data Synthesis:** Like many applications of LLMs, our approach is susceptible to hallucination, where synthesized data may not always faithfully reflect real-world distributions. While our framework benefits from LLM-augmented synthesis, ensuring the reliability and factual consistency of synthesized data remains an open challenge. Future research could focus on refining prompt designs or incorporating external validation mechanisms to enhance the trustworthiness of synthesized content.

Despite these considerations, our approach provides a solid foundation for addressing data sparsity in sentiment analysis on streaming reviews, and we believe that further refinements in these areas can further enhance its applicability.

References

Reinald Kim Amplayo. 2019. Rethinking attribute representation and injection for sentiment classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5601–5612. Association for Computational Linguistics.

Hao Chen, Zefan Wang, Feiran Huang, Xiao Huang, Yue Xu, Yishi Lin, Peng He, and Zhoujun Li. 2022. Generative adversarial framework for cold-start item recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 2565–2571. ACM.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1650–1659. The Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jing Du, Zesheng Ye, Lina Yao, Bin Guo, and Zhiwen Yu. 2022. Socially-aware dual contrastive learning for cold-start recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1927–1932. ACM.

Weibo Gao, Hao Wang, Qi Liu, Fei Wang, Xin Lin, Linan Yue, Zheng Zhang, Rui Lv, and Shijin Wang. 2023. Leveraging transferable knowledge concept graph embedding for cold-start cognitive diagnosis. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 983–992. ACM.

Guibing Guo. 2013. Improving the performance of recommender systems by alleviating the data sparsity and cold start problems. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 3217–3218. IJCAI/AAAI.

Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 667–676. ACM.

Young Ae Kim and Jaideep Srivastava. 2007. Impact of social influence in e-commerce decision making. In *Proceedings of the 9th International Conference on Electronic Commerce: The Wireless World of Electronic Commerce, 2007, University of Minnesota, Minneapolis, MN, USA, August 19-22, 2007*, volume 258 of *ACM International Conference Proceeding Series*, pages 293–302. ACM.

Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1073–1082. ACM.

Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy Chen, Zhengyuan Liu, and Diyi Yang. 2023a. CoAnnotating: Uncertainty-guided work allocation between human and large language models for data annotation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1487–1505, Singapore. Association for Computational Linguistics.

Moxin Li, Wenjie Wang, Fuli Feng, Yixin Cao, Jizhi Zhang, and Tat-Seng Chua. 2023b. Robust prompt optimization for large language models against distribution shifts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1539–1554, Singapore. Association for Computational Linguistics.

Pan Li and Alexander Tuzhilin. 2020. DDTCDR: deep dual transfer cross domain recommendation. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 331–339. ACM.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023c. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10443–10461. Association for Computational Linguistics.

Yuanyuan Liang, Jianing Wang, Hanlun Zhu, Lei Wang, Weining Qian, and Yunshi Lan. 2023. Prompting large language models with chain-of-thought for few-shot knowledge base question generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4329–4343, Singapore. Association for Computational Linguistics.

Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on heterogeneous information networks for cold-start recommendation. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1563–1573. ACM.

Chenyang Lyu, Jennifer Foster, and Yvette Graham. 2020. Improving document-level sentiment analysis with user and product context. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6724–6729. International Committee on Computational Linguistics.

Amit Mittal, Meenal Arora, and Ajay Rana. 2022. Imep: Influence maximization on social media with the impact of e-commerce products. In *5th International Conference on Contemporary Computing and Informatics, IC3I 2022, Uttar Pradesh, India, December 14-16, 2022*, pages 1789–1793. IEEE.

Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 188–197. Association for Computational Linguistics.

Suvash Sedhain, Scott Sanner, Darius Braziunas, Lexing Xie, and Jordan Christensen. 2014. Social collaborative filtering for cold-start recommendations. In *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*, pages 345–348. ACM.

Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. 2024. Beyond memorization: Violating privacy via inference with large language models. In *The Twelfth International Conference on Learning Representations*.

Chenkai Sun, Jinning Li, Yi Ren Fung, Hou Pong Chan, Tarek F. Abdelzaher, ChengXiang Zhai, and Heng Ji. 2023. Decoding the silent majority: Inducing belief augmented social graph with large language model for response forecasting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 43–57. Association for Computational Linguistics.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1014–1023. The Association for Computer Linguistics.

Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. 2017. Dropoutnet: Addressing cold start in recommender systems. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4957–4966.

Chunyang Wang, Yanmin Zhu, Aixin Sun, Zhaobo Wang, and Ke Wang. 2023. A preference learning decoupling framework for user cold-start recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 1168–1177. ACM.

Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 417–426. ACM.

Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Llmrec: Large language models with graph augmentation for recommendation. *CoRR*, abs/2311.00423.

Yuhao Wu, Karthick Sharma, Chun Seah, and Shuhao Zhang. 2023. Sentistream: A co-training framework for adaptive online sentiment analysis in evolving data streams. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6198–6212. Association for Computational Linguistics.

Zhenchao Wu and Xiao Zhou. 2023. M2EU: meta learning for cold-start recommendation via enhancing user preference estimation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 1158–1167. ACM.

