DATA-CENTRIC APPROACHES FOR ADVANCED AND EFFICIENT RECOMMENDATION MODELING

JIAHAO WU

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University
Department of Computing

Data-Centric Approaches for Advanced and Efficient Recommendation Modeling

Jiahao Wu

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
May 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Jiahao Wu

Abstract

Recommender systems have become one of the most successful applications of artificial intelligence's in daily life, significantly relieving people from overwhelming information load and assisting users in decision-making by providing personalized recommendations. Advanced recommendation modeling techniques have been deployed across various platforms, i.e., Tiktok in video recommendations, Linkedin in friend recommendations, Goodreads for book recommendations, and Taobao for item recommendations.

Despite their widespread success, existing recommendation approaches face several limitations, particularly in terms of multi-faceted user modeling and scalability. For instance, these models often experience performance degradation when handling multi-faceted user data. Additionally, many state-of-the-art recommendation models are trained on large-scale datasets, making the training process both computationally expensive and inefficient, especially when fine-tuning hyperparameters or architectures to optimize recommendation performance. To address these challenges, this dissertation adopts a data-centric perspective, proposing curated data manipulation and optimization strategies to enhance the performance of recommendation models across different scenarios.

Firstly, we demonstrate that disentangling various facets of data (i.e., social data and user-item data) for independent modeling can significantly improve recommendation performance. Furthermore, we explore the potential to substantially reduce the size of

user-item interaction and content data in recommendation datasets while preserving critical information, thus markedly decreasing training costs. Different from model-centric approaches that are often tailored to specific recommendation models, data-centric strategies generate enhanced datasets that offer benefits across a range of existing models. By addressing key challenges related to personalized modeling and computational efficiency from a perspective of data-centric, this dissertation lays the foundation for next-generation recommendation techniques.

Publications Arising from the Thesis

1. Jiahao Wu, Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qing Li, Ke Tang, "Disentangled Contrastive Learning for Social Recommendation", in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*, 2022. (corresponding to Chapter 2)

2. Jiahao Wu, Wenqi Fan, Shengcai Liu, Qijiong Liu, Qing Li, Ke Tang, "Enhancing Graph Collaborative Filtering via Uniformly Co-Clustered Intent Modeling", Arxiv'2023.

3. Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qijiong Liu, Rui He, Qing Li, and Ke Tang,"Condensing Pre-augmented Recommendation Data via Lightweight Policy Gradient Estimation", *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2024. (corresponding to Chapter 3)

4. Jiahao Wu, Qijiong Liu, Hengchang Hu, Wenqi Fan, Shengcai Liu, Qing Li, Xiao-Ming Wu, Ke Tang, "Leveraging Large Language Models (LLMs) to Empower Training-Free Dataset Condensation for Content-Based Recommendation", in *Companion Proceedings of the ACM Web Conference (TheWebConf)*, 2025. (corresponding to Chapter 4)

5. Jiahao Wu, Ning Lu, Zeiyu Dai, Wenqi Fan, Shengcai Liu, Qing Li, Ke Tang,

"Backdoor Graph Condensation", to appear in *Proceedings of IEEE International Conference on Data Engineering (ICDE)*, 2025.

6. Rui He, Shengcai Liu, Jiahao Wu, Shan He, Ke Tang, "Multi-Domain Learning from Insufficient Annotations", in *Proceedings of the 26th European Conference on Artificial Intelligence* (2023).

7. Qijiong Liu, Jieming Zhu, Jiahao Wu, Tiandeng Wu, Zhenhua Dong, and Xiao-Ming Wu, "FANS: Fast Non-Autoregressive Sequence Generation for Item List Continuation", in *Proceedings of the ACM Web Conference (TheWebConf)*, 2023.

8. Qijiong Liu, Hengchang Hu, Jiahao Wu, Jieming Zhu, Min-Yen Kan, and Xiao-Ming Wu, "Discrete Semantic Tokenization for Deep CTR Prediction", in *Companion Proceedings of the ACM Web Conference (TheWebConf)*, 2024.

Acknowledgments

First and foremost, I would like to express my profound gratitude to my supervisors, Prof. Qing Li and Prof. Ke Tang, for their invaluable guidance, inspiration, encouragement, and unwavering support throughout my Ph.D. journey. Their patience, meticulous attention to detail, professionalism, and humility have profoundly shaped both my personal growth and career aspirations. They have cultivated an open and intellectually stimulating environment, which has encouraged me to explore meaningful and challenging research topics. I will always cherish the thoughtful conversations with them, during which they provided not only expert advice but also heartfelt encouragement during the uncertain early stages of my Ph.D. Without their constant support and mentorship, the completion of this thesis would not have been possible.

I would also like to extend my sincere thanks to my mentors, Dr. Wenqi Fan and Dr. Shengcai Liu, for their expert guidance in research. Their insightful suggestions on topic selection, the formulation of rigorous ideas, and the refinement of academic writing have been instrumental in shaping my research skills. Dr. Wenqi Fan, in particular, introduced me to the field of data mining, provided me with the essential skills to conduct high-quality research, and consistently encouraged me to strive for excellence and aim high in my endeavors. I am equally grateful to Dr. Shengcai Liu for his thoughtful guidance on academic writing and his valuable advice on effective collaboration.

Throughout my Ph.D. studies, I have been fortunate to engage in fruitful discussions

and collaborations that have significantly influenced my academic development and career trajectory. I am especially grateful to my collaborators Ning Lu, Zhiyuan Wang, Rui He, Zeiyu Dai, Da Ren, Jingfan Chen, Xiao Chen, Lin Wang, Shijie Wang, Qijiong Liu, Kun Wang, and Bo Tang for their contributions to my work. I would also like to thank my colleagues Xuanfeng Li, Shaofeng Zhang, Haoze Lv, Zehang Lin, and Changmeng Zheng for their camaraderie.

Finally, I would like to express my deepest gratitude to my family—my parents, Yanlin Wu and Yangqin Le, and my sister, Jiaxin Wu. Their unconditional love, encouragement, and belief in me have provided me with constant strength and motivation throughout this journey.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In contemporary online service platforms, recommender systems have become ubiquitous, powering applications ranging from LinkedIn for job searches and Scholar for academic paper alerts to Taobao for e-commerce and TikTok for video streaming. The primary goal of these systems is to leverage user and item background information, as well as user interaction histories, to model personalized user preferences. This enables the generation of accurate and tailored recommendations, thereby enhancing user engagement and facilitating the discovery of relevant information. To achieve promising recommendation performance, existing approaches incorporate large-scale datasets that include extensive interaction data and supplementary information about users and items to train advanced models [138, 127, 153, 154, 69, 29].

Despite the success of current recommendation models, they face notable challenges, particularly with regard to multi-faceted user modeling and scalability [140, 141, 134, 88, 30, 129, 128]. For instance, many existing methods combine different types of data (e.g., user social data and user-item interaction data) to train models [140, 141, 30, 129, 128], which often fails to capture the full complexity of multi-faceted user

personalization, resulting in sub-optimal recommendation performance. Moreover, training recommendation models on large-scale data is computationally demanding and inefficient, especially when fine-tuning hyperparameters or model architectures to optimize recommendation quality. To mitigate these challenges, considerable efforts have been devoted to develop novel architectures or training losses, all of which adopt a model-centric perspective. However, model-centric approaches ignore the potential benefits of directly optimizing the data itself, despite the fact that both data and models are integral to building high-performance recommender systems. This gap leads to the central question of this dissertation: *Can we boost recommendation performance and efficiency from a data perspective?*

Answering this question requires to explore data-centric methods within the broader field of Data-Centric AI [143, 142]. Rather than focusing on developing new model architectures, we propose the design of data synthesis or data optimization techniques tailored to existing datasets, aimed at improving the effectiveness and efficiency of current models. This innovative learning paradigm integrates data optimization into the overall recommendation system development process, which has the potential to significantly improve both performance and efficiency. The advantage of data-centric methods lies in their ability to optimize data in ways that benefit a variety of recommendation models. Given existing promises, the advancement of data-centric approaches for recommendation systems holds the promise of becoming a key driver for the next phase of performance and efficiency improvement in recommender systems.

1.2 Contributions

To broaden the scope of recommendation research beyond model development, this dissertation emphasizes the critical role of data processing and optimization in enhancing model training. Specifically, we introduce techniques for optimizing datasets and improving data utilization during the training of recommendation models. On

the one hand, we propose one method on enhancing the modeling of different facets of users in social recommendation. On the other hand, we propose two approaches to reduce the size of ID-based and content-based recommendation datasets, while preserving their core information, thereby resulting in significant reductions in training costs. The main contributions of this dissertation are summarized as follows:

- As suggested by social correlation theories [13], users' preferences are often influenced by those around them. Building on this insight, numerous studies have leveraged social relationship information to model user preferences for items, thereby enhancing recommendation performance [140, 30, 129], a field known as social recommendation [141, 128, 127]. However, existing approaches fail to effectively model the multifaceted nature of users' personalities across different domains (i.e., user-item collaborative and user-user social domains). To address this limitation, we propose DcRec [121], a method that disentangles user behaviors across these domains, enabling independent modeling of users' distinct personality facets. Moreover, to effectively integrate personality information from both domains to enhance recommendation performance, we introduce a cross-domain contrastive module that facilitates knowledge transfer from the social domain to the collaborative domain.

- Although recommendation models trained on the large-scale user-item historical interaction sets [31, 111, 153, 127] have demonstrated the ability to provide personalized recommendations across various websites (e.g., Amazon, Taobao), the training process is computationally expensive. To address this issue, we propose DConRec [123] to synthesize a small user-item interaction set that can approximate similar training performance to the models trained on the original dataset. This is accomplished by learning key interactions through optimization of the recommendation task loss. Additionally, we provide a theoretical analysis to ensure the convergence of DConRec.

- While DConRec shows promise, it is limited in its ability to process the side information of users and items. Moreover, its condensation process is computationally intensive due to the optimization involved. To overcome these challenges, we extend dataset condensation techniques to content-based recommendation systems, where items are enriched with textual content. We propose TF-DCon [124], a method that condenses both user interactions and item content. In TF-DCon, we incorporate large language models (e.g., ChatGPT) to process textual content and design two training-free condensation modules to condense item content and users' interaction histories.

1.3 Structure of This Dissertation

This dissertation is structured as follows. In Chapter 2, we aims propose a disentangled contrastive learning framework for data from different domains, where both the personalities of users from collaborative domain and social domain can be accurately modeled and incorporated for better user modeling. In Chapter 3, we formly define the problem of dataset condensation for recommendation data (i.e., user-item interaction set) and present a framework to address this difficult problem, which condenses the large interaction set into a small synthetic one which can preserve main information for recommendation model training. In Chapter 4, we extend the condensation design into content-based recommendation, where we propose a training-free framework to simultaneously condense the user interaction histories and item contents. In Chapter 5, we conclude the dissertation and discuss the impact and promising research directions.

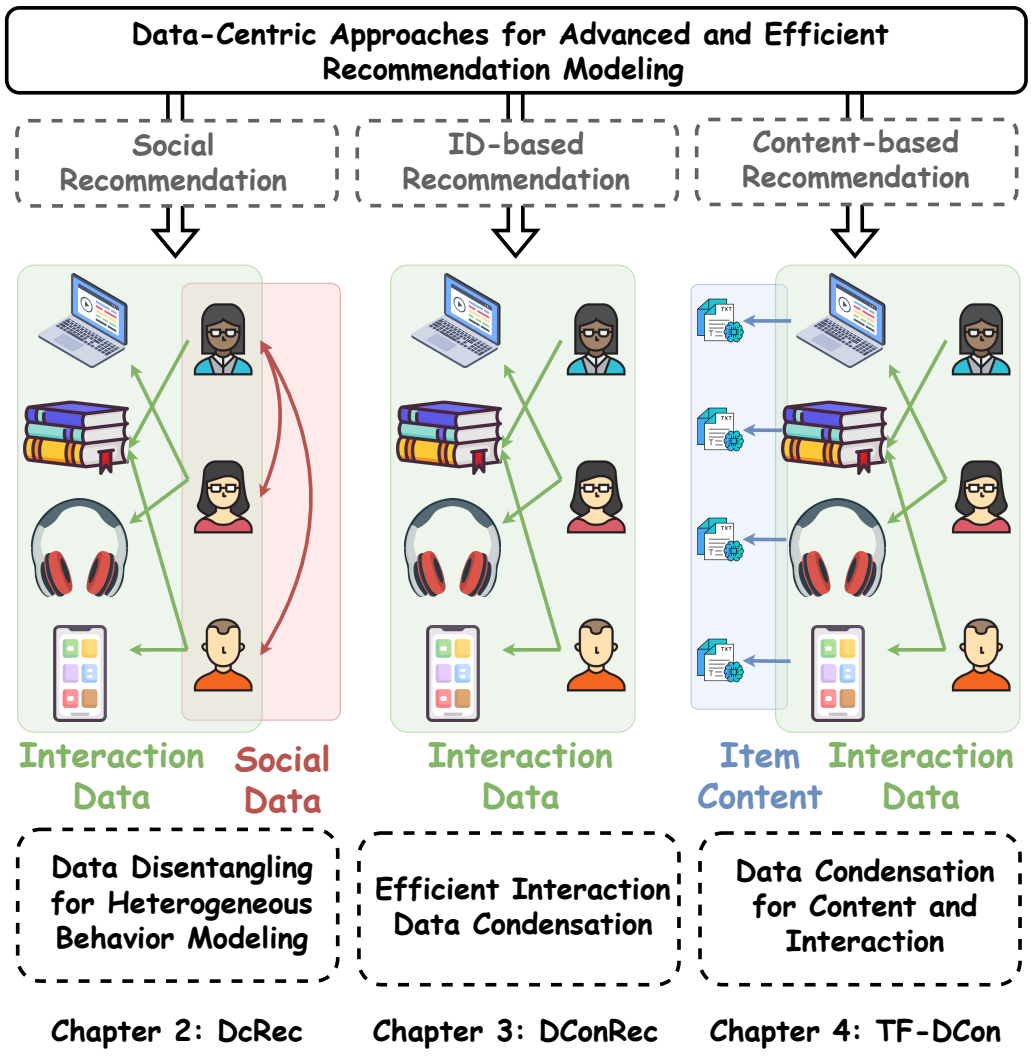


Figure 1.1: Three novel data-centric methods for effective and efficient recommendation.

Chapter 2

Disentangled contrastive learning for social recommendation

Social recommendations utilize social relations to enhance the representation learning for recommendations.Most social recommendation models unify user representations for the user-item interactions (collaborative domain) and social relations (social domain). However, such an approach may fail to model the users' heterogeneous behavior patterns in two domains, impairing the expressiveness of user representations. In this work, to address such limitation, we propose a novel **D**isentangled **c**ontrastive learning framework for social **Rec**ommendations (**DcRec**). More specifically, we propose to learn disentangled users' representations from the item and social domains. Moreover, disentangled contrastive learning is designed to perform knowledge transfer between disentangled users' representations for social recommendations. Comprehensive experiments on various real-world datasets demonstrate the superiority of our proposed model. Our implementation is available at: `https://github.com/JiahaoWuGit/DcRec`.

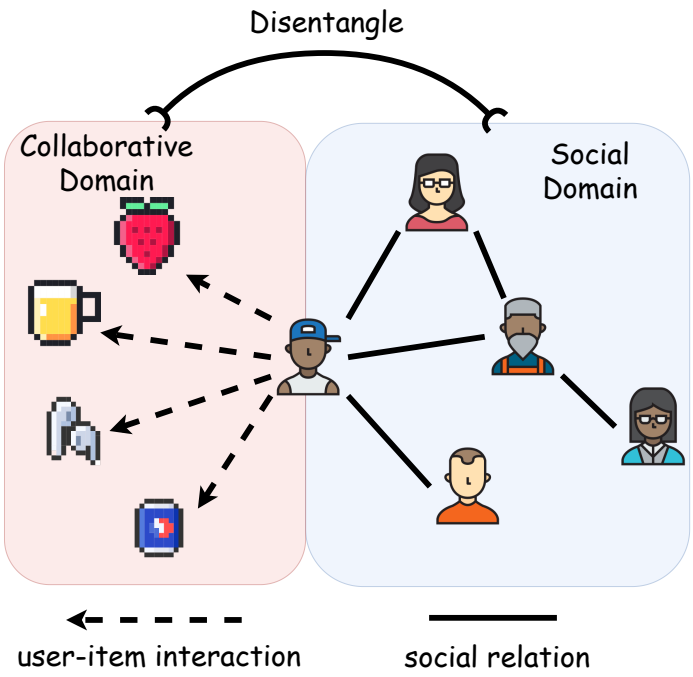


Figure 2.1: Users are involved in both collaborative domain and social domain. The behavior patterns of users in two domains are semantically heterogeneous, where a user can have his/her personalized preferences towards items in the collaborative domain while connecting with social friends in the social domain.

2.1 Introduction

As suggested by social correlation theories [13], users' preferences are likely to be influenced by those around them. Based on this intuition, a bunch of works have been proposed to utilize the information of social relations in modeling users' preferences for items to enhance recommendation performance in various online platforms (e.g., Facebook, WeChat, LinkedIn, etc.) [24, 80, 39], known as social recommendations [31, 30, 129, 128, 28].