Yang Xu, Lei Zhu, Jingjing Li, Fengling Li, and Heng Tao Shen. 2024. Temporal social graph network hashing for efficient recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3541–3555.

Jiacheng Ye, Chengzu Li, Lingpeng Kong, and Tao Yu. 2023. Generating data for symbolic language with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8418–8443, Singapore. Association for Computational Linguistics.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*.

Linhai Zhang, Xin Zhang, and Deyu Zhou. 2025. Relevant emotion ranking with topic-enhanced emotion transition. *Knowl. Based Syst.*, 311:113015.

Xin Zhang, Linhai Zhang, and Deyu Zhou. 2023. Sentiment analysis on streaming user reviews via dual-channel dynamic graph neural network. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7208–7220. Association for Computational Linguistics.

Xin Zhang, Linhai Zhang, Deyu Zhou, and Guoqiang Xu. 2024. Fine-grainedly synthesize streaming data based on large language models with graph structure understanding for data sparsity. *CoRR*, abs/2403.06139.

Deyu Zhou, Meng Zhang, Linhai Zhang, and Yulan He. 2021. A neural group-wise sentiment analysis model with data sparsity awareness. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14594–14601. AAAI Press.

Zhihui Zhou, Lilin Zhang, and Ning Yang. 2023. Contrastive collaborative filtering for cold-start item recommendation. *CoRR*, abs/2302.02151.

Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. 2021. Transfer-meta framework for cross-domain recommendation to cold-start users. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1813–1817. ACM.

Yuecai Zhu, Fuyuan Lyu, Chengming Hu, Xi Chen, and Xue Liu. 2022. Encoder-decoder architecture for supervised dynamic graph learning: A survey. *CoRR*, abs/2203.10480.

Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. 2020. Recommendation for new users and new items via randomized training and mixture-of-experts transformation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1121–1130. ACM.

A Experiment Details

Sparse User Categorization. Users with more than five reviews are considered non-sparse, while those with five or fewer are categorized as sparse based on clustering results. Table 4 shows the proportion of reviews contributed by these two groups, which together constitute the majority of reviews. Thus, we focus on these user types, where users with 0 to 5 reviews are classified as sparse (long-tail or extreme), and those with 5 to 10 reviews are non-sparse.

Mid-tail User Identification. Figure 6 further divides non-sparse users who exhibit temporal sparsity. We compute the number of reviews per user per day and apply K-means clustering using statistical indicators such as mean, standard deviation, minimum, and maximum review counts. The top-right region in the figure represents highly active users with fluctuating review frequencies, while the top-left region corresponds to similarly active users with more stable review patterns. The bottom-right region consists of less active users who occasionally review in bursts, and the bottom-left region contains users with consistently low review activity. Based on these distributions, we define mid-tail users as those in the top-right and bottom-right groups.

Long-tail and Extreme User Categorization. Figure 7 illustrates the distinction between long-tail and extreme users based on second-order neighborhood connectivity. Sparse users with limited self-data but sufficient second-order neighbors are classified as long-tail users, as their profiles can be supplemented with synthesized data. Conversely, users with both limited self-data and few second-order neighbors are categorized as extreme users, where direct data synthesis is required.

Temporal Distribution of Interpolated Data. In data interpolation, selecting insertion positions is crucial for synthesizing meaningful data. Figure 8 presents the temporal distribution of interpolated data across 10 time intervals.

B Prompt Templates

We use OpenAI's gpt-3.5-turbo[3] for data synthesis, employing specific prompts tailored for different user categories. In the mid-tail scenario, P_{um} (Figure 9) is used for user profile generation, P_{pm} (Figure 10) for product profiles, P_{so} (Figure 11) for selecting second-order products, and P_{sd} (Figure 12) for data synthesis. In the long-tail scenario, P_{ul} (Figure 13) is used for generating user profiles, while $\mathrm{P}_{pl} = \mathrm{P}_{pm}$ is applied to product profiles. The prompts P_{so} and P_{sd} remain the same as in the mid-tail setting. For extreme users, $\mathrm{P}_{ue} = \mathrm{P}_{um}$ is used for user profiles, $\mathrm{P}_{pe} = \mathrm{P}_{pm}$ for product profiles, and P_{sd} follows the same prompt as in mid-tail users.

C Synthetic-Data Proportion Experiments

We conduct a series of controlled experiments on three datasets (*Appliances*, *Gift_Cards*, *Magazine_Subscriptions*) using the DC-DGNN framework, in which synthetic data is inserted at six stages: 0% (NoSynth), Front 20%, Front 40%, Front 60%, Front 80%, and 100% (FullSynth).

Appliances (approximately 100 users) In this extremely sparse setting, adding a small amount of synthetic data (Front 20%) yields negligible changes in Accuracy or F1 and may introduce minor fluctuations due to noise. As the synthetic data ratio increases most metrics improve but Accuracy at Front 80% slightly dips, which indicates that moderate augmentation often outperforms both no augmentation and full augmentation in very small datasets (see Table 5).

Gift_Cards and Magazine_Subscriptions (larger scale) Both datasets show a generally monotonic increase in Accuracy and F1 as the synthetic data ratio increases, which suggests that when ample real data is available the negative impact of noise is reduced and the model benefits consistently from additional synthetic samples (see Tables 6 and 7).

Based on these results, we conclude that for very small and sparse datasets such as Appliances with only approximately 100 users moderate synthetic data ratios (for example 40 to 60%) achieve the best balance between performance improvement and noise. Therefore, the synthetic data ratio and insertion stage must be calibrated on a per dataset basis.

[3] https://platform.openai.com/docs/api-reference/models

Dataset	Total R	U10 R	U10 R proportion	U5 R	U5 R proportion
Magazine_Subscriptions	2330	1178	0.506	764	0.328
Appliances	203	87	0.429	116	0.571
Gift_Cards	2966	1502	0.506	1044	0.352

Table 4: Statistical analysis of the ratio of user-associated reviews to the total review count across various hierarchical levels. U10 R refers to the number of reviews associated with users with ten or fewer reviews. U5 R refers to the number of reviews associated with users with five or fewer reviews.

Variation	Accuracy	Precision	Recall	F1	MSE	RMSE	MAE
Appliances_0%_NoSynth	0.7143	0.2381	0.3333	0.2778	1.0476	1.0235	0.4762
Appliances_Front20%	0.7143	0.2381	0.3333	0.2778	1.0476	1.0235	0.4762
Appliances_Front40%	0.7619	0.4250	0.4778	0.4492	0.5238	0.7237	0.3333
Appliances_Front60%	0.7619	0.4556	0.5389	0.4889	0.7619	0.8729	0.3810
Appliances_Front80%	0.7143	0.4000	0.4556	0.4222	0.5714	0.7559	0.3810
Appliances_FullSynth	0.8571	0.5441	0.5833	0.5625	0.6667	0.8165	0.2857

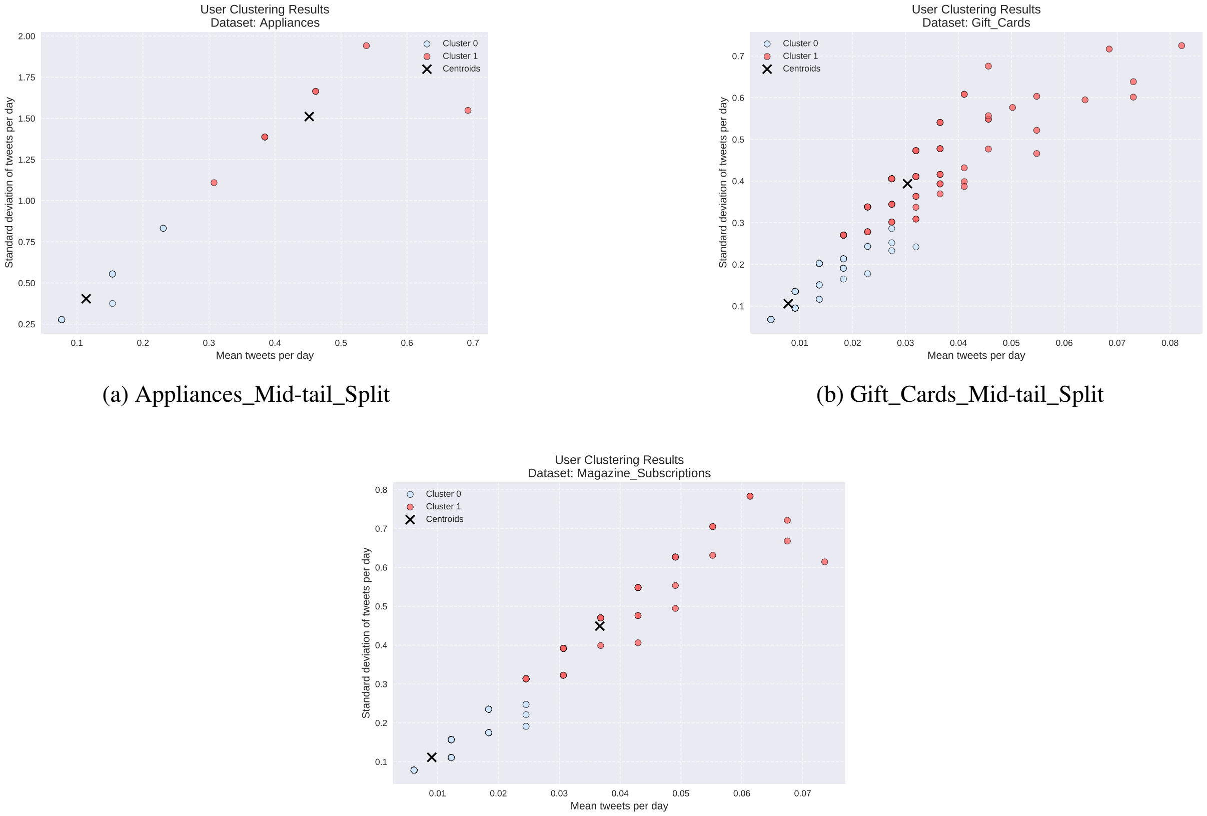
Table 5: Results on the Appliances dataset under different synthetic data proportions.