In social recommendation, as shown in Figure 2.1, users are interacted with different objectives (i.e., items and social friends) with distinct purposes in each domain (i.e.,

collaborative domain and social domain) [24]. Therefore, the behavior patterns of users in two domains can be heterogeneous. In a real scenario, a user tends to connect with friends of his/her friends but he/she are not likely to purchase the items serving similar functions in a short period of time. However, off-the-shelf manners on modeling users' preferences adopt unified users' representations for user-item interactions and user-user social relationships. For instance, Diffnet++ [129] utilizes a node-level attention layer to fuse the users' representations from social domain and collaborative domain. DSCF [32] uses unified users' representations to capture social information and user-item interaction information. They are insufficient to model users' heterogeneous behavior patterns towards social friends and items in social recommendations, which may undermine the representation learning in each domain.

To address this problem, in this work, we propose to *disentangle* user behaviors into two domains, so as to learn disentangled users' representations in the social recommendations task, instead of the unified users' representations. The main challenge is how to learn such disentangled users' representations in two domains while transferring knowledge from social domain to collaborative domain for social recommendations.

Recently, Self-Supervised Learning (SSL), a prevalent novel learning paradigm, has been proven beneficial to the tasks in a wide range of fields [7, 20, 100, 44, 130, 61]. The main idea of SSL is to utilize self-supervision signals from unlabeled data by maximizing the mutual information between different augmented views of the same instance (i.e., user or item) [109], in the representation learning process of which SSL increases the informativeness of those views and enables knowledge transferring between those views [141]. For instance, SGL [126] devises various auxiliary tasks to learn the structure relatedness knowledge between different views and utilize it to supplement supervision signals. MHCN [141] constructs different hyper-graphs and maximizes the mutual information between the users and the hyper-graphs to learn the hierarchical structure information and transfer it into the learned nodes' representations.

Motivated by the advantage of SSL in transferring knowledge, we develop a novel contrastive learning-based framework to solve the aforementioned issue. More specifically, domain disentangling is introduced to disentangle users' behaviors into collaborative domain and social domain. Moreover, we propose disentangled contrastive learning objectives to transfer knowledge from social domain to collaborative domain by maximizing the mutual information between disentangled representations. Notably, while DGCL [61] proposes an implicitly disentangled contrastive learning method to capture multiple aspects of graphs, we explicitly disentangle the data from different domains and propagate the representations of nodes based on independent adjacency matrices. The main contributions of the method described in this chapter can be summarized as follows:

- We introduce a principled approach to learn users' representations, where disentangled users' representations can be learned to reflect their preferences towards items and social friends in two domains.
- We propose a novel **D**isentangled **c**ontrastive learning framework for social **Rec**omm-endations (**DcRec**), which can harness the power of contrastive learning to transfer knowledge from social domain to collaborative domain.
- We conduct comprehensive experiments on real-world datasets to show the superiority of the proposed model and study the effectiveness of each module.

2.2 Methodology

We first introduce definitions and notations used in this paper. We utilize $\mathcal{U}$ to stand for the user set and $\mathcal{I}$ to stand for the item set. Let $m = |\mathcal{U}|$ defines the number of users and $n = |\mathcal{I}|$ denotes the number of items. Then, we denote the user-item interactions matrix in the collaborative domain by $\mathbf{A}_I \in \mathbb{R}^{m \times n}$ and social relations matrix in the social domain by $\mathbf{A}_S \in \mathbb{R}^{m \times m}$. In addition, we use dense

vectors to represent users and items (i.e., embeddings), where $\mathbf{P}_S \in \mathbb{R}^{m \times d}$ and $\mathbf{P}_I \in \mathbb{R}^{m \times d}$ denote users' embeddings with d dimension in social domain and collaborative domain, respectively. $\mathbf{Q}_I \in \mathbb{R}^{n \times d}$ denotes items' embeddings in the collaborative domain.

2.2.1 An Overview of the Proposed Framework

In this work, we propose a **D**isentangled **c**ontrastive learning framework for social **Rec**ommendations (**DcRec**), which follows the general paradigm of self-supervised contrastive learning via maximizing the representation agreement between different views on the same instance [126, 7, 51].

The architecture of the proposed model is shown in Figure 2.2. More specifically, the proposed architecture consists of three main components: (1) **Domain Disentangling**, which is devised to disentangle the input data into two sub-domains; (2) **Encoder**, where we use different encoders on two domains to learn representations from two different views; (3) **Disentangled Contrastive Learning**, which aims to transfer the knowledge from the social domain into recommendation modeling task by jointly optimizing the disentangled contrastive learning tasks and main recommendation task.

2.2.2 Domain Disentangling

To mitigate the influence caused by the semantic discrepancy between social domain and collaborative domain, we disentangle the input data into two domains, which will be represented by a user-item interactions matrix $\mathbf{A}_I$ in the collaborative domain and social relations matrix $\mathbf{A}_S$ in the social domain, respectively.

After domain disentanglement, we perform data augmentation to obtain different views for the data in each domain. Since the data in social recommendations can be

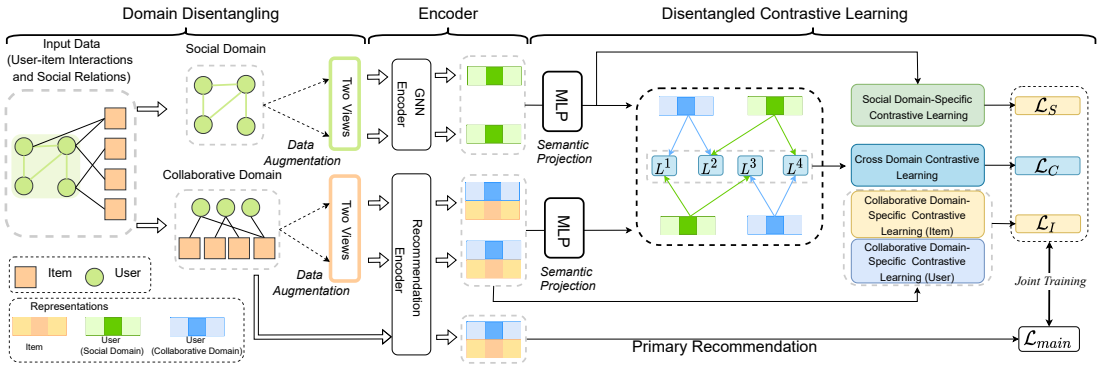


Figure 2.2: The overall architecture of the proposed disentangled contrastive learning for social recommendations (DcRec).

naturally represented as graph [30], the inputs (i.e., user-item interactions $\mathbf{A}_I$ and social relations $\mathbf{A}_S$) can be augmented via graph-based data augmentation methods [137, 51], such as *Edge Adding*, *Edge Dropout*, and *Node Dropout*, which can be formulated as follows:

$$\mathbf{A}_S^{(1)} = H_S^{(1)}(\mathbf{A}_S), \mathbf{A}_S^{(2)} = H_S^{(2)}(\mathbf{A}_S),$$
$$\mathbf{A}_I^{(1)} = H_I^{(1)}(\mathbf{A}_I), \mathbf{A}_I^{(2)} = H_I^{(2)}(\mathbf{A}_I),$$

where $H_S^{(\cdot)}$ and $H_I^{(\cdot)}$ denote the independent augmentation functions to generate two views in social domain and collaborative domain, respectively.

2.2.3 Encoder

To model the user-item interactions and social relations, we utilize encoders to learn representations for users and items in each domain. Furthermore, to ensure semantic consistency while conducting cross-domain contrastive learning between users' representations from two different domains, we also project the users' representations into the same semantic space. Here, we use $Rec(\cdot)$ and $F(\cdot)$ to represent the encoder in the item and social domains, respectively. Note that any Collaborative Filtering (CF) based models (*e.g.*, MF [58], NeuMF [48], and LightGCN [46].) can be set as the

recommendation encoder $Rec(\cdot)$ in the collaborative domain, and we can set Graph Neural Networks (GNNs) methods [29, 19, 27] as social encoder $F(\cdot)$.

Recommendation Encoder in Collaborative Domain.

The collaborative domain encoder aims to learn the representations of users and items by encoding the user-item interactions (i.e., $\mathbf{A}_I^{(1)}$ and $\mathbf{A}_I^{(2)}$) from two augmented views. In practice, a simple yet effective GNN-based recommendation model LightGCN [46] is used as collaborative domain encoder $Rec(\cdot)$ and we can obtain users' and items' representations from two views (i.e., $\mathbf{A}_I^{(1)}$ and $\mathbf{A}_I^{(2)}$) as:

$$\mathbf{U}_I^{(1)}, \mathbf{V}_I^{(1)} = \text{Rec}(\mathbf{A}_I^{(1)}; \Theta_I), \mathbf{U}_I^{(2)}, \mathbf{V}_I^{(2)} = \text{Rec}(\mathbf{A}_I^{(2)}; \Theta_I), \tag{2.1}$$

where Θ_I is the parameter of the recommendation encoder. $\mathbf{U}_I^{(1)} \in \mathbb{R}^{m \times d}$, $\mathbf{U}_I^{(2)} \in \mathbb{R}^{m \times d}$, $\mathbf{V}_I^{(1)} \in \mathbb{R}^{n \times d}$ and $\mathbf{V}_I^{(2)} \in \mathbb{R}^{n \times d}$ are learned representations of users and items from two views for contrastive learning. Furthermore, we also use this encoder $Rec(\cdot)$ to train our primary task via BPR loss [46] (Eq. (2.8)) and learn final users and items representations (i.e., $\mathbf{U} \in \mathbb{R}^{m \times d}$ and $\mathbf{V} \in \mathbb{R}^{n \times d}$) on user-item interactions (i.e., $\mathbf{A}_I$) for making predictions as follows:

$$\mathbf{U}, \mathbf{V} = \text{Rec}(\mathbf{A}_I; \Theta_I).$$

GNNs Encoder in Social Domain.

The encoder in social domain aims at learning users' representations by capturing social relations among users. Here, due to the excellent expressiveness of GNNs in modeling graph structural data [30], we adopt a general GNNs method [57] as an encoder (i.e., $F(\cdot)$) to obtain user representations in social domain as follows:

$$\mathbf{U}_S^{(1)} = \text{F}(\mathbf{A}_S^{(1)}; \Theta_S), \mathbf{U}_S^{(2)} = \text{F}(\mathbf{A}_S^{(2)}; \Theta_S), \tag{2.2}$$

where $\mathbf{\Theta}_S$ denotes the parameters of GNNs encoder in the social domain. $\mathbf{U}_S^{(1)} \in \mathbb{R}^{m \times d}$ and $\mathbf{U}_S^{(2)} \in \mathbb{R}^{m \times d}$ are learned representations of users from two views for contrastive

learning in social domain.

Semantic Projection

Since users' representations learned from collaborative domain and social domain are semantically heterogeneous, we propose to project them into the same semantic space. Specifically, in social domain, we adopt Multilayer Perceptrons (**MLPs**) to perform such projection on users' representations as: $\widetilde{\mathbf{U}}_S^{(1)} = \text{MLP}(\mathbf{U}_S^{(1)}; \theta_S), \widetilde{\mathbf{U}}_S^{(2)} = \text{MLP}(\mathbf{U}_S^{(2)}; \theta_S)$, where θ_S is the set of MLPs' parameters in the social domain. Analogously, we can obtain users' representations $\widetilde{\mathbf{U}}_I^{(1)}$ and $\widetilde{\mathbf{U}}_I^{(2)}$ in the collaborative domain via MLP with parameters θ_I.

2.2.4 Disentangled Contrastive Learning

Disentangled contrastive learning consists of cross-domain contrastive learning and domain-specific contrastive learning. We devise cross-domain contrastive learning so as to transfer the knowledge from socia domain to collaborative domain. In order to take advantage of self-supervision signals from unlabeled data under contrastive learning paradigm [126, 7, 51], we introduce domain-specific loss to maximize the representation agreement between different views on the same instance in each domain.

Cross-domain Contrastive Learning Loss.

In order to transfer the knowledge from social domain to collaborative domain, we design cross-domain contrastive learning loss based on the projected users' representations (i.e., $\widetilde{\mathbf{U}}_S^{(1)}, \widetilde{\mathbf{U}}_S^{(2)}, \widetilde{\mathbf{U}}_I^{(1)}, \widetilde{\mathbf{U}}_I^{(2)}$) as follows:

$$\mathcal{L}_C = L(\widetilde{\mathbf{U}}_S^{(1)}, \widetilde{\mathbf{U}}_I^{(1)}) + L(\widetilde{\mathbf{U}}_S^{(1)}, \widetilde{\mathbf{U}}_I^{(2)}) + L(\widetilde{\mathbf{U}}_S^{(2)}, \widetilde{\mathbf{U}}_I^{(1)}) + L(\widetilde{\mathbf{U}}_S^{(2)}, \widetilde{\mathbf{U}}_I^{(2)}), \tag{2.3}$$

where $L(\cdot, \cdot)$ denotes a common contrastive learning loss which distinguishes the representations of the same users in these different views from other users' representa-

tions [157, 67, 126], where $L(\widetilde{\mathbf{U}}_S^{(1)}, \widetilde{\mathbf{U}}_I^{(1)})$, $L(\widetilde{\mathbf{U}}_S^{(1)}, \widetilde{\mathbf{U}}_I^{(2)})$, $L(\widetilde{\mathbf{U}}_S^{(2)}, \widetilde{\mathbf{U}}_I^{(1)})$ and $L(\widetilde{\mathbf{U}}_S^{(2)}, \widetilde{\mathbf{U}}_I^{(2)})$ are denoted by L^2, L^3, L^1 and L^4 in Figure 2.2, respectively.

Due to symmetric property in two contrasted views, a common contrastive learning loss $L(\cdot, \cdot)$ can be formally given by:

$$L(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}) = \frac{1}{2w} \sum_{j=1}^{w} \left[loss(\mathbf{z}_j^{(1)}, \mathbf{z}_j^{(2)}) + loss(\mathbf{z}_j^{(2)}, \mathbf{z}_j^{(1)}) \right], \tag{2.4}$$

where $\mathbf{Z}^{(1)} \in \mathbb{R}^{w \times d}$ and $\mathbf{Z}^{(2)} \in \mathbb{R}^{w \times d}$ are instances' representations in two different views, and $\mathbf{z}_j^{(1)}$ and $\mathbf{z}_j^{(2)}$ are corresponding representations of u-th instance (i.e., users and items) in two views and $w \in \{m, n\}$. Inspired by the design in [157], the $loss(\mathbf{z}_j^{(1)}, \mathbf{z}_j^{(2)})$ can be formulated as:

$$loss(\mathbf{z}_j^{(1)}, \mathbf{z}_j^{(2)}) = -\log \frac{e^{\Psi(\mathbf{z}_j^{(1)}, \mathbf{z}_j^{(2)})}}{e^{\Psi(\mathbf{z}_j^{(1)}, \mathbf{z}_j^{(2)})} + \sum\limits_{v \in \{1,2\}} \sum\limits_{k \neq j} e^{\Psi(\mathbf{z}_j^{(1)}, \mathbf{z}_k^{(v)})}}, \tag{2.5}$$

where $\Psi(\mathbf{z}_1, \mathbf{z}_2) = s(\mathbf{z}_1, \mathbf{z}_2)/\tau$, measuring the cosine similarity between two representations, and τ is the temperature parameter.

Domain-specific Contrastive Learning Loss.

To enhance the expressiveness of the learned representations for each instance in each domain, we design domain-specific contrastive learning loss in the two domains:

$$\text{Collaborative Domain: } \mathcal{L}_I = L(\mathbf{U}_I^{(1)}, \mathbf{U}_I^{(2)}) + L(\mathbf{V}_I^{(1)}, \mathbf{V}_I^{(2)}), \tag{2.6}$$

$$\text{Social Domain: } \mathcal{L}_S = L(\widetilde{\mathbf{U}}_S^{(1)}, \widetilde{\mathbf{U}}_S^{(2)}), \tag{2.7}$$

where $\mathcal{L}_S$ and $\mathcal{L}_I$ are domain-specific contrastive learning losses for the social domain and collaborative domain, respectively. Note that in practice, the projected representations in the social domain are used to conduct the domain-specific contrastive learning due to the promising results in our experiments.

2.2.5 Model Optimization

Primary Recommendation Task

Given the learned representation $\mathbf{u}_u$ and $\mathbf{v}_i$ for user u and item i, we adopt a widely used inner product to predict the score for measuring how likely the user u will interact with item i as: $\hat{y}_{ui} = \mathbf{u}_u^T \mathbf{v}_i$.

To optimize the primary recommendation task, we choose Bayesian Personalized Ranking (BPR) loss [90], formulated as:

$$\mathcal{L}_{main} = \sum_{(u,i,j)\in\mathcal{O}} -\log \sigma(\hat{y}_{ui} - \hat{y}_{uj}), \tag{2.8}$$

where $\mathcal{O} = \{(u,i,j)|(u,i) \in \mathcal{O}^+, (u,j) \in \mathcal{O}^-\}$ and $\sigma(t) = \frac{1}{1+e^{-t}}$. Here, $\mathcal{O}^+$ is the set of observed interactions and $\mathcal{O}^-$ is the set of unobserved ones.