Variation	Accuracy	Precision	Recall	F1	MSE	RMSE	MAE
GiftCards_0%_NoSynth	0.8754	0.2189	0.2500	0.2334	0.3064	0.5535	0.1717
GiftCards_Front20%	0.8721	0.1780	0.1992	0.1880	0.1751	0.4184	0.1414
GiftCards_Front40%	0.8721	0.4732	0.2897	0.3064	0.1481	0.3849	0.1347
GiftCards_Front60%	0.8923	0.3787	0.3667	0.3706	0.1279	0.3577	0.1145
GiftCards_Front80%	0.8990	0.7241	0.5086	0.5030	0.1212	0.3482	0.1077
GiftCards_FullSynth	0.8956	0.4384	0.5000	0.4671	0.1145	0.3383	0.1077

Table 6: Results on the Gift_Cards dataset under different synthetic data proportions.

Variation	Accuracy	Precision	Recall	F1	MSE	RMSE	MAE
Magazine_0%_NoSynth	0.7554	0.4290	0.4107	0.4016	0.7768	0.8814	0.3820
Magazine_Front20%	0.7639	0.5851	0.4753	0.4830	0.5193	0.7206	0.3133
Magazine_Front40%	0.7768	0.4126	0.4794	0.4370	0.5064	0.7116	0.3004
Magazine_Front60%	0.8155	0.7670	0.6302	0.6210	0.3133	0.5597	0.2189
Magazine_Front80%	0.8155	0.5576	0.5495	0.4991	0.3906	0.6249	0.2361
Magazine_FullSynth	0.7983	0.6879	0.5853	0.5385	0.4206	0.6485	0.2575

Table 7: Results on the Magazine_Subscriptions dataset under different synthetic data proportions.