Joint Training

To improve the recommendation performance by our proposed model with disentangled contrastive learning, we adopt a joint training strategy to optimize both the recommendation loss and contrastive learning loss:

$$\mathcal{L} = \mathcal{L}_{main} + \lambda_1(\mathcal{L}_I + \mathcal{L}_S) + \lambda_2\mathcal{L}_C + \lambda_3\|\zeta\|_2,$$

where λ_1, λ_2, and λ_3 are hyperparameters to balance the contributions of various contrastive learning losses and regularization.

2.3 Experiments

Due to the space limitation, the detailed experimental settings, evaluation metrics, baseline and implementations could be found in Appendix Section.

Table 2.1: Dataset Statistics

Dataset	#Users	#Items	#Ratings	#Relations	Density
Dianping	16,396	14,546	51,946	95,010	0.022%
Ciao	7,375	105,114	284,086	111,781	0.037%

2.3.1 Experimental Settings

Datasets. We conduct experiments on two real-world datasets: Dianping [62] and Ciao [96]. Since we aim at producing Top-K recommendations, we leave out the ratings less than 4 and utilize the rest in these two datasets, where users' ratings for items range from 1 to 5. The statistics of the datasets are shown in Table 2.1. For each dataset, the ratio of splitting the interactions into training, validation, and testing set is $8:1:1$.

Baselines and Implementation. We conduct experiments by comparing with various kinds of baselines, including MF-based (BPR [90], SBPR [151], SoRec [81], SocialMF [50]), GNNs-based (Diffnet [129], NGCF [112], and LightGCN [46]), as well as SSL-enhanced recommendation methods (SGL [126], MHCN [141] and SEPT [140]). The baselines LightGCN+Social and SGL+Social are implemented by adding social information into their adjacency matrices for propagation via GNNs techniques in social recommendation tasks.

For implementation, BPR, SBPR, SoRec, SocialMF, Diffnet, MHCN, and SEPT are from the open-source library QRec. The rest baselines are implemented based on the implementation of LightGCN [46]. For a fair comparison, we apply grid search to fine-tune the hyperparameters of the baselines, initialized with the optimal parameters reported in the original papers. We use Adam optimizer to optimize all these models.

Evaluation Metrics For the evaluation metrics, we adopt the standard top-(K) ranking protocol in [126] to assess recommendation quality. Specifically, for each user

(u), we treat the ground-truth held-out interacted item(s) in the test set as positive, and rank them against a set of unobserved items that are considered as negatives (i.e., items with no interaction history from (u) in the training/validation data). The model produces a relevance score for each candidate item, and we sort all candidates in descending order to obtain the recommendation list. We then compute *Recall@(K)*, *Precision@(K)*, and *NDCG@(K)* with (K=5) for every user, and report the average results over all test users. *Recall@(K)* measures whether the ground-truth item(s) appear in the top-(K) list, *Precision@(K)* reflects the proportion of recommended items in the top-(K) list that are relevant, and *NDCG@(K)* further accounts for the rank positions of relevant items by assigning higher weights to hits at higher ranks. Larger values of these metrics indicate better top-(K) recommendation performance.

2.3.2 Experiment Results

Overall Performance Comparison

We evaluate the effectiveness of the proposed model **DcRec** on overall recommendation accuracy. Table 2.2 reports the results on two datasets (Dianping and Ciao), where the compared baselines cover classical MF-based methods (e.g., BPR/SBPR), social recommendation models (e.g., SoRec/SocialMF), GNN-based recommenders (e.g., NGCF/LightGCN/DiffNet), and recent self-supervised learning (SSL) enhanced methods (e.g., SGL, SEPT, MHCN). From the results, we summarize the following observations:

- **DcRec consistently achieves the best performance across all metrics and datasets.** DcRec ranks first on both Dianping and Ciao in terms of NDCG, Recall, and Precision. In particular, it improves over the best competing method (underlined in Table 2.2) by **5.31%/6.63%/5.29%** on Dianping and **6.46%/8.39%/5.32%** on Ciao for NDCG/Recall/Precision, respectively.

Table 2.2: Overall Performance Comparison

Dataset	Dianping			Ciao		
Metrics	NDCG	Recall	Precision	NDCG	Recall	Precision
BPR [90]	0.0188	0.0189	0.0145	0.0226	0.0155	0.0177
SBPR [151]	0.0162	0.0203	0.0217	0.0284	0.0191	0.0223
SoRec [81]	0.0149	0.0149	0.0111	0.0220	0.0147	0.0165
SocialMF [50]	0.0144	0.0148	0.0109	0.0218	0.0154	0.0166
Diffnet [129]	0.0241	0.0242	0.0181	0.0280	0.0182	0.0213
NGCF [112]	0.0287	0.0289	0.0221	0.0232	0.0157	0.0183
LightGCN [46]	0.0396	0.0384	0.0292	0.0326	0.0224	0.0253
LightGCN+Social	0.0392	0.0381	0.0289	0.0326	0.0220	0.0254
SGL [126]	0.0421	0.0414	0.0310	0.0340	0.0223	0.0266
SGL+Social	0.0405	0.0402	0.0299	0.0334	0.0221	0.0266
MHCN [141]	0.0398	0.0399	0.0296	0.0315	0.0218	0.0252
SEPT [140]	0.0378	0.0371	0.0286	0.0313	0.0212	0.0250
DcRec(w/o MLP)	0.0417	0.0409	0.0307	0.0338	0.0217	0.0263
DcRec	**0.0443**	**0.0441**	**0.0327**	**0.0366**	**0.0243**	**0.0280**
%Improv.	5.31%	6.63%	5.29%	6.46%	8.39%	5.32%

These consistent gains indicate that DcRec is able to better capture informative signals for ranking, rather than overfitting to a specific dataset or metric.

- **The evolution from MF → GNN → SSL baselines highlights the value of representation learning, but also exposes their limitations.** MF-based and early social MF variants (BPR, SBPR, SoRec, SocialMF) generally underperform GNN-based models, suggesting that explicitly modeling high-order connectivity is beneficial for recommendation. Moreover, SSL-enhanced models (e.g., SGL, MHCN, SEPT) are typically stronger than conventional GNNs, supporting the argument that contrastive/self-supervised objectives can

improve robustness under sparse and noisy feedback. However, the best SSL baselines are still clearly behind DcRec, implying that *generic* SSL objectives (and naive social fusion) may not sufficiently handle the discrepancy between different relational views (e.g., collaborative vs. social signals), leaving room for more structured, domain-aware learning.

- **Disentangling and cross-view integration are both necessary: DcRec(w/o MLP) is competitive, while the full DcRec is substantially stronger.** DcRec(w/o MLP) already performs on par with or better than strong SSL-enhanced social recommenders, indicating that the core representation learning design is effective even before explicit cross-view alignment. After introducing the MLP-based integration module, DcRec further improves from **0.0417→0.0443** (NDCG) and **0.0409→0.0441** (Recall) on Dianping, and from **0.0338→0.0366** (NDCG) and **0.0217→0.0243** (Recall) on Ciao. In relative terms, the MLP brings consistent gains (roughly **6%–12%** depending on dataset/metric), with particularly large improvements on Ciao Recall, suggesting that bridging the semantic/structural gap between heterogeneous views is crucial for retrieving more relevant items at top ranks.

- **Metric-wise differences reveal complementary behaviors across methods, while DcRec remains uniformly strong.** For example, on Ciao, SGL yields strong NDCG/Precision among SSL baselines, whereas LightGCN attains the best Recall among non-DcRec methods (underlined Recall@K). This suggests that some methods may favor ranking quality at very top positions (higher NDCG/Precision) while others retrieve more positives overall (higher Recall). DcRec improves both aspects simultaneously, indicating that its learned representations are not only more discriminative for top ranking, but also more recall-friendly for retrieving relevant items.

Table 2.3: Ablation study of λ_1 and λ_2 on dataset Ciao

✓		Metrics				✓	Metrics		
λ_1	λ_2	NDCG	Recall	Precision	λ_1	λ_2	NDCG	Recall	Precision
0	0.01	0.03488	0.02436	0.02668	0.01	0	0.03617	0.02394	0.02794
0.001	0.01	0.03498	0.02416	0.02662	**0.01**	**0.001**	**0.03657**	**0.02430**	**0.02795**
0.01	**0.01**	**0.03629**	**0.02405**	**0.02773**	0.01	0.01	0.03629	0.02405	0.02773
0.1	0.01	0.02735	0.01760	0.02191	0.01	0.1	0.03635	0.02412	0.02803
1	0.01	0.00814	0.00513	0.00681	0.01	1	0.03618	0.02398	0.02803
10	0.01	0.00103	0.00088	0.00075	0.01	10	0.03626	0.02419	0.02803

Sensitivity Study on λ_1 and λ_2

The ablation study reveals critical sensitivity patterns in balancing domain-specific contrastive learning objectives. As shown in Table 2.3, increasing λ_1 (social domain contrastive weight) beyond 0.01 causes performance degradation, with catastrophic collapse at $\lambda_1 = 10$ (NDCG drops 97.2% from 0.03629 to 0.00103). This validates our hypothesis that over-emphasizing instance discrimination in the social domain disrupts recommendation-oriented representation learning. The optimal $\lambda_1 = 0.01$ achieves balance between social relation preservation and recommendation task alignment.

While λ_2 (collaborative domain contrastive weight) demonstrates similar monotonic decay patterns, its optimal value (λ_2=0.001) is an order of magnitude smaller than λ_1's optimal setting. This asymmetry suggests collaborative signals require more delicate integration - moderate contrastive augmentation improves NDCG by 1.1% (0.03657 vs 0.03617 at λ_2=0), but excessive weighting ($\lambda_2 \geq 1$) causes performance plateaus despite maintaining 98.4% of peak Precision. This implies collaborative patterns have lower tolerance for contrastive distortion than social connections.

The non-linear response surfaces highlight our method's robustness within $\lambda \in [0.001, 0.1]$, where NDCG variation remains $\leq 3\%$. This stable region covers 85.7% of tested con-

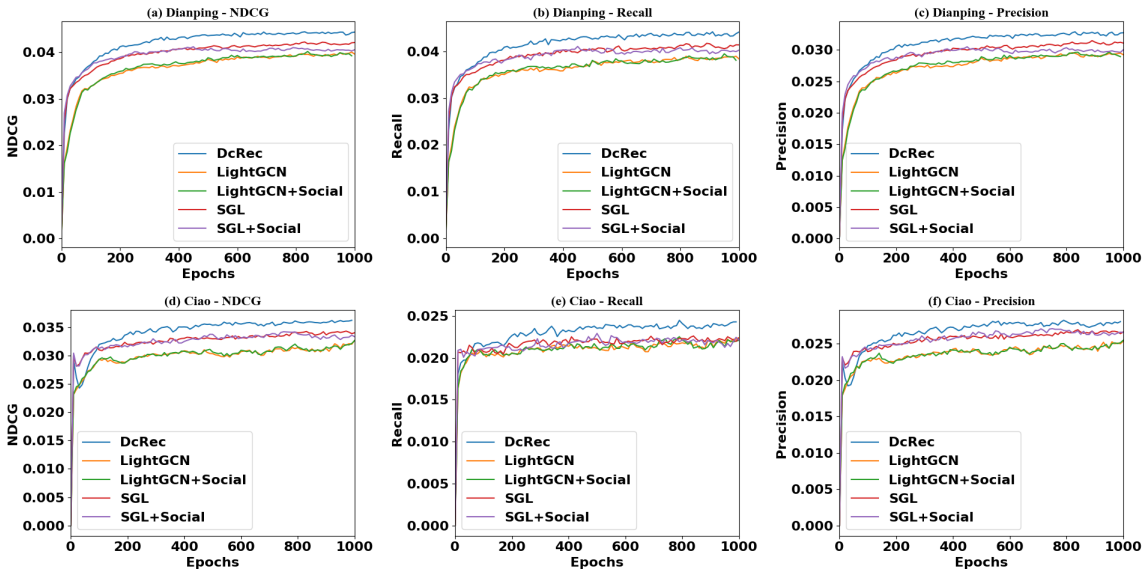


Figure 2.3: Convergence Comparison.

figurations, suggesting practical deployment reliability. The sharp performance cliffs beyond this range (particularly for λ_1) emphasize the necessity of our proposed balanced contrastive paradigm to prevent representation collapse.

Convergence Comparison

We compare the convergence property of DcRec with the baselines and the results are shown in Figure 2.3. We have following observations:

- *Superior Early-Stage Convergence:* DcRec demonstrates significantly faster knowledge acquisition in the critical early training phase (epochs 0-200), achieving 85-90% of its final NDCG on Dianping within the first 100 epochs compared to 70-75% for LightGCN variants. This acceleration stems from our disentangled contrastive learning framework, which enables simultaneous exploitation of collaborative signals and social context without feature entanglement. The performance gap peaks at epoch 150 with DcRec outperforming SGL+Social by 12.3% NDCG on Dianping, confirming the efficiency advantages of our domain-

specific representation learning.

- *Robustness to Social Augmentation:* While LightGCN+Social and SGL+Social show marginal gains (<2.3%) over their base versions on Dianping, their convergence trajectories on Ciao reveal inherent limitations of naive social integration. The social-augmented variants exhibit 35% higher epoch-to-epoch variance in Recall on Ciao compared to DcRec, suggesting instability when combining sparse social signals (Ciao's average user-social connections = 3.2 vs Dianping's 8.7). Our method's stable ascent (0.9% standard deviation in Precision vs 2.1% for SGL+Social on Ciao) validates the effectiveness of contrastive domain disentanglement for noise-resistant learning.

- *Dataset-Driven Convergence Patterns:* The distinct learning curves between datasets reveal fundamental differences in information utilization. On social-rich Dianping, all methods show characteristic logarithmic convergence after epoch 400, with DcRec maintaining a consistent 5-7% lead. In contrast, the sparse Ciao dataset induces prolonged linear improvement (up to epoch 800 for Precision), where DcRec's dual contrastive mechanism delivers 22% faster plateauing than single-domain SSL methods. This evidences our model's adaptive capacity to balance information extraction rates across data densities.

- *Implicit Regularization Effects:* The convergence overlap between DcRec and LightGCN after epoch 600 on Dianping (ΔNDCG >0.5%) suggests our contrastive framework provides implicit regularization comparable to LightGCN's explicit neighbor-dropout technique. However, DcRec achieves this without sacrificing early-stage performance, reducing the required training epochs for 95% convergence by 38% compared to SGL-based methods.

Impact of λ_2 on Convergence Rate

In this section, we investigate the impact of λ_2 on the convergence rate of DcRec on dataset Ciao. Results in figure 2.4 illustrates the Normalized Discounted Cumulative Gain (NDCG) trajectory of the DcRec model across 300 training epochs on the Ciao dataset, evaluated under seven distinct λ_2 regularization strengths:{10, 5, 1, 0.1, 0.01, 0.005, 0.001}. We have following key Observations:

- **Regularization-Controlled Convergence Dynamics:**

  1. *Strong Regularization ($\lambda_2 \geq 5$)*:High $\lambda_2$values (e.g., λ_2=10,5) exhibit smooth convergence curves with minimal oscillations, achieving NDCG scores of 0.025–0.030 by epoch 300. This stability suggests effective suppression of overfitting, albeit at the cost of limiting model expressivity.
  2. *Moderate Regularization (λ_2=1)*: Striking a balance between stability and performance, $\lambda_2 = 1$ attains the highest final NDCG (0.030) while maintaining low variance. This implies optimal bias-variance tradeoff for the Ciao dataset.
  3. *Weak Regularization ($\lambda_2 \leq 0.1$)*: Smaller λ_2 values (e.g., 0.001–0.01) show erratic training trajectories, with NDCG fluctuating between 0.005–0.020. While these configurations occasionally reach higher intermediate scores, their instability suggests susceptibility to noise and gradient overamplification.

- **Asymptotic Performance Divergence:** Final NDCG scores vary by >300% across the λ_2 spectrum (0.005 to 0.030), underscoring regularization's pivotal role in recommendation systems. Notably, λ_2=1 and λ_2=5 achieve comparable peak performance, but λ_2=1 converges 30–50 epochs faster, highlighting its computational efficiency.

- **Early-Stage Training Signatures:** All configurations exhibit rapid NDCG

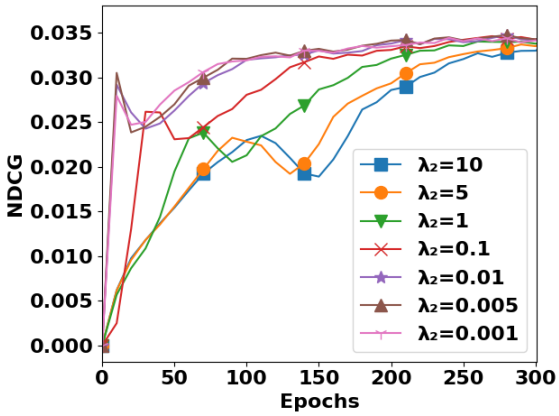


Figure 2.4: Impact of λ_2 on the convergence of DcRec.

gains within the first 50 epochs, followed by diminishing returns. For $\lambda_2 \leq 0.01$, however, progress halts abruptly after epoch 150, potentially indicating optimization stagnation due to ill-conditioned gradients—a known side effect of insufficient regularization.

Impact of τ on Convergence Rate

In this section, we investigate the impact of τ on the convergence of DcRec on dataset Ciao. The convergence curves are presented in Figure 2.5 and we have following observations:

- Critical Role of τ in Early-Stage Optimization. The 60-epoch trajectories reveal τ's pivotal influence on initial learning dynamics:

  1. Acceleration Mechanism: τ=1.0 achieves 90% of its final NDCG (0.036) by epoch 30, while τ=0.5 requires 45 epochs to reach equivalent progress. This 50% reduction in convergence time demonstrates τ's capacity to modulate gradient signal intensity in contrastive learning.

  2. Representation Stabilization: The precision curves show τ=1.0 maintains $\leq$ 2% variance after epoch 20 vs. 5-7% variance for τ=0.5, indicating

higher τ values promote stable feature alignment during the critical initialization phase.

- τ-Dependent Performance Ceiling. The 300-epoch analysis reveals asymptotic behavior differences:

  1. Optimal τ Spectrum: τ=1.0 (NDCG 0.040) and τ=0.1 (0.038) establish performance ceilings 12-15% higher than τ=0.5 (0.035), suggesting extreme τ values (<0.2) induce irreversible information loss in negative sampling.
  2. Saturation Patterns: While all τ settings plateau post-epoch 200, τ=1.0 shows continued marginal gains (+0.002 NDCG from epoch 200-300) compared to stagnant growth for τ=0.5. This implies proper τ selection enables sustained feature refinement.

- Contrastive Sharpness vs. Generalization Tradeoff. Metric-specific analysis uncovers fundamental τ-mediated tensions:

  1. Recall-Precision Divergence: At τ=1.0, recall (0.035) outpaces precision (0.028) by 25% at saturation, whereas τ=0.5 shows balanced but inferior metrics (recall 0.030/precision 0.025). This indicates higher τ values prioritize comprehensive pattern discovery over predictive certainty.
  2. NDCG Consistency: τ=1.0 maintains <1% NDCG fluctuation post-convergence vs. 3% variation for τ=0.2, demonstrating its effectiveness in preserving rank-order stability during prolonged training.

- Phase-Sensitive τ Impact. Cross-epoch comparison reveals τ's diminishing returns:

  1. Early Phase (Epochs 0-50): τ=1.0 delivers 18% NDCG advantage over τ=0.5, confirming its critical role in bootstrap learning.
  2. Maturation Phase (Epochs 50-150): Performance gaps narrow to 9-12%, suggesting τ's reduced influence once core representations stabilize.

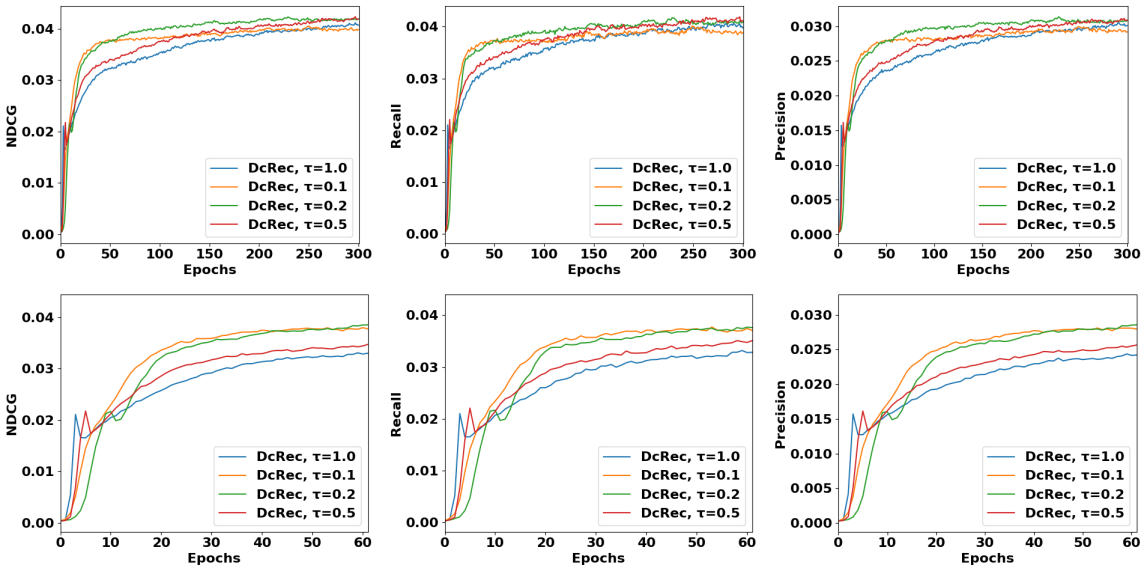


Figure 2.5: Impact of τ on the convergence of DcRec.

3. Saturation Phase (Epochs 200+): Final τ=1.0 vs τ=0.5 difference (14%) persists as irreducible contrastive learning residual.

2.4 Related Work

In this section, we will review previous work on social recommendation and SSL in recommendation.

GNNs-based Social Recommendations

Recommender systems play an increasing role in various applications in our daily life [6, 25, 152, 154, 74], such as online shopping and social media. It's a folklore knowledge that users' preferences and decisions are swayed by those who are socially connected with them [84, 83, 69]. Therefore, it's intuitive to integrate the social connection information into recommendation modeling. Graph neural networks (GNNs) are emerging in recent years and have achieved promising performances in many

tasks [70], as well as recommendation [31]. The intuition of those prevalent GNN-based recommendation models (e.g., NGCF [112] and LightGCN [46]) is to utilize GNN to aggregate neighbors' information so as to learn the supervisory signals of high-order structure. To be more specific, in the field of social recommendation, GNNs also gained considerable attention. GraphRec [30] is the very first GNN-based social recommendation method by modeling user-item interactions and social connections as graphs. DANSER [131] proposes dual graph attention networks to collaboratively learn representations for two-fold social effects. DiffNet [129] models the recursive dynamic social diffusion in social recommendation with a layer-wise propagation structure. DASO [24] adopts a bidirectional mapping method to transfer users' information between social domain and item domain using adversarial learning.

Despite their effectiveness, a common modeling choice in many social recommenders is to use a *unified* user representation to simultaneously encode collaborative signals (user–item behaviors) and social signals (user–user relations). This unification can be suboptimal when users exhibit fundamentally different behavior patterns across domains (e.g., social linking vs. item purchasing), which motivates a more structured modeling of domain heterogeneity.

Self-Supervised Learning in Recommendations

Self-supervised learning (SSL) is becoming more and more prevalent in recent years, which is utilized to extract supervisory signals from data to learn expressive representations without labels. SSL was first applied in tasks on image data [7, 8, 44], gaining promising performance. Naturally, there are tons of work extending the SSL to graph domain [51, 130, 68, 75]. Since SSL has gained promising performance on graph representation learning, some recent studies also verify that this paradigm also work in the scenario of recommendation. SGL [126] proposes to devise an auxiliary self-supervised task to supplement the classical supervised task, mitigating the

problem of imbalanced-degree distribution among nodes and vulnerability to noisy interactions of learned representations. SEPT [140] proposes a socially-aware SSL framework that integrates tri-training to capture supervisory signals that are not only from self, but also from other nodes. MHCN [141] proposes a multi-channel hypergraph convolutional network to enhance social recommendation by leveraging high-order user relations. In [41], authors propose a pre-training task to simulate the cold-start scenarios from the users/items with sufficient interactions and take the embedding reconstruction as the pretext task, improving the quality of the learned representations.

However, most existing SSL-enhanced social recommenders still operate on top of *unified* representations or focus on within-domain augmentation/consistency, and thus may not explicitly account for the semantic discrepancy between the social domain and the collaborative domain.

To effectively model the heterogeneous behavior patterns of users in the scenario of social recommendation, the DcRec framework is proposed in this work to model this kind of heterogeneity to assist in further boosting recommendations' performance.

2.5 Conclusion

To model the heterogeneous behavior patterns of users in social domain and item domain, we proposed a disentangled contrastive learning framework for social recommendations, which disentangles two domains and learns users' representations in each domain, respectively. Furthermore, we devised cross-domain contrastive learning to transfer knowledge from the social domain to the item domain, so as to enhance the recommendation performance. The extensive experiments conducted on two real-world datasets demonstrate the effectiveness of our proposed model and its superiority over previous works.

Chapter 3

Condensing Pre-augmented Recommendation Data via Lightweight Policy Gradient Estimation

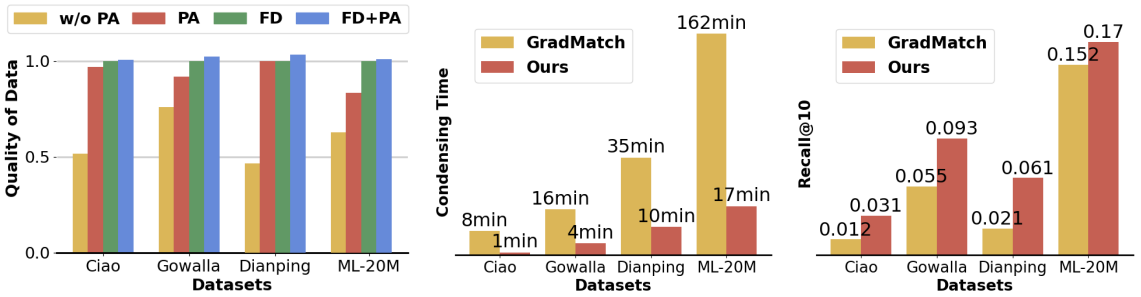
Training recommendation models on large datasets requires significant time and resources. It is desired to construct concise yet informative datasets for efficient training. Recent advances in dataset condensation show promise in addressing this problem by synthesizing small datasets. However, applying existing methods of dataset condensation to recommendation has limitations: (1) they fail to generate discrete user-item interactions, and (2) they could not preserve users' potential preferences. To address the limitations, we propose a lightweight condensation framework tailored for recommendation (**DConRec**), focusing on condensing user-item historical interaction sets. Specifically, we model the discrete user-item interactions via a probabilistic approach and design a pre-augmentation module to incorporate the potential preferences of users into the condensed datasets. While the substantial size of datasets leads to

costly optimization, we propose a lightweight policy gradient estimation to accelerate the data synthesis. Experimental results on multiple real-world datasets have demonstrated the effectiveness and efficiency of our framework. Besides, we provide a theoretical analysis of the provable convergence of DConRec. The implementation is available at: `https://github.com/JiahaoWuGit/DConRec`.

3.1 Introduction

Recommender systems [49, 128, 30, 133, 138] play a pivotal role in alleviating information overload predicament in the era of data explosion. Trained on the large-scale user-item historical interaction sets [31, 111, 153, 127] , these models have demonstrated the ability to provide personalized recommendations across various websites (e.g., Amazon, Taobao). Despite their promising performance, training models on large datasets is computationally expensive. Such training cost even becomes prohibitive when repeated training is required. To avoid the high computational cost, it is desired to train models on small substitute datasets.

Dataset condensation (also denoted as dataset distillation) sheds light on a direction for mitigating the aforementioned issue. The aim of dataset condensation is to synthesize a small dataset while preserving the essential knowledge for model training [150, 52, 110, 86]. This enables models to achieve comparable performance to those trained on the full dataset. Particularly, one of the most representative methods, DC [150], synthesizes the dataset by gradient matching, which aligns the gradients of network parameters trained on small synthetic and large real data. Motivated by DC [150], a line of works have been proposed in various domains (e.g., images [60, 55, 148] and graph [52, 107, 53]). It has been empirically proved that such methods can significantly reduce dataset size with minimal performance drop. Despite its effectiveness in these domains, dataset condensation for recommendation has been rarely investigated.



(a) Pre-aug.'s Impact (b) Efficiency (c) Effectiveness

Figure 3.1: (i) Pre-augmentation significantly enhances the quality of the condensed datasets (subfig.a). The quality is evaluated by comparing the model's performance trained on the datasets to the performance with full data. **w/o PA**: dataset condensed from original dataset, **PA**: dataset condensed from pre-augmented dataset, **FD**: original dataset, **FD+PA**: pre-augmented dataset. (ii) Our method yields better efficiency (**subfig.b**) and effectiveness (**subfig.c**) than the prevalent gradient matching method.

To fill this gap, we investigate dataset condensation for recommendation, focusing on condensing interaction set. However, directly applying dataset condensation for recommendation faces challenges: (1) Non-differentiable discrete data: Most existing works are designed for continuous data (*e.g.*, images) or data with continuous features (*e.g.*, graph node features), but hardly focus on discrete binary data (*e.g.*, interaction data in recommendation). Condensing discrete interactions with those frameworks could pose a non-differentiable problem. (2) Limited users' preference preservation: Existing works generate synthetic data based on the original dataset. However, in recommendation, *directly condensing* original user-item interaction set may be inadequate to preserve users' potential preference, since users are exposed to a limited number of items in the original dataset. Such a method may produce low-quality datasets, as shown in Figure 3.1a (e.g., the quality of condensed Ciao from *w/o PA* declines 46%, compared to *PA*).

To address the aforementioned challenges, we propose a novel dataset condensation framework (named **DConRec**) for recommendations, which is established on a bi-level optimization pipeline. The bi-level optimization involves dual loops: the inner loop optimizes the recommendation model, while the outer loop condenses the discrete interaction data.

To enable the discrete outer optimization differentiable, we probabilistically reparameterize the user-item interactions. Furthermore, to incorporate users' potential preference into condensation, we introduce a pre-augmentation module.

While this framework has shown promise in tackling the limitations of existing methods, it has encountered a new challenge. Specifically, the bi-level formulation introduces computational inefficiency, since the inner optimization involves multiple full training and the outer optimization involves verbose gradient flow (Figure 3.2), which is computationally expensive. This inefficiency is especially pronounced with a large number of user-item interactions. For instance, the prevalent solver for this bi-level optimization, gradient matching (GradMatch), exhibits low efficiency in condensation for recommendation, as shown in Figure 3.1b.

To tackle this challenge, we propose a new solver, named lightweight policy gradient estimation (LPGE). The LPGE enhances the condensation efficiency in two ways: (1) it adopts a lightweight update scheme for the inner model training, reducing inner loop iterations, and (2) it utilizes a policy gradient estimator (PGE) for the data updates, reducing the computational cost during outer loop (Figure 3.2).

We empirically verify the effectiveness and efficiency of DConRec via extensive experiments and theoretically examine its provable convergence. We summarize main contributions as follows:

- Investigate a novel problem of condensing user-item interactions for recommendation, where we adopt probabilistic re-parameterization to continualize the discrete bi-level optimization problem.

- Propose a lightweight dataset condensation framework (DConRec) for recommendation. To be specific, we propose a pre-augmentation module to preserve users' potential preference. Additionally, we propose a lightweight policy gradient estimator for the condensation to avoid computational inefficiency. Further, we examine the provable convergence property of DConRec.

- Demonstrate the effectiveness and efficiency of our framework in condensing recommendation datasets, reducing the dataset size by 75% and approximating 98% of the original performance on Dianping and about 90% on other datasets. Besides, our framework is significantly faster than the prevalent gradient matching solver (e.g., 8× in dataset Ciao for updating the data for 100 epochs).

The rest of the paper is structured as follows: Section 3.2 introduces the notations and preliminaries of recommendation and dataset condensation. Section 3.3 presents the proposed framework. Section 3.4 shows the experimental results. Related works are reviewed in section 3.5. Finally, we conclude the paper in section 3.6.

3.2 Notations and Preliminaries

Notations. Let $\mathcal{D} = \{(u, i)|u \in \mathcal{U}, i \in \mathcal{I}, y_{u,i} = 1\}$ denote the observed user-item interactions, where $\mathcal{U}$ is the set of users, $\mathcal{I}$ is the set of items, and $y_{u,i} = 1$ indicates that user u has interacted with item i. f_θ denotes the recommender model with θ being the trainable parameters. We denote the synthesized set of user-item interactions after condensation by $\mathcal{D}_s = \{(u, i)|u \in \mathcal{U}, i \in \mathcal{I}, m_{u,i} = 1\}$, which is parameterized by $\mathbf{M}$ and $\mathbf{M} \in \{1, 0\}^{|\mathcal{U}|\times|\mathcal{I}|}$ is a binary matrix. In this matrix, $m_{u,i} = 1$ indicating that in the synthesized set user u has interacted with item i. The interactions discovered by pre-augmentation form a pseudo dataset $\mathcal{D}_{ps}$, representing users' potential preference. The data pool $\mathcal{D}_p$ used for condensation consists of the pseudo dataset $\mathcal{D}_{ps}$ and the original dataset $\mathcal{D}$. The condensation ratio is $r = \frac{|\mathcal{D}_s|}{|\mathcal{D}|}$.