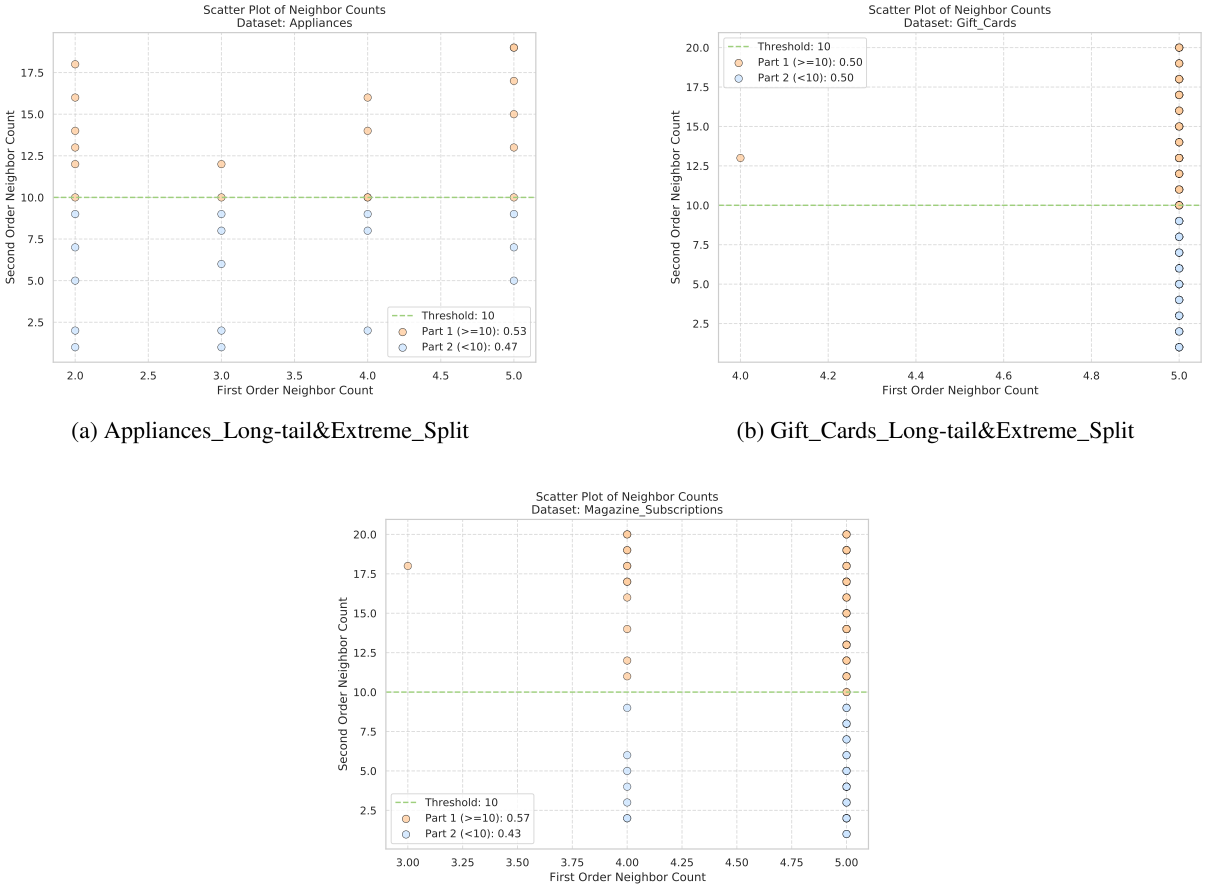


(a) Appliances_Mid-tail_Split

(b) Gift_Cards_Mid-tail_Split

(c) Magazine_Subscriptions_Mid-tail_Split

Figure 6: Non-Data Sparse User Division. This section discusses users who are sparse in time rather than in data. The data points in the upper right corner indicate users with abundant but uneven data. The red dots in the figure are defined as mid-tail users.

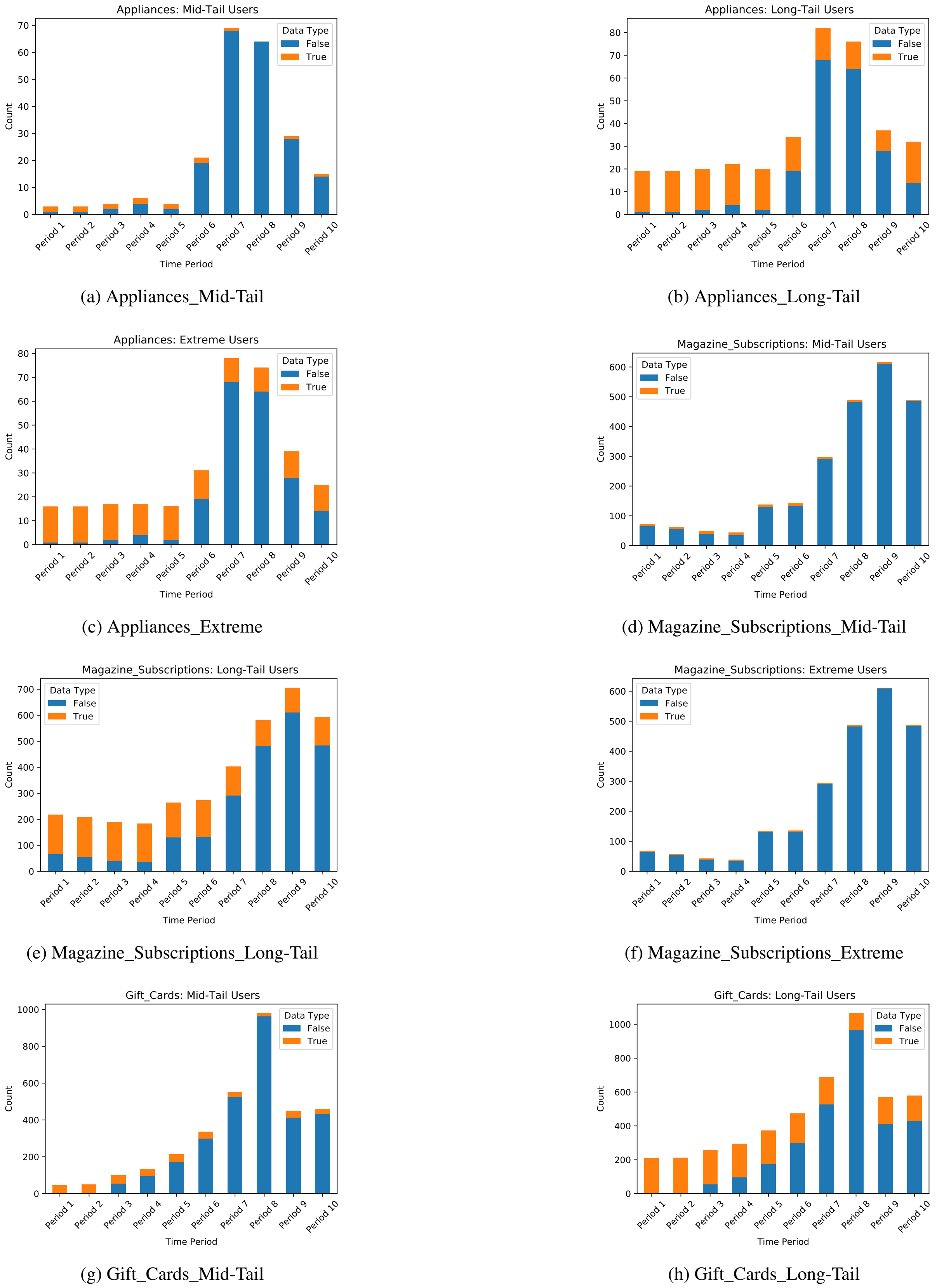


(a) Appliances_Long-tail&Extreme_Split

(b) Gift_Cards_Long-tail&Extreme_Split

(c) Magazine_Subscriptions_Long-tail&Extreme_Split

Figure 7: Data-Sparse User Division and Corresponding Proportions. The orange points exhibit abundant second-order homogeneous relationships and are defined as long-tail users, while the blue points have sparse second-order homogeneous relationships and are defined as extreme cases.