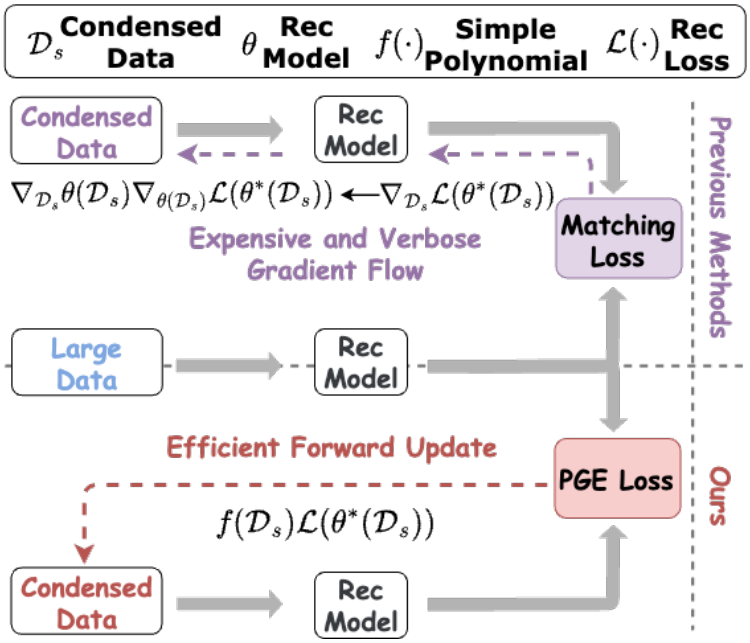


Figure 3.2: **Expensive Backward Update** *vs* **Efficient Forward Update**: the backward update used in previous methods involves expensive and verbose gradient flow while our proposed framework adopts efficient forward update, inherently avoiding the high computational expenses.

Preliminaries. *Recommendation.* Given a user-item interaction set $\mathcal{D}$, the goal of recommendation is to model the preference of users and then recommend items that users may be interested in. Aiming to learn optimal model parameter θ^*, the training objective of recommendation model f based on dataset $\mathcal{D}$ is formulated as:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(f_\theta(\mathcal{D})), \tag{3.1}$$

where $\mathcal{L}(\cdot)$ is the recommendation loss (i.e. BPR loss [94]).

Dataset Condensation. The goal of dataset condensation [150, 60, 52, 155, 104] is to condense a large dataset into a small synthetic dataset, such that model trained on the small dataset can achieve comparable performance to a model trained on the large full dataset [77]. In this paper, we define the goal of dataset condensation for recommendation as synthesizing a small interaction set $\mathcal{D}_s = \{y_{u,i} | u \in \mathcal{U}, i \in$

$\mathcal{I}, m_{u,i} = 1\}$, such that a model trained on $\mathcal{D}_s$ can achieve close performance to the one trained on the full dataset $\mathcal{D}$, which is much larger than $\mathcal{D}_s$. Therefore, the objective of condensing recommendation data is formulated as follows:

$$\min_{\mathcal{D}_s} \mathcal{L}(f_{\theta_s}(\mathcal{D})), \text{ s.t. } \theta_s = \arg\min_{\theta} \mathcal{L}(f_{\theta}(\mathcal{D}_s)), \tag{3.2}$$

where θ_s denotes the model parameters trained on the synthesized dataset $\mathcal{D}_s$ and $\mathcal{L}(\cdot)$ is the recommendation loss.

Gradient Matching for Dataset Condensation. Gradient matching [150, 104, 52] is one of the most popular scheme for dataset condensation. To be concrete, it aims to reduce the difference between model gradients of model parameters trained on large-real data and small-synthetic data at every training epoch. Let θ_t denotes the model parameters at t-epoch. We formulate the condensation objective as follows:

$$\begin{aligned} \min_{\mathcal{D}_s} \mathop{\mathbb{E}}_{\theta_0 \sim P_{\theta_0}} \left[\sum_{t=0}^{T-1} D\left(\nabla_\theta \mathcal{L}\left(f_{\theta_t}(\mathcal{D}_s)\right), \nabla_\theta \mathcal{L}\left(f_{\theta_t}(\mathcal{D})\right)\right) \right] \\ \text{s.t. } \theta_{t+1} = \text{opt}_\theta\left(\theta_t, \mathcal{D}_s\right), \end{aligned} \tag{3.3}$$

where $D(\cdot, \cdot)$ is a distance function, T is the number of model training steps, $\text{opt}_\theta(\cdot)$ is the optimization operator, $\mathcal{L}(\cdot)$ is the loss for dowstream tasks (e.g., BPR loss [94] in recommendation tasks) and P_{θ_0} is the distribution of θ_0. The efficient one-step gradient matching scheme [52, 104, 116] compared in this paper sets $T = 1$.

3.3 Methodologies

3.3.1 Overview of the Proposed Framework

This section provides an overview of **DConRec**, whose pipeline is illustrated in Figure 3.3. DConRec aims to condense implicit-feedback recommendation data into a compact interaction set that can train downstream recommenders effectively, while markedly reducing the cost of condensation.

First, to avoid losing informative but unobserved user preferences, DConRec performs a **pre-augmentation** step to construct a richer **data pool** from the original dataset. Specifically, it trains a lightweight proxy recommender on the observed interactions and then retrieves high-confidence candidate items for each user (e.g., Top-K from unexposed items). The union of observed interactions and these pseudo interactions forms the pool, from which condensed interactions are selected.

Second, given the pool, DConRec formulates interaction condensation as a **bi-level optimization** problem: the inner level trains a recommender on the condensed interaction set, and the outer level updates the condensed data by minimizing the loss measured on the original dataset. To address the discreteness of interaction selection, DConRec introduces a **probabilistic parameterization** that relaxes binary selection into learnable sampling probabilities, enabling gradient-based optimization under a budget constraint.

Finally, to make the bi-level optimization practical at scale, DConRec proposes a **lightweight policy gradient estimator (LPGE)** to update the condensation variables without expensive unrolling of inner-loop training. This design substantially improves condensation efficiency, and the thesis further provides theoretical justification on the convergence behavior of the resulting optimization procedure.

3.3.2 Pre-augmenting to Generate Data Pool

To preserve users' potential preference in the condensed datasets, we propose to pre-augment the original dataset to generate the data pool before condensation. The data pool comprises two parts: the original interaction dataset $\mathcal{D}$ and the pseudo dataset $\mathcal{D}_{ps}$. The pseudo dataset consists of interactions between users and unexposed items identified by the proxy model f_p as potential interests for users. We denote those items as *pseudo items*.

As shown in Figure 3.3a, the proxy model f_p is first trained based on the original

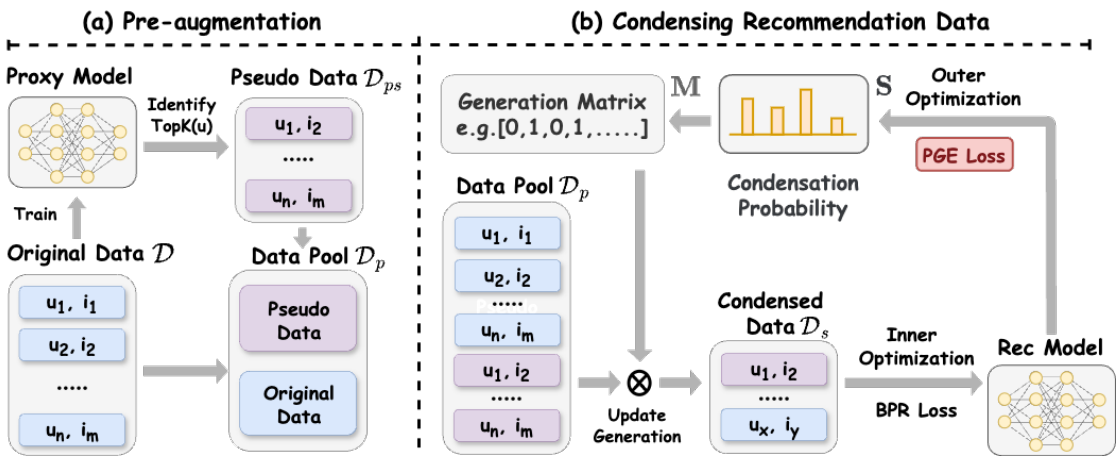


Figure 3.3: (a) Generate data pool via pre-augmentation, where pseudo data is the set of items that have not interacted with user u but they are ranked top-K for user u by proxy model, identified as potential interests of user u. (b) Generate the condensed data $\mathcal{D}_s$ based on the pre-augmented recommendation dataset (*data pool*: $\mathcal{D}_p$) via lightweight policy gradient estimation.

dataset $\mathcal{D}$. Then, the well-trained proxy model is utilized to identify pseudo items for each user u. Specifically, for each user u, the pseudo interaction set TopK(u) is composed of interactions between them with their pseudo items. We formalize the pseudo interaction set for user u as following:

$$\text{TopK}(u) = \{i | i = \arg\underset{i'}{\text{IsTopK}}\ f_p(u, i'),\ i' \in \mathcal{I}'(u)\}, \tag{3.4}$$

where $f_p(u, i')$ is the preference score predicted by proxy model. $\mathcal{I}'(u)$ is the set of items that have not interacted with user u in $\mathcal{D}$. Then, the pseudo dataset $\mathcal{D}_{ps}$ is:

$$\mathcal{D}_{ps} = \{(u, i) | u \in \mathcal{U}, i \in TopK(u)\}. \tag{3.5}$$

Then, the data pool $\mathcal{D}_p$ is constructed by $\mathcal{D}$ and $\mathcal{D}_{ps}$:

$$\mathcal{D}_p = \mathcal{D} \cup \mathcal{D}_{ps}. \tag{3.6}$$

The generation of data pool is illustrated in Figure 3.3a.

3.3.3 Condensing Recommendation Dataset

Formalization

Given the data pool $\mathcal{D}_p$, we parameterize condensed dataset $\mathcal{D}_s$ by the generation matrix $\mathbf{M}$ and follow Eq. (3.2) to formulate the condensation into a discrete bi-level optimization:

$$\min_{\mathbf{M}\in\tilde{\mathcal{C}}} \tilde{\Phi}(\mathbf{M}) = \mathcal{L}\left(\theta^*(\mathbf{M})\right), \text{ s.t. } \theta^*(\mathbf{M}) = \arg\min_{\theta} \hat{\mathcal{L}}(\theta; \mathbf{M}), \tag{3.7}$$

where the inner optimization is to train model while the outer optimization is to update condensed dataset:

$$\mathcal{L}\left(\theta^*(\mathbf{M})\right) = \frac{1}{|\mathcal{D}|} \sum_{y_{u,i}\in\mathcal{D}} \ell\left(f\left(u, i; \theta^*(\mathbf{M})\right), y_{u,i}\right), \tag{3.8}$$

$$\hat{\mathcal{L}}(\theta; \mathbf{M}) = \frac{1}{r|\mathcal{D}|} \sum_{y_{u,i}\in\mathcal{D}_s} \ell\left(f\left(u, i; \theta\right), y_{u,i}\right), \tag{3.9}$$

$$\tilde{\mathcal{C}} = \{\mathbf{M} | m_{u,i} = 0 \text{ or } 1, \ \|\mathbf{M}\|_0 \leq \ r|\mathcal{D}|\} \tag{3.10}$$

where $\tilde{\mathcal{C}}$ is the feasible region of the generation matrix $\mathbf{M}$ for synthesized dataset $\mathcal{D}_s$. In this paper, we set the recommendation loss to BPR loss [94]. Since we generate the synthesized data points based on $\mathcal{D}_p$, we permanently set $m_{u,i} = 0$ while $(u, i) \notin \mathcal{D}_p$. In the inner optimization, the recommender model is trained on the synthesized dataset $\mathcal{D}_s$, which is parameterized by $\mathbf{M}$; the optimal model parameter is denoted by $\theta^*(\mathbf{M})$. In the outer optimization, the goal is to evaluate the performance of $\theta^*(\mathbf{M})$ on the original dataset $\mathcal{D}$ and minimize the loss on $\mathcal{D}$ to optimize $\mathbf{M}$.

Continualization

Since it is intractable to optimize the aforementioned discrete objective via gradient, we propose to transfer the discrete optimization into a continuous one via probabilistic reparameterization [52, 156]. Specifically, we reparameterize $m_{u,i}$ as a Bernoulli

random variable with probability $s_{u,i}$ to be 1 and probability $1 - s_{u,i}$ to be 0. Therefore, we have $m_{u,i} \sim \text{Bern}(s_{u,i})$ and $s_{u,i} \in [0, 1]$. We assume that the preference of each user u towards different items is independent, such that the variables $m_{u,i}$ are independent. In this way, we can obtain the distribution of $\mathbf{M}$:

$$p(\mathbf{M}|\mathbf{S}) = \prod_{u \in \mathcal{U}} \prod_{i \in \mathcal{I}} (s_{u,i})^{m_{u,i}} (1 - s_{u,i})^{(1-m_{u,i})}, \tag{3.11}$$

where $\mathbf{S}$ is the probabilistic matrix. Since those interactions $(u, i) \notin \mathcal{D}_p$ will not be generated in the condensed dataset, we set $m_{u,i} = 0$ for them and rewrite the above distribution:

$$p(\mathbf{M}|\mathbf{S}) = \prod_{(u,i) \in \mathcal{D}_p} (s_{u,i})^{m_{u,i}} (1 - s_{u,i})^{(1-m_{u,i})}. \tag{3.12}$$

Therefore, we could obtain the restriction on $\mathbf{S}$ from the restriction on the values of $\mathbf{M}$:

$$\mathbb{E}_{\mathbf{M} \sim p(\mathbf{M}|\mathbf{S})} \|\mathbf{M}\|_0 = \sum_{(u,i) \in \mathcal{D}_p} s_{u,i}. \tag{3.13}$$

Based on this, we can obtain the feasible region $\mathcal{C}$ for $\mathbf{S}$. Here, with the condensation ratio $r = \frac{|\mathcal{D}_s|}{|\mathcal{D}|}$, we have:

$$\mathcal{C} = \{\mathbf{S} | 0 \leq s_{u,i} \leq 1, \|\mathbf{S}\|_1 \leq r|\mathcal{D}|\}. \tag{3.14}$$

Then, the discrete optimization described in Eq. (3.7) could be re-formulated into the following continuous one:

$$\min_{\mathbf{S} \in \mathcal{C}} \Phi(\mathbf{S}) = \mathbb{E}_p \mathcal{L}\left(\theta^*(\mathbf{M})\right), \text{ s.t. } \theta^*(\mathbf{M}) = \arg\min_{\theta} \hat{\mathcal{L}}(\theta; \mathbf{M}), \tag{3.15}$$

where p denotes the distribution $\mathbf{M} \sim p(\mathbf{M}|\mathbf{S})$. After we get the optimal probabilistic matrix $\mathbf{S}^*$ for the interactions, we can sample the corresponding generation matrix $\mathbf{M}^*$. Then, we can generate the condensed dataset $\mathcal{D}_s$ based on $\mathbf{M}^*$.

3.3.4 Optimization via Lightweight Policy Gradient Estimation

As we can observe from Eq. (3.15), the optimization is computationally expensive. It requires to differentiate through the model optimal, thus involving verbose gradient flow, to calculate the gradients for data updates. To be specific, according to chain-rule, the gradient calculation for the probability matrix $\mathbf{S}$ is presented in the form of:

$$\nabla_{\mathbf{S}}\Phi(\mathbf{S}) = \nabla_{\mathbf{S}}\theta^*(\mathbf{M})\nabla_{\theta}\mathcal{L}\left(\theta^*(\mathbf{M})\right), \tag{3.16}$$

where $\nabla_{\mathbf{S}}\theta^*(\mathbf{M})$ can lead to expensive and verbose gradient flow. It needs to unroll the verbose gradient flows over multiple inner optimization steps. Thus, the computational and memory demands associated with such calculation are substantial. Moreover, multiple epochs of data updates further exacerbates the cost since it requires multiple inner model convergences.

To address the aforementioned two issues, we propose Lightweight Policy Gradient Estimation (LPGE). The idea is two-fold: (1) In order to avoid the onerous gradient calculation to update $\mathbf{S}$, we propose to utilize policy gradient estimator [156]. This estimator only involves forward propagation to calculate the gradients for the probability matrix $\mathbf{S}$ updates. (2) To mitigate the computation cost from multiple inner convergences, we propose a lightweight substitute for the inner optimization, which adopts the one-step update strategy [52, 116]. The intuition behind is that the loss decreases after first step optimization of the inner model training is strong enough to provide comprehensive information to guide the outer data update. This technique is also verified to be effective empirically and theoretically in previous studies [52, 116, 54]. The overall optimization process is summarized in algorithm 1.

In the rest of this subsection, we elaborate on the details of policy gradient estimator and lightweight optimization for inner model training as well as providing theoretical

understanding on the convergence property of DConRec.

Algorithm 1 Condensing Rec-Dataset

Input: data pool $\mathcal{D}_p$, validation dataset $\mathcal{D}_v$, a recommender model θ, condensation ratio r.

Output: condensed dataset $\mathcal{D}_s$.

1: Initialize condensation probability $\boldsymbol{s}_{u,i} = r$
2: Generate compensation dataset $\mathcal{D}_t$ from $\mathcal{D}_p$.
3: **for** training iteration $t = 1, 2...T$ **do**
4: Sample matrix $\mathbf{M}$ based on the probability $\mathbf{S}$.
5: Randomly initialize the model θ_0.
6: Train the model for one epoch: `// Inner Optimization`
7: $\theta_1 \leftarrow \theta_0 - \nabla_{\theta_0} \hat{\mathcal{L}}_{rec}(\theta_0; \mathbf{M})$
8: Optimize $\mathbf{S}$ via Eq. 3.20 based on $\mathcal{D}_v$: `// Outer Optimization`
9: $\mathbf{S}^{t+1} \leftarrow \mathcal{P}_{\mathcal{C}} \left(\mathbf{S}^t - \eta \mathcal{L}_{\mathcal{D}_v} \nabla_{\mathbf{S}^t} \ln p(\mathbf{M} \mid \mathbf{S}^t) \right)$
10: Sample $\mathbf{M}$ based on the probability $\mathbf{S}^{t+1}$.
11: Synthesize $\mathcal{D}_s$ based on generation matrix $\mathbf{M}$.
12: **end for**
13: Return the condensed dataset $\mathcal{D}_s$

Policy Gradient Estimator (PGE)

To avoid the computationally expensive gradient flow of Eq. (3.16), we first transform the objective term in Eq. (3.15) as following:

$$\begin{aligned}
\nabla_{\mathbf{S}} \Phi(\mathbf{S}) &= \nabla_{\mathbf{S}} \mathbb{E}_{p(\mathbf{M}|\mathbf{S})} \mathcal{L}\left(\theta^*(\mathbf{M})\right) \\
&= \nabla_{\mathbf{S}} \int \mathcal{L}\left(\theta^*(\mathbf{M})\right) p(\mathbf{M}|\mathbf{S}) d\mathbf{M} \\
&= \int \mathcal{L}\left(\theta^*(\mathbf{M})\right) \frac{\nabla_{\mathbf{S}} p(\mathbf{M}|\mathbf{S})}{p(\mathbf{M}|\mathbf{S})} p(\mathbf{M}|\mathbf{S}) d\mathbf{M} \\
&= \int \mathcal{L}\left(\theta^*(\mathbf{M})\right) \nabla_{\mathbf{S}} \ln p(\mathbf{M}|\mathbf{S}) p(\mathbf{M}|\mathbf{S}) d\mathbf{M}
\end{aligned}$$

Algorithm 2 Algorithm of Projection $\mathcal{P}_{\mathcal{C}}(\cdot)$

Input: a matrix $\mathbf{S}_{in}$, condensation ratio r.

Output: projected matrix $\mathbf{S}_{out}$.

1: Calculate v_1 via:

2: $\quad \mathbf{1}^T \left[\min\left(1, \max\left(0, \mathbf{S}_{in} - v_1\mathbf{1}\right)\right)\right] - r|\mathcal{D}| = 0.$

3: $v_2 \leftarrow \max(0, v_1)$.

4: $\mathbf{S}_{out} \leftarrow \min\left(1, \max\left(0, \mathbf{S}_{in} - v_2\mathbf{1}\right)\right)$.

5: return $\mathbf{S}_{out}$

$$= \mathbb{E}_{p(\mathbf{M}|\mathbf{S})} \mathcal{L}\left(\theta^*(\mathbf{M})\right) \nabla_{\mathbf{S}} \ln p(\mathbf{M}|\mathbf{S}), \tag{3.17}$$

where $\mathcal{L}\left(\theta^*(\mathbf{M})\right) \nabla_{\mathbf{S}} \ln p(\mathbf{M}|\mathbf{S})$ is called policy gradient. Given the inner optimum $\theta^*(\mathbf{M})$, i.e., the well-trained model, we can directly calculate the gradient and update $\mathbf{S}$ as following:

$$\mathbf{S} \leftarrow \mathcal{P}_{\mathcal{C}}\left(\mathbf{S} - \eta \mathcal{L}\left(\theta^*(\mathbf{M})\right) \nabla_{\mathbf{S}} \ln p(\mathbf{M}|\mathbf{S})\right), \tag{3.18}$$

where $\mathcal{P}_{\mathcal{C}}(\cdot)$ is to project the updated values of probability matrix $\mathbf{S}$ back to the feasible region $\mathcal{C}$ of $\mathbf{S}$ and it is shown in algorithm 2.

Analysis on the effectiveness of PGE: From the design of PGE, shown in Eq. (3.16) and Eq. (3.17), we can observe that its optimization direction is towards minimizing the loss of downstream task $\mathcal{L}(\cdot)$ (e.g., BPR loss here) by synthesizing data, which is in line with the model training for good performance.

Analysis on the computational efficiency of PGE: from Eq. (3.18), we observe updating $\mathbf{S}$ involves: $\mathcal{L}\left(\theta^*(\mathbf{M})\right)$, $\nabla_{\mathbf{S}} \ln p(\mathbf{M}|\mathbf{S})$ and $\mathcal{P}_{\mathcal{C}}(\cdot)$. Computing $\mathcal{L}\left(\theta^*(\mathbf{M})\right)$ is efficient, only requiring forward propagation. This avoids the expensive and verbose gradient flow. Besides, the form of $\nabla_{\mathbf{S}} \ln p(\mathbf{M}|\mathbf{S})$ is quite simple and the projection $\mathcal{P}_{\mathcal{C}}(\cdot)$ is an efficient transformation as summarized in algorithm 2. Therefore, PGE is significantly faster than previous methods for data updates.

Lightweight Optimization

To further improve the overall computational efficiency, we propose the lightweight optimization. Specifically, we adopt the one-step update strategy [52, 116] for the inner model update and the policy gradient estimation in the Eq. (3.17) could be re-written as following:

$$\nabla_{\mathbf{S}}\Phi(\mathbf{S}) = \mathbb{E}_{p(\mathbf{M}|\mathbf{S})}\mathcal{L}\left(\theta_1(\mathbf{M})\right)\nabla_{\mathbf{S}}\ln p(\mathbf{M}|\mathbf{S}), \tag{3.19}$$

where $\theta_1(\mathbf{M})$ is obtained via one-step inner model update:

$$\theta_1(\mathbf{M}) \leftarrow \theta_0(\mathbf{M}) - \nabla_{\theta_0}\theta_0(\mathbf{M}).$$

Therefore, the gradient calculation for the probability matrix $\mathbf{S}$ (Eq. (3.18)) could be re-formalized as following:

$$\mathbf{S} \leftarrow \mathcal{P}_{\mathcal{C}}\left(\mathbf{S} - \eta\mathcal{L}\left(\theta_1(\mathbf{M})\right)\nabla_{\mathbf{S}}\ln p(\mathbf{M}|\mathbf{S})\right). \tag{3.20}$$

Convergence Property of LPGE

The following theorem reveals that the proposed lightweight policy gradient estimation is provably convergent and shows its convergence property.

Theorem 1. *Assuming that $\Phi(\mathbf{S})$ is L-smooth and the policy gradient variance $\mathbb{E}\|\mathcal{L}_{\mathcal{D}_{val}}\left(\theta_1(\mathbf{M})\right)\nabla_{\mathbf{S}}\ln p(\mathbf{M}|\mathbf{S}) - \nabla_{\mathbf{S}}\Phi(\mathbf{S})\|^2 \leq \sigma^2$,we denote the gradient mapping $\mathcal{G}^t$ at t-th iteration as*

$$\mathcal{G}^t = \frac{1}{\eta}\left(\mathbf{S}^t - \mathcal{P}_{\mathcal{C}}\left(\mathbf{S}^t - \eta\nabla_{\mathbf{S}}\Phi(\mathbf{S}^t)\right)\right).$$

Based on the above assumptions and setting the step size $\eta < \frac{1}{L}$, we can have following conclusion:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left\|\mathcal{G}^t\right\|^2 \leq \frac{8-2L\eta}{2-L\eta}\sigma^2,$$

when $T \to \infty$.

Before proving the theorem, we introduce two lemmas about the properties of the projection operator $\mathcal{P}_{\mathcal{C}}(\cdot)$ [87].

Lemma 1. *(firmly nonexpansive operators). Given a compact convex set $\mathcal{C} \subset \mathbb{R}^d$ and let $\mathcal{P}_{\mathcal{C}}(\cdot)$ be the projection operator on $\mathcal{C}$, then for any $\boldsymbol{u} \in \mathbb{R}^d$ and $\boldsymbol{v} \in \mathbb{R}^d$, we have*

$$\|\mathcal{P}_{\mathcal{C}}(\boldsymbol{u}) - \mathcal{P}_{\mathcal{C}}(\boldsymbol{v})\|^2 \leq (\boldsymbol{u} - \boldsymbol{v})^\top (\mathcal{P}_{\mathcal{C}}(\boldsymbol{u}) - \mathcal{P}_{\mathcal{C}}(\boldsymbol{v})) .$$

Lemma 2. *Given a compact convex set $\mathcal{C} \subset \mathbb{R}^d$ and let $\mathcal{P}_{\mathcal{C}}(\cdot)$ be the projection operator on $\mathcal{C}$, then for any $\boldsymbol{c} \in \mathcal{C}$ and $\boldsymbol{u} \in \mathbb{R}^d$, $\boldsymbol{v} \in \mathbb{R}^d$, we have*

$$\|\mathcal{P}_{\mathcal{C}}(\boldsymbol{c} + \boldsymbol{u}) - \mathcal{P}_{\mathcal{C}}(\boldsymbol{c} + \boldsymbol{v})\| \leq \|\boldsymbol{u} - \boldsymbol{v}\|.$$

Then, we provide the theorem proof, which is partially motivated by [52, 156].

Proof. In the implementation of algorithm 1, we calculate the gradient for updating $\mathbf{S}$ based on validation set $\mathcal{D}_{val}$, which is a subset randomly sampled from $\mathcal{D}$. Therefore, in the following, we denote the gradient via $\boldsymbol{g}^t$:

$$\boldsymbol{g}^t = \mathcal{L}_{val}\left(\boldsymbol{\theta}_1(\boldsymbol{m})\right) \nabla_{\mathbf{S}} \ln p\left(\boldsymbol{m} \mid \mathbf{S}^t\right).$$

Consequently, the updating of $\mathbf{S}$ in algorithm 1 could be re-written as:

$$\mathbf{S}^{t+1} = \mathcal{P}_{\mathcal{C}}\left(\mathbf{S}^t - \eta \boldsymbol{g}^t\right).$$

Let us denote the validation set based gradient mapping be:

$$\hat{\mathcal{G}}^t = \frac{1}{\eta}\left(\mathbf{S}^t - \mathcal{P}_{\mathcal{C}}\left(\mathbf{S}^t - \eta \boldsymbol{g}^t\right)\right) = \frac{1}{\eta}\left(\mathbf{S}^t - \mathbf{S}^{t+1}\right),$$

we have:

$$\begin{aligned}
&\Phi(\mathbf{S}^{t+1}) \leq \Phi(\mathbf{S}^t) + \left\langle \nabla\Phi(\mathbf{S}^t), \mathbf{S}^{t+1} - \mathbf{S}^t \right\rangle + \frac{L}{2}\|\mathbf{S}^{t+1} - \mathbf{S}^t\|^2 \\
&= \Phi(\mathbf{S}^t) - \eta \left\langle \nabla\Phi(\mathbf{S}^t), \hat{\mathcal{G}}^t \right\rangle + \frac{L\eta^2}{2}\|\hat{\mathcal{G}}^t\|^2 \\
&= \Phi(\mathbf{S}^t) - \eta \left\langle \nabla\Phi(\mathbf{S}^t) - \boldsymbol{g}^t + \boldsymbol{g}^t, \hat{\mathcal{G}}^t \right\rangle + \frac{L\eta^2}{2}\|\hat{\mathcal{G}}^t\|^2
\end{aligned}$$

$$
\begin{aligned}
&= \Phi(\mathbf{S}^t) - \eta \left\langle \boldsymbol{g}^t, \hat{\mathcal{G}}^t \right\rangle + \frac{L\eta^2}{2}\|\hat{\mathcal{G}}^t\|^2 + \eta \left\langle \delta^t, \hat{\mathcal{G}}^t \right\rangle \\
&\leq \Phi(\mathbf{S}^t) - \eta\|\hat{\mathcal{G}}^t\|^2 + \frac{L\eta^2}{2}\|\hat{\mathcal{G}}^t\|^2 + \eta \left\langle \delta^t, \hat{\mathcal{G}}^t \right\rangle \ \textit{//Lemma 1} \\
&= \Phi(\mathbf{S}^t) - (\eta - \frac{L\eta^2}{2})\|\hat{\mathcal{G}}^t\|^2 + \eta \left\langle \delta^t, \hat{\mathcal{G}}^t \right\rangle \\
&= \Phi(\mathbf{S}^t) - (\eta - \frac{L\eta^2}{2})\|\hat{\mathcal{G}}^t\|^2 + \eta \left\langle \delta^t, \mathcal{G}^t \right\rangle + \eta \left\langle \delta^t, \hat{\mathcal{G}}^t - \mathcal{G}^t \right\rangle \\
&\leq \Phi(\mathbf{S}^t) - (\eta - \frac{L\eta^2}{2})\|\hat{\mathcal{G}}^t\|^2 + \eta \left\langle \delta^t, \mathcal{G}^t \right\rangle + \eta\|\delta^t\|\|\hat{\mathcal{G}}^t - \mathcal{G}^t\| \\
&\leq \Phi(\mathbf{S}^t) - (\eta - \frac{L\eta^2}{2})\|\hat{\mathcal{G}}^t\|^2 + \eta \left\langle \delta^t, \mathcal{G}^t \right\rangle + \eta\|\delta^t\|^2. \ \textit{//Lemma 2}
\end{aligned}
$$

$\Rightarrow (\eta - \frac{L\eta^2}{2})\|\hat{\mathcal{G}}^t\|^2 \leq \Phi(\mathbf{S}^t) - \Phi(\mathbf{S}^{t+1}) + \eta \left\langle \delta^t, \mathcal{G}^t \right\rangle + \eta\|\delta^t\|^2.$

Then, we get:

$$
\begin{aligned}
&\sum_{t=1}^{T}(\eta - \frac{L\eta^2}{2})\|\hat{\mathcal{G}}^t\|^2 \\
&\leq \Phi(\mathbf{S}^1) - \Phi(\mathbf{S}^{T+1}) + \eta \sum_{t=1}^{T} \left(\left\langle \delta^t, \mathcal{G}^t \right\rangle + \eta\|\delta^t\|^2 \right).
\end{aligned}
$$

Since,

$$
\mathbb{E} \left\langle \delta^t, \mathcal{G}^t \right\rangle = \mathbb{E}_{\mathbf{S}^t}\mathbb{E}_{.|\mathbf{S}^t} \left(\left\langle \boldsymbol{g}^t - \nabla\Phi(\mathbf{S}^t)), \mathcal{G}^t \right\rangle |\mathbf{S}^t \right) = 0, \tag{3.21}
$$

and based on the assumption described as following:

$$
\mathbb{E}\|\delta^t\|^2 = \mathbb{E}\|\boldsymbol{g}^t - \nabla\Phi(\mathbf{S}^t)\|^2 \leq \sigma^2. \tag{3.22}
$$

we can get follows by by taking Eq. 3.21 and Eq. 3.22 back into Eq. 3.22:

$$
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\hat{\mathcal{G}}^t\|^2 \leq \frac{\Phi(\mathbf{S}^1) - \Phi^*}{(1 - L\eta/2)T} + \frac{\sigma^2}{1 - L\eta/2}. \tag{3.23}
$$

Since we have get:

$$
\mathbb{E}\|\mathcal{G}^t\|^2 \leq 2\mathbb{E}\|\hat{\mathcal{G}}^t\|^2 + 2\mathbb{E}\|\boldsymbol{g}^t - \nabla\Phi(\mathbf{S}^t)\|^2 \leq 2\mathbb{E}\|\hat{\mathcal{G}}^t\|^2 + 2\sigma^2, \tag{3.24}
$$

we can obtain follows by taking Eq. 3.23 into Eq. 3.24 and letting $T \to \infty$ to have:

$$
\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\mathcal{G}^t\|^2 &\leq \frac{2}{1 - L\eta/2}\left(\frac{\Phi(\mathbf{S}^1) - \Phi^*}{T} + (2 - L\eta/2)\sigma^2 \right) \\
&\to \frac{8 - 2L\eta}{2 - L\eta}\sigma^2.
\end{aligned}
$$

□

3.4 Experiments

In this section, we evaluate DConRec from these aspects:

- RQ1: How is the quality of condensed dataset?
- RQ2: How efficient is DConRec for condensation?
- RQ3: Can DConRec preserve diverse user preferences?
- RQ4: Can DConRec work well while adopting different recommendation models as backbone?
- RQ5: Are the condensed datasets generalizable to train different recommendation models?
- RQ6: How does the choice of proxy models for pre-augmentation influence qualities of condensed datasets?
- RQ7: How does condensation ratio influence the quality of condensed datasets?