(a) Appliances_Mid-Tail

(b) Appliances_Long-Tail

(c) Appliances_Extreme

(d) Magazine_Subscriptions_Mid-Tail

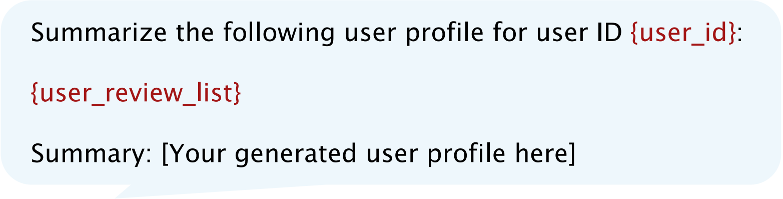
(e) Magazine_Subscriptions_Long-Tail

(f) Magazine_Subscriptions_Extreme

(g) Gift_Cards_Mid-Tail

(h) Gift_Cards_Long-Tail

Figure 8: Distribution of interpolation positions along the timeline corresponding to different sparse categories across datasets.

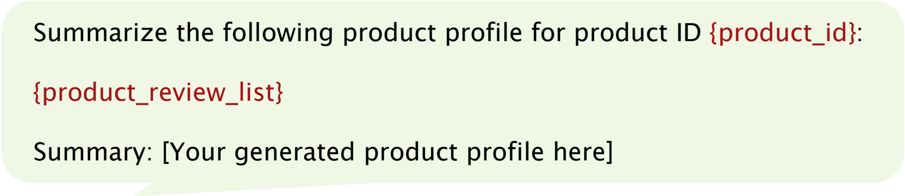


```
Summarize the following user profile for user ID {user_id}:

{user_review_list}

Summary: [Your generated user profile here]
```

Figure 9: The prompt used for generating user profiles in the mid-tail and extreme scenarios, defined as P_{um} and P_{ue} in the paper, respectively, takes as input the selected reviews of the user.

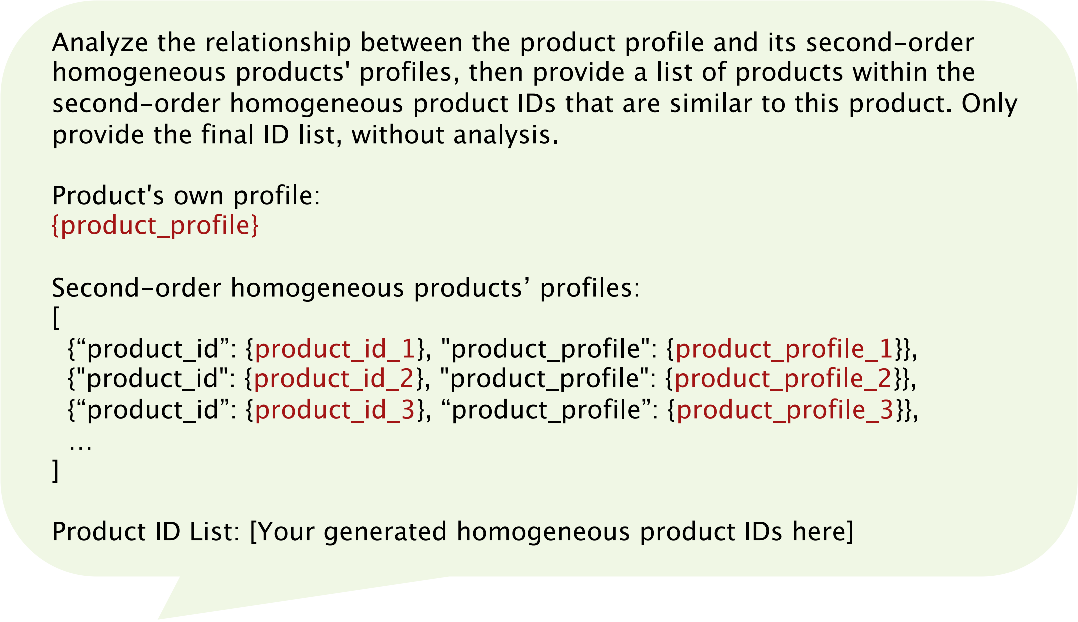


```
Summarize the following product profile for product ID {product_id}:

{product_review_list}

Summary: [Your generated product profile here]
```

Figure 10: The prompt used for generating product profiles in the mid-tail, long-tail, and extreme scenarios, defined as P_{pm}, P_{pl}, and P_{pe} in the paper, respectively, takes as input the selected reviews of the product.