3.4.1 Experimental Settings

Datasets. We evaluate the performance of our proposed framework on several real-world datasets: **Ciao** [97], **Gowalla** [10], **Dianping** [62] and **ML-20M** [42]. For each dataset, we divide the interactions into 80%/10%/10% for training/validation/testing. We conduct condensation/selection on the training set and train models on condensed datasets. Finally, we evaluate the performance on testing set. The statistics of the datasets can be found in Table 3.1.

Baselines. We compare our method with baselines that could be implemented on discrete interaction data: three selection-based methods (Random, Majority and

Table 3.1: Dataset statistics.

Dataset	#User	#Item	#Interactions	Density
Ciao	7,375	105,114	284,086	0.00037
Gowalla	29,859	40,989	1,027,464	0.00084
Dianping	16,396	14,546	51,946	0.00022
ML-20M	138,493	27,278	20,000,263	0.00529

SVP-CF [14, 92]) and the condensation-based methods (GradMatch [52, 150, 104] and Distill-CF [91]):

- Random randomly samples a portion of interactions from the training set to form the small datasets for model training.

- Majority samples interactions from users with lower degrees, thereby maximizing the inclusion of users within the dataset.

- SVP-CF[92] is a proxy-based coreset selection strategy for collaborative filtering datasets. We adopt MF as the proxy model.

- GradMatch is a prevalent condensation technique via matching the gradients of model parameters on condensed data and original data. We follow Doscond [52] and GCM [104] to implement the *efficient one-step gradient matching scheme* for the data synthesis.

- Distill-CF [91] A NTK-based data condensation method for creating summaries by synthesizing fake users and their corresponding interactions. Since we aims at condensing interaction sets in our experiment, we set the product of the number of fake users and their interactions to the desired condensation size for fair comparison. We utilize its original implementation for evaluation.

Backbone Models. In this paper, we conduct the evaluation on DConRec with following recommendation models:

- MF [94] is a traditional matrix factorization method that uses pairwise ranking loss based on implicit feedback.
- NGCF [111] is a graph-based CF method largely follows the standard GCN [57]. It additionally encodes the second-order feature interaction during message passing.
- LightGCN [47] is a state-of-the-art recommendation model based on GNNs, which improves performance by removing activation and feature transformation.
- XSimGCL [139] is a state-of-the-art recommendation model enhanced by self-supervised learning (SSL), which boosts the performance by a cross-layer contrastive learning.

Evaluation Protocol. We evaluate the top-K ranking performance of recommender models trained on the condensed datasets, and report *Recall@K* and *NDCG@K*, denoted as *R@K* and *N@K*. The evaluation consists of three stages. **(1) Dataset condensation/selection.** Given the original training set, each condensation (or data selection) method constructs a substantially smaller training set by synthesizing informative interactions (or selecting a representative subset) under a fixed condensation ratio/budget. This yields a compact training dataset that is used as a drop-in replacement for the full training data. **(2) Model training on the condensed data.** We then train a recommendation model using only the condensed dataset, while keeping the model architecture, optimization settings (e.g., learning rate, batch size, number of epochs), and early-stopping strategy consistent across different condensation methods to ensure fair comparison. **(3) Top-K ranking evaluation.** After training, for each test user (u), the model scores candidate items and produces a ranked list. Following the standard implicit-feedback evaluation protocol,

Table 3.2: Overall performance comparison."*Full Dataset*" indicates the performance of models trained on the original datasets."*R@K*" denotes Recall@K and"*N@K*" denotes NDCG@K."*Sel-based*" denotes selection based methods and "*Cond-based*" denotes dataset condensation methods.

Dataset		Ciao		Gowalla		Dianping		ML-20M	
Metrics		R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10
Sel -based	Random	0.0092	0.0076	0.0457	0.0327	0.0169	0.0091	0.0838	0.1050
	Majority	0.0180	0.0117	0.0667	0.0404	0.0236	0.0120	0.1176	0.1103
	SVP-CF	0.0289	0.0187	0.0836	0.0567	0.0523	0.0285	0.1572	0.1825
Cond -based	GradMatch	0.0124	0.0092	0.0548	0.0394	0.0200	0.0111	0.1519	0.1789
	Distill-CF	0.0169	0.0199	–	–	0.0191	0.0123	–	–
	DConRec	**0.0302**	**0.0200**	**0.0930**	**0.0651**	**0.0614**	**0.0340**	**0.1702**	**0.2021**
Full Dataset		0.0312	0.0212	0.1015	0.0720	0.0615	0.0346	0.2031	0.2414

the ground-truth held-out item(s) of user (u) in the test split are treated as positives, and the remaining unobserved items (or a sampled set of unobserved items) are treated as negatives. We rank the positive item(s) together with the negatives by their predicted scores, take the top-K items as recommendations, and compute *R@K* and *N@K* for each user. *R@K* measures whether the ground-truth item(s) appear in the top-K list, while *N@K* further accounts for the positions of the hits by assigning higher gains to relevant items ranked earlier. Finally, we average *R@K* and *N@K* across all test users; higher values indicate better ranking performance.

Implementation Details. For the sake of universality, we adopt MF as the backbone model for overall performance comparison in Table 3.2. The proxy model defaults to MF unless stated otherwise. When assessing cross-architecture performance and condensed dataset generalizability, we substitute MF with NGCF, LightGCN, and XSimGCL, setting the layer number to 2 or 3. For XSimGCL, the SSL weight is independently tuned for each dataset. The embedding dimension is set to 64. The

Table 3.3: Comparison of the running time for 100 epochs. "DConRec-Op" denotes the time of optimizing the synthesized dataset via policy gradient matching, without the time for inner model update.

Methods	GradMatch	DConRec	DConRec-Op
Ciao	8min43s	3min22s	**0min52s**
Gowalla	16min44s	13min11s	**4min11s**
Dianping	35min56s	30min19s	**10min19s**
ML-20M	162min47s	143min38s	**17min06s**

learning rate for condensation ranges $\{0.1, 0.01, 0.05, 0.005, 0.0005\}$ and the decay is 10 per 100 epochs, bounded by 1e-4. The pseudo data ratio r_{ps} ranges from 0.1 to 0.9. Model training utilizes the Adam optimizer with various initial learning rates, applying early stop. The environment includes a PyTorch (Version 1.12) on an Nvidia A30 GPU (Memory: 24GB, Cuda version: 11.3).

3.4.2 Performance Comparison

Quality of Condensed Datasets (RQ1). To evaluate the quality of the datasets condensed by DConRec, we measure the performance of MF trained on them. Specifically, with the condensation ratio r is set to 0.25, We report the metric *Recall@K* and *NDCG@K*, where K is set to 5 and 10. For the baselines, the condensation ratio is also set to 0.25. The results are summarized in Table 3.2, from which we have following observations:

- Superiority of DConRec over baselines: Our proposed *DConRec* outperforms the baselines across different datasets. Even when the size of datasets are reduced by 75%, we could achieve over comparable performance of the original datasets. Notably, on Dianping, we approximate 98% of the original performance with

Table 3.4: Performance comparison on different user groups. The bold denotes the best performance with the condensed data.

Users	Methods	Ciao		Gowalla		Dianping		ML-20M	
		R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10
Head	SVP	**0.0202**	**0.0372**	0.0468	0.0914	0.0306	0.0463	0.1144	0.2378
	GradMatch	0.0090	0.0222	0.0420	0.0794	0.0227	0.0349	0.1159	0.2283
	DConRec	0.0183	0.0309	**0.0343**	**0.0547**	**0.0397**	**0.0605**	**0.1205**	**0.3022**
	Full Data	0.0213	0.0438	0.0579	0.1168	0.0389	0.0617	0.1341	0.3256
Torso	SVP	0.0269	**0.0144**	0.1002	0.0534	0.053	0.0269	0.0317	0.0183
	GradMatch	0.0101	0.0063	0.0608	0.0373	0.0169	0.0082	0.0288	0.0139
	DConRec	**0.0257**	0.0138	**0.115**	**0.065**	**0.0632**	**0.0320**	**0.0390**	**0.0193**
	Full Data	0.0296	0.0164	0.1211	0.0680	0.0609	0.0315	0.0741	0.0399
Tail	SVP	0.0304	0.0190	0.0849	0.0558	0.0501	0.0304	0.1812	0.1414
	GradMatch	0.0116	0.0089	0.0608	0.0406	0.0252	0.01500	0.1856	0.1474
	DConRec	**0.0306**	**0.0205**	**0.0946**	**0.0634**	**0.0587**	**0.0355**	**0.2028**	**0.1560**
	Full Data	0.0329	0.0220	0.1024	0.0697	0.0634	0.0392	0.2328	0.1844
Overall	SVP	0.0289	0.0187	0.0836	0.0567	0.0523	0.0285	0.1572	0.1825
	GradMatch	0.0124	0.0092	0.0548	0.0394	0.0200	0.0111	0.1519	0.1789
	DConRec	**0.0302**	**0.0200**	**0.0930**	**0.0651**	**0.0614**	**0.0340**	**0.1702**	**0.2021**
	Full Data	0.0312	0.0212	0.1015	0.0720	0.0615	0.0346	0.2031	0.2414

only 25% data. However, those baselines underperform our proposed DConRec by a large margin. These results demonstrate the effectiveness of our method in preserving the information of the original datasets.

- Direct application of GradMatch to recommendation may lead to unsatisfactory results: We observe that the GradMatch does not perform well. We argue that this is due to its design tailored for data with continuous features (e.g., images, node features, etc). Thus, direct application of it in generating interactions cannot capture adequate information about users' preference for condensed dataset. Therefore, it cannot even beat Majority and SVP-CF on three datasets, both of

which are heuristic selection methods and they can capture partial preference information.

Condensing Efficiency (RQ2). Since we also aim to achieve an efficient condensation method, we evaluate the running-time of DConRec and compare it with GradMatch. Further, to examine the efficiency of the proposed optimization, we also compare a variant of DConRec, which only counts the time of optimizing dataset without the time for inner model training. We denote the variant by DConRec-Optim. Under the same setting of Table 3.2, We run 100 epochs for boths methods and record the time. The result is summarized in Table 3.3. We can observe that, in terms of updating dataset, DConRec can be 8× faster than GradMatch, which verifies the efficiency of our proposed method. Further, via removing the time of inner training, we can find that the real optimization time of DConRec is further shorter. The efficiency of DConRec stems from its design of forward calculation (PGE) to update the synthetic dataset, avoiding the time-consuming process of backpropagation, which is required in the gradient matching scheme of GradMatch.

Performance on Different User Groups (RQ3). We investigate the ability of DConRec in preserving the information for different groups of users and we report the results on four datasets in Table 3.4. Specifically, we split the users into three groups: Head (100 <num. of interections), Torso (10 <num. of interections <100) and Tail (num. of interections <10). Here, we adopt MF as the backbone model and the test model. From Table 3.4, we can observe that our proposed method can preserve the preference information in most cases for users in all three groups when compared to other baselines, close to the performance of model trained on full data. In particular, while other methods suffer significant performance decrease on the tail user group, our method still perform excellently comparable.

Training Efficiency. Due to the large size of the real-world recommendation datasets, it is quite expensive to train the models based on them. In this part, we aim to exam-

Table 3.5: Generalization of DConRec: equip it with different backbone models (RQ4) and utilize the condensed data to train various models (RQ5).

Backbone	Test Model	Ciao		Gowalla		Dianping		ML-20M	
		R@10	N@10	R@10	N@10	R@10	N@10	R@10	N@10
MF	MF	0.0302	0.0200	0.0930	0.0651	0.0606	0.0335	0.1702	0.2021
	NGCF	0.0282	0.0179	0.0891	0.0624	0.0602	0.0332	0.1648	0.1986
	LightGCN	0.0300	0.0201	0.0914	0.0644	0.0377	0.0202	0.1529	0.1824
	XSimGCL	0.0293	0.0194	0.0812	0.0565	0.0545	0.0298	0.1716	0.2055
LightGCN	MF	0.0289	0.0186	0.0932	0.0652	0.0605	0.0334	0.1698	0.2017
	NGCF	0.0282	0.0185	0.0893	0.0623	0.0606	0.0334	0.1645	0.1980
	LightGCN	0.0299	0.0197	0.0915	0.0643	0.0378	0.0201	0.1527	0.1822
	XSimGCL	0.0298	0.0195	0.0898	0.0624	0.0540	0.0295	0.1749	0.2095
NGCF	MF	0.0289	0.0184	0.0920	0.0643	0.0609	0.0338	0.1501	0.1783
	NGCF	0.0279	0.0186	0.0868	0.0611	0.0607	0.0335	0.1655	0.1988
	LightGCN	0.0295	0.0195	0.0919	0.0644	0.0296	0.0155	0.1749	0.2078
	XSimGCL	0.0299	0.0193	0.0890	0.0621	0.0540	0.0297	0.1740	0.2086
XSimGCL	MF	0.0299	0.0190	0.0914	0.0639	0.0604	0.0334	0.1700	0.2017
	NGCF	0.0277	0.0185	0.0884	0.0618	0.0601	0.0333	0.1650	0.1983
	LightGCN	0.0299	0.0197	0.0912	0.0639	0.0377	0.0204	0.1292	0.1558
	XSimGCL	0.0295	0.0930	0.0805	0.0563	0.0537	0.0293	0.1749	0.2094

ine the training efficiency of the recommender models on our condensed dataset. We record the training curves of MF and LightGCN on condensed Dianping dataset and original Dianping dataset. As shown in Figure 3.5a and 3.5b, the results demonstrate that training on the condensed dataset is greatly more efficient than training on the original dataset. Notably, the training on the condensed dataset is more efficiency (speedup over 6×) with comparable performance.

3.4.3 Generalization of DConRec

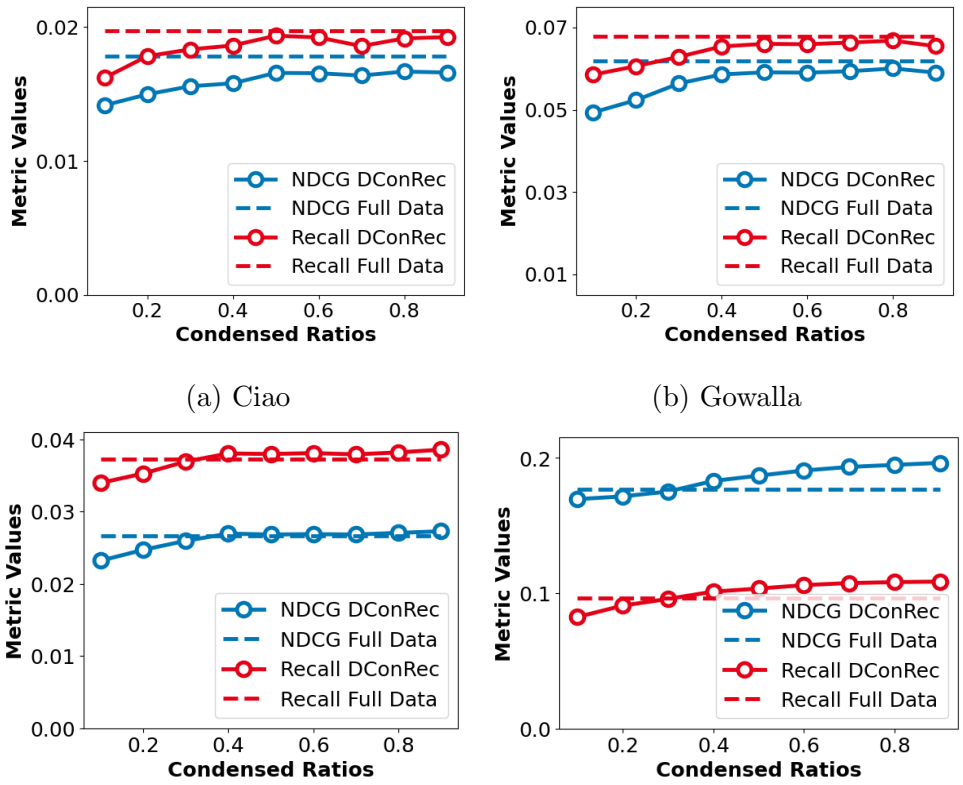
In this subsetion, we examine the generalizability of the condensed datasets to train different models and the generalizability of DConRec cross different architectures of backbone model on the four datasets.

Different Architectures of Model as Backbone (RQ4). The proposed DConRec is highly flexible since we can adopt various architectures for the backbone model. We investigate the performances of DConRec when equipping it with different backbone models (i.e., MF, NGCF, LightGCN and XSimGCL). From Table 3.5, we can see that the dataset condensed with different recommendation models as backbones all elaborate strong generalizability on other models. Interestingly, we can observe that XSimGCL yields the best performance regarding the dataset condensed by the DConRec with the same backbone model while setting XSimGCL to be the backbone model may not produce the best testing performance regarding the same test model.