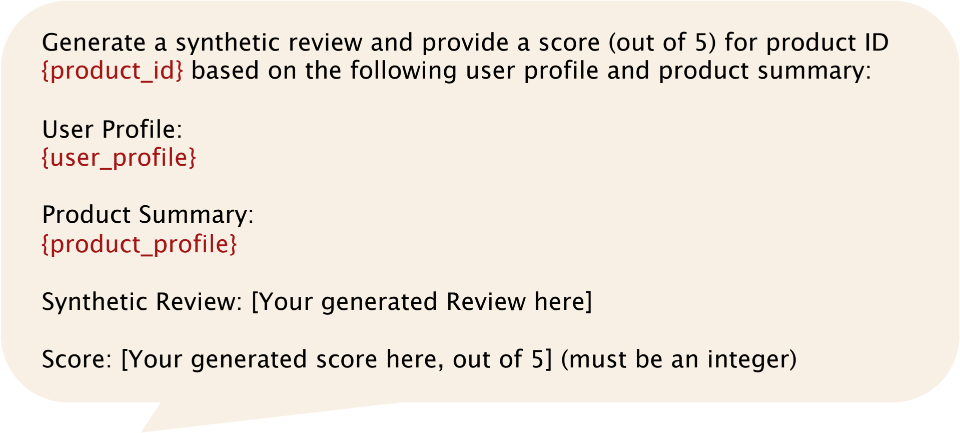
```
Analyze the relationship between the product profile and its second-order
homogeneous products' profiles, then provide a list of products within the
second-order homogeneous product IDs that are similar to this product. Only
provide the final ID list, without analysis.

Product's own profile:
{product_profile}

Second-order homogeneous products’ profiles:
[
  {“product_id”: {product_id_1}, "product_profile": {product_profile_1}},
  {"product_id": {product_id_2}, "product_profile": {product_profile_2}},
  {“product_id”: {product_id_3}, “product_profile”: {product_profile_3}},
  ...
]

Product ID List: [Your generated homogeneous product IDs here]
```

Figure 11: The prompt used for selecting second-order homogeneous products in the mid-tail and long-tail scenarios, defined as P_{so} in the paper, takes as input the profile of the product itself along with the profile of the second-order homogeneous products.



Generate a synthetic review and provide a score (out of 5) for product ID {product_id} based on the following user profile and product summary:

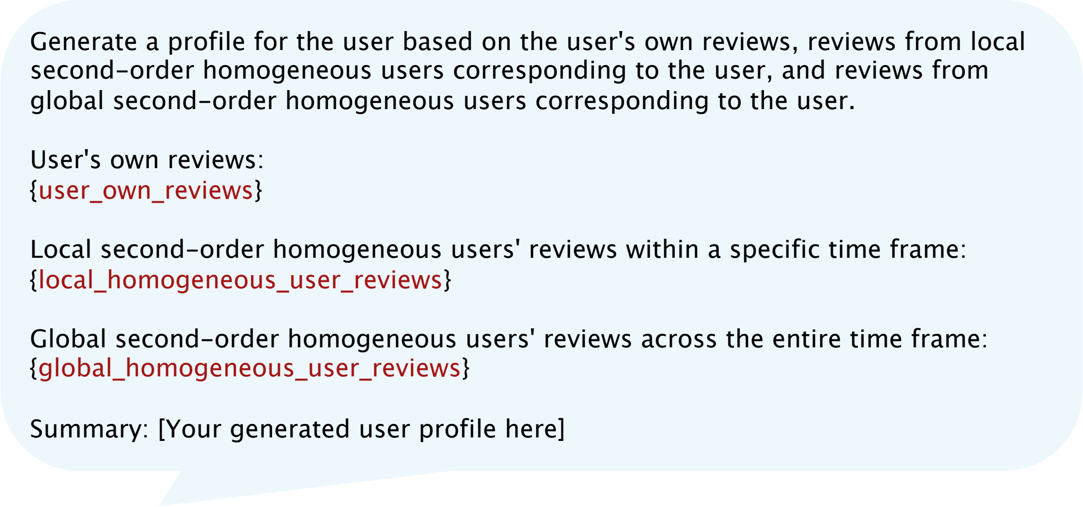
User Profile:
{user_profile}

Product Summary:
{product_profile}

Synthetic Review: [Your generated Review here]

Score: [Your generated score here, out of 5] (must be an integer)

Figure 12: The prompt used for inducing data for mid-tail, long-tail, and extreme user scenarios, defined as P_{sd} in the paper, takes as input the user profile and the product profile.



Generate a profile for the user based on the user's own reviews, reviews from local second-order homogeneous users corresponding to the user, and reviews from global second-order homogeneous users corresponding to the user.

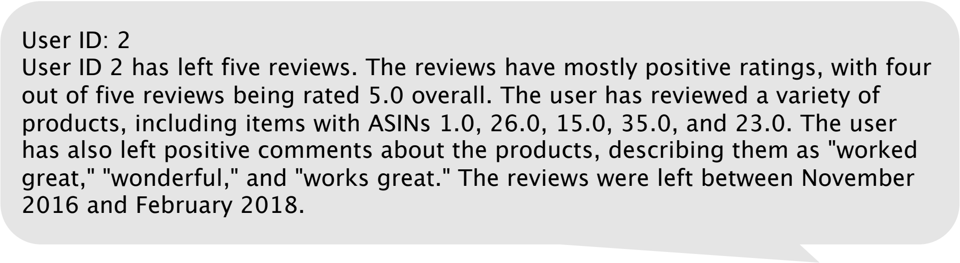
User's own reviews:
{user_own_reviews}

Local second-order homogeneous users' reviews within a specific time frame:
{local_homogeneous_user_reviews}