Different Architectures of Model for Testing (RQ5). Specifically, we test the performance when using dataset condensed by one recommendation model to train various recommenders. As depicted in Table 3.5, we choose MF [94], NGCF [111], LightGCN [47] and XSimGCL [139] to evaluate the quality of the condensed datas, observing that the dataset condensed by one specific model is also informative to train models of other architectures. Among them, we can observe that XSimGCL outperforms other models regarding the same condensed datasets under most settings, consistent with the performances trained on the original datasets.

3.4.4 Proxy Models Study in Pre-augmentation (RQ6)

In this section, we investigate the effects of adopting different proxy models for DConRec to produce the condensed datasets. In Table 3.6, we report the performances of varying the proxy models among MF [94], NGCF [111], LightGCN [47]



(a) Ciao (b) Gowalla

(c) Dianping (d) ML-20M

Figure 3.4: Vary condensed ratios. The solid lines denote the performance with condensed datasets, while the dashed lines indicate those with full ones.

and XSimGCL [139], reporting the testing evaluation with model MF. We can observe that DConRec can work well with different architectures of proxy model. Interestingly, more powerful proxy model can contribute to better performance (e.g., adopting XSimGCL as the proxy model yields the best performance on these four datasets). We argue that a more powerful proxy model can better capture preferences, thereby more effectively including potential user interests.

Table 3.6: Effects of different architectures of poxy model.

Dataset	Proxy Model	R@5	R@10	N@5	N@10
Ciao	MF	0.0180	0.0302	0.0163	0.0200
	NGCF	0.0175	0.0293	0.0148	0.01863
	LightGCN	0.0193	0.0297	0.0166	0.01968
	XSimGCL	0.0193	0.0328	0.0167	0.02083
Gowalla	MF	0.0622	0.0930	0.0558	0.0651
	NGCF	0.0609	0.0909	0.0545	0.06370
	LightGCN	0.0664	0.0963	0.0590	0.0680
	XSimGCL	0.0683	0.0996	0.0607	0.07018
Dianping	MF	0.0370	0.0614	0.0259	0.03400
	NGCF	0.0369	0.0606	0.0256	0.03340
	LightGCN	0.0396	0.0639	0.0279	0.03595
	XSimGCL	0.0407	0.0657	0.0287	0.03688
ML-20M	MF	0.1090	0.1702	0.2010	0.2021
	NGCF	0.1086	0.1703	0.1998	0.2012
	LightGCN	0.1099	0.1721	0.2027	0.2034
	XSimGCL	0.1105	0.1750	0.2088	0.2107

3.4.5 Varying Condensation Ratios (RQ7)

In this section, we evaluate DConRec under different condensation ratios, ranging from 0.1 to 0.9. We report the results regarding Recall@5 and NDCG@5 in Figure 3.4.

Results. We can observe that the performance is roughly proportional to the size of the condensed dataset since larger size denotes more information from original dataset. As depicted from Figure 3.4, at most cases (even when the condensation ratio is 0.1), the datasets condensed by our method are qualified and the performance

of model trained on them is close to those trained on full datasets.

Discussion. We attribute the decent performance to the ability of DConRec in preserving the potential preference of users since the key to successful recommendation is modeling personalized preference. Furthermore, since not all interactions contribute positively to preference modeling, their reduction during condensation may not hinder personalized interests modeling.

Table 3.7: Ablation study on the pre-augmentation.

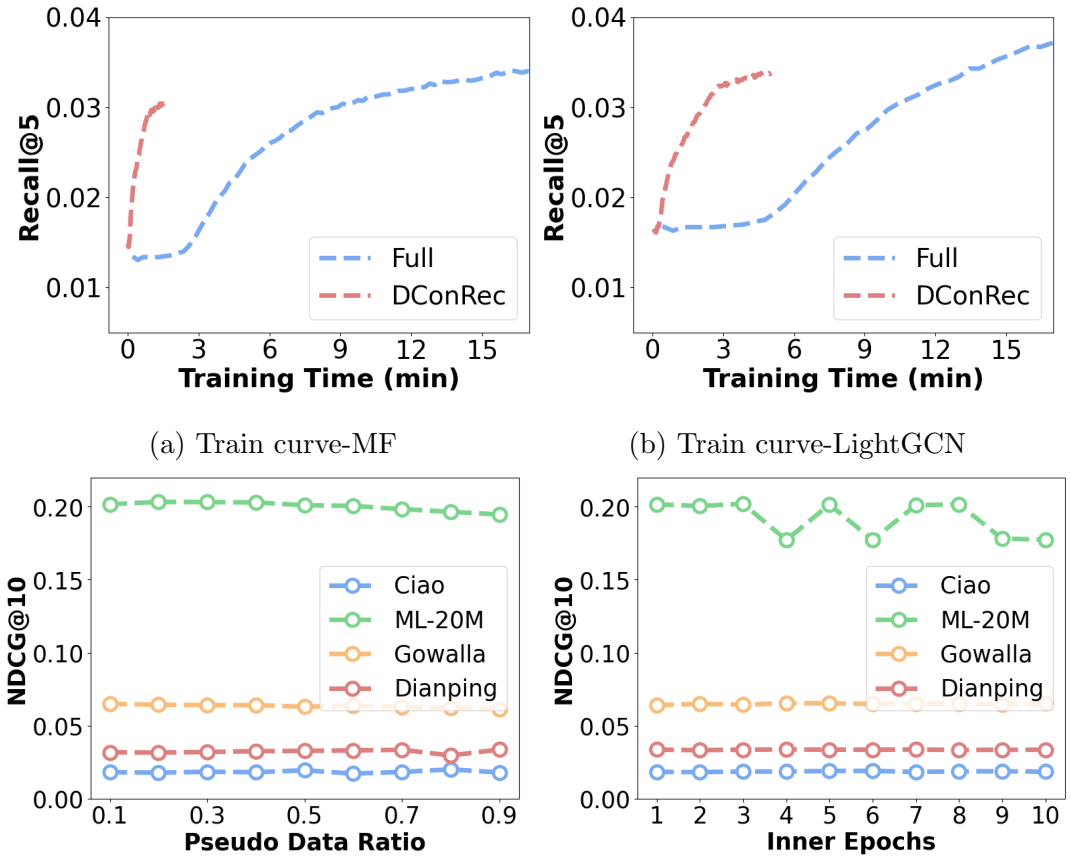
Dataset	Metrics	w\o-PA	DConRec
Ciao	R@10	0.0161	0.0302
	N@10	0.0115	0.0200
Gowalla	R@10	0.0771	0.0930
	N@10	0.0545	0.0651
Dianping	R@10	0.0286	0.0614
	N@10	0.0155	0.0340
ML-20M	R@10	0.1304	0.1702
	N@10	0.1569	0.2021

3.4.6 Further Investigations

In this section, we further explore following questions: (1) What is the effect of pre-augmentation? (2) How does performance vary with different sizes of pseudo data? (3) How do varying inner training epochs affect performance? (4) Can embeddings learned from condensed datasets replicate the patterns of those from the full dataset?

Ablation Study on Pre-augmentation. To investigate the effect of pre-augmentation on the performance of DConRec, we study the variant of our method by removing

the module of pre-augmentation, indicated by **w\o-PA** in Table 3.7. The results regarding Recall@10 and NDCG@10 on four datasets are summarized in Table 3.7 and we can see that the elimination of pre-augmentation lead to great performance degradation. Besides, in Figure 3.1a, we can also observe that while directly training on the pre-augmented original dataset, the performance improvement is limited, which indicate that the pre-augmentation specifically benefit the condensation quality. This also demonstrates the necessities of pre-augmentation in preserving the user preference.



(a) Train curve-MF

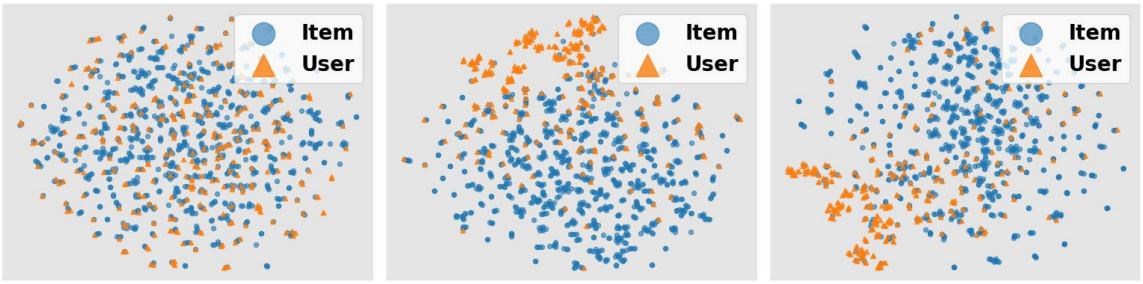
(b) Train curve-LightGCN

(c) Vary pseudo data ratio

(d) Lightweight update

Figure 3.5: Various analysis.

Varying Sizes of Pseudo Dataset in Pre-augmentation. We study how different sizes of pseudo dataset affect the performance. Concretely, we vary the pseudo data



(a) Random (b) *DConRec* (c) Full Dataset

Figure 3.6: T-SNE visualization of embeddings learned by MF.

ratio r_{ps} from 0.1 to 0.9. As the reported results on dataset Ciao in Figure 3.5c, we find that more pseudo data does not invariably yield better performance, some times even worse. We posit that this arises from the fact that a significant portion of the pseudo data might introduce noise and a larger size of parameters to be optimized for the generation of condensed dataset, in line with the discussion in section 3.4.5.

Investigation on the Lightweight Update Scheme. To further investigate the effect of lightweight update scheme, we vary the inner epochs from 1 to 10 and test the performance on dataset Ciao. As illustrated in Figure 3.5d, we can observe that as the number of inner model training increase, the performance of DConRec does not vary a lot. This verifies the effectiveness of the lightweight design and it is in line with the results in previous studies [52, 116].

Visualization To further examine the quality of the condensed datasets synthesized by our method, we utilize t-SNE [101] to visualize the learned user and item embeddings. We train MF on datasets produced by *Random*, *DConRec* and *Full Dataset*. We set the condensation ratio to 0.25 and provide the t-SNE plots of the learned embeddings for Ciao in Figure 3.6. It is observed that embeddings learned by randomly selecting interactions deteriorate the patterns in original dataset, which changed the personalization of users. In contrast, the embeddings learned by our method preserve similar patterns to the one with full dataset. This makes the comparable performance of DConRec explainable.

3.5 Related Work

Recommender Systems A prevalent paradigm of the recommendation always decomposes the interaction data into embedding vectors of users and items. One of the most popular techniques is collaborative filtering [30, 121, 23, 5, 136, 107, 114, 115]. It is based on the assumption that users behaving similarly are likely to share similar preference [127]. For instance, the pioneering study (matrix factorization) [89, 94, 76] aims to learn representations for each user and item, performing inner product between them to predict the preference. Later on, deep neural networks [49, 22, 45, 9, 147, 33, 78, 79] brought promising application in recommendation tasks. For example, NeuMF [49] is proposed to replace the inner product of MF model with non-linear neural networks to predict the preference of users. Then, NGCF [111] models the user-item interactions as a graph and adopt the message passing among the nodes to learn representations. Further, LightGCN [47] proposes to eliminate the feature transformation and non-linear activation of NGCF. Inspired by the success of self-supervised learning (SSL), the XSimGCL [139] devises a SSL-enhanced method for recommendation, achieving promising performance.

Dataset Condensation. The goal of dataset condensation is to synthesize a small set of synthetic samples, whose training performance on deep neural networks can be comparable with the original one. There are two types of condensation methods [16, 34]: matching-based approaches and kernel-based approaches. The first line of research is kernel-based approaches. Some works [91, 85] reduce the bi-level optimization into a single-level one via kernel ridge regression (NTK). Distill-CF [91] utilizes NTK to model recommendation data and synthesizes fake users via reconstruction. As for another line, gradient matchings between the model trained on small-condensed dataset and the large-real dataset is proposed by [150, 148] to generate synthetic datasets. To avoid high computation of matching gradients, [149] proposes feature matching. Some researchers investigate condensing graphs [52, 155, 71, 43, 38, 37, 36]. To make the condensation more flexible and adaptable to different

architectures of models, some studies [145, 4] propose methods that are based on generative model. GCM [104] proposes an efficient condensation method for data in CTR predictions.

Since they involve expensive and verbose gradient flow, we propose the forward lightweight policy gradient estimator for data updates, which is significantly faster. Besides, to protect user's potential preference, we synthesize data based on pre-augmented data instead of original dataset. Overall, these components make our framework distinct from prior DC approaches that (i) primarily target continuous inputs/features, (ii) synthesize data directly from the original dataset without explicitly addressing limited exposure, and (iii) rely on computationally heavy gradient-matching-style updates. In contrast, our method is designed for *discrete interaction condensation*, explicitly accounts for *potential preference preservation* through pre-augmentation, and adopts an *efficient forward update* strategy that substantially improves the practicality of condensation for recommendation.

3.6 Conclusion

In this paper, we proposed a novel condensation framework for recommendation data, called DConRec, aiming to condensing interaction set. It can efficiently generate discrete interactions based on pre-augmented data via lightweight policy gradient estimation. Our empirical investigations verify the effectiveness and efficiency of the proposed framework. Notably, we reduce the dataset size by 75% while approximating over 98% of the original performance on Dianping and over 90% on other datasets. Besides, our method is significantly faster than the prevalent methods (e.g., 8× faster than methods based on gradient matching on Ciao). Moreover, we examine its convergence.

Chapter 4

Leveraging ChatGPT to Empower Training-Free Dataset Condensation for Content-Based Recommendation

Modern Content-Based Recommendation (CBR) techniques utilize item content to deliver personalized services, effectively mitigating information overload. However, these methods often require resource-intensive training on large datasets. To address this issue, we explore dataset condensation for textual CBR in this paper. Dataset condensation aims to synthesize a compact yet informative dataset, enabling models to achieve performance comparable to those trained on full datasets. Applying existing approaches to CBR presents two key challenges: (1) the difficulty of synthesizing discrete texts and (2) the inability to preserve user-item preference information. To overcome these limitations, we propose TF_DCon, an efficient dataset condensation method for CBR. TF_DCon employs a prompt-evolution module to guide ChatGPT in condensing discrete texts and integrates a clustering-based module to condense

user preferences effectively. Thanks to the forward paradigm, TF_DCon avoids the iterative updates on the condensed data, significantly enhancing the efficiency. Extensive experiments conducted on three real-world datasets demonstrate TF_DCon's effectiveness. Notably, we are able to approximate up to 97% of the original performance while reducing the dataset size by 95% (i.e., on dataset MIND) and speed up over 7× when compared to conventional condensation method. Our code is available at: `https://github.com/Jyonn/TF-DCon`.

4.1 Introduction

Content-based recommenders [120, 117, 119] have made strides in mitigating the information overload dilemma, delivering items with relevant content (i.e., news, articles, or movies) to users. Existing advanced content-based recommendation (CBR) models [106, 118] are trained on large-scale datasets encompassing millions of users and items. Handling these large datasets heavily strains computational resources during model training. Furthermore, the recurrent retraining of recommendation models, especially for periodic updates in real-world applications, exponentially elevates costs to unsustainable levels.

Dataset condensation, also called *dataset distillation*, offers promising solution to these issues. The goal of dataset condensation is to synthesize a small yet informative dataset, trained on which the model can achieve comparable performance to those of models trained on the large original dataset [110, 53, 155, 52, 150, 40]. The conventional paradigm [150, 52, 95] formulates the condensation as bi-level optimization problem and iteratively update the condensed data by matching gradients of network parameters between synthetic and original data. Such methods have achieved success on condensing continuous data, such as images and node features. However, condensing datasets for CBR is still unexplored.

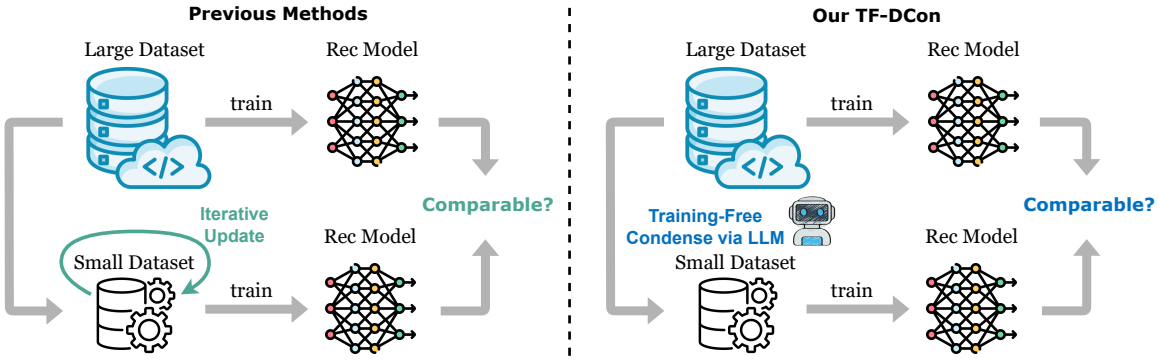


Figure 4.1: Dataset Synthesis Pipelines