Global second-order homogeneous users' reviews across the entire time frame:
{global_homogeneous_user_reviews}

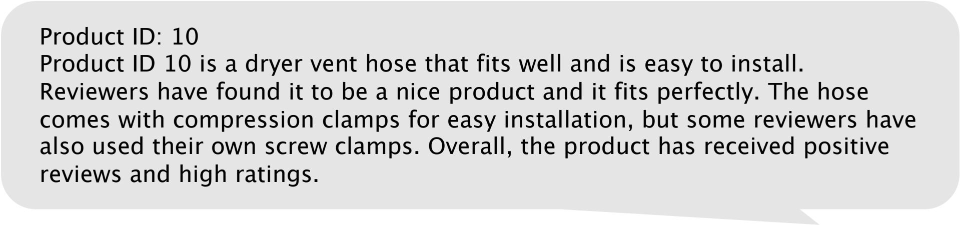
Summary: [Your generated user profile here]

Figure 13: The prompt used for understanding the local and global user relationships in the long-tail scenario, defined as P_{ul} in the paper, takes as input the user's own reviews, the reviews of locally second-order homogeneous users, and the reviews of globally second-order homogeneous users.



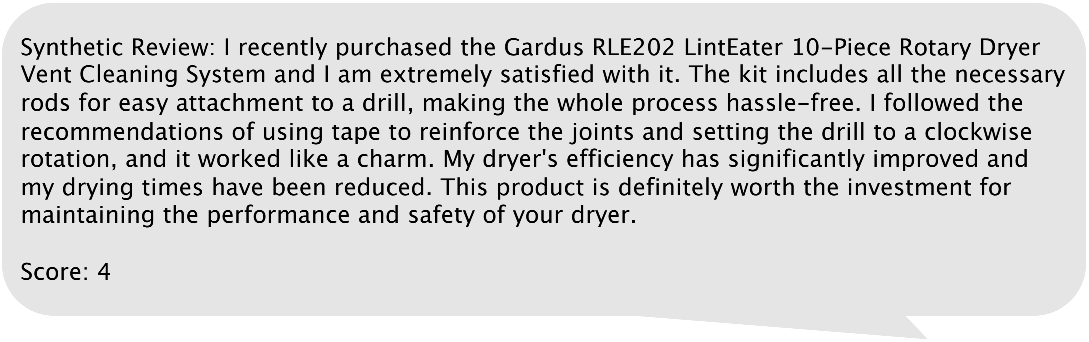
User ID: 2
User ID 2 has left five reviews. The reviews have mostly positive ratings, with four out of five reviews being rated 5.0 overall. The user has reviewed a variety of products, including items with ASINs 1.0, 26.0, 15.0, 35.0, and 23.0. The user has also left positive comments about the products, describing them as "worked great," "wonderful," and "works great." The reviews were left between November 2016 and February 2018.

Figure 14: Example of user profiles generated by GPT.



Product ID: 10
Product ID 10 is a dryer vent hose that fits well and is easy to install. Reviewers have found it to be a nice product and it fits perfectly. The hose comes with compression clamps for easy installation, but some reviewers have also used their own screw clamps. Overall, the product has received positive reviews and high ratings.

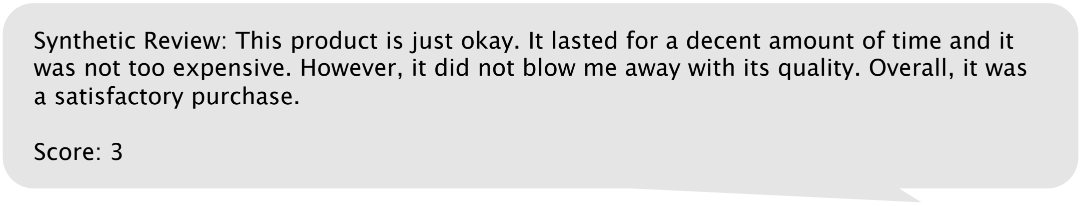
Figure 15: Example of product profiles generated by GPT.



Synthetic Review: I recently purchased the Gardus RLE202 LintEater 10-Piece Rotary Dryer Vent Cleaning System and I am extremely satisfied with it. The kit includes all the necessary rods for easy attachment to a drill, making the whole process hassle-free. I followed the recommendations of using tape to reinforce the joints and setting the drill to a clockwise rotation, and it worked like a charm. My dryer's efficiency has significantly improved and my drying times have been reduced. This product is definitely worth the investment for maintaining the performance and safety of your dryer.

Score: 4

Figure 16: Example of synthesized data demonstrating positive sentiment generated by GPT.



Synthetic Review: This product is just okay. It lasted for a decent amount of time and it was not too expensive. However, it did not blow me away with its quality. Overall, it was a satisfactory purchase.

Score: 3

Figure 17: Example of synthesized data demonstrating neutral sentiment generated by GPT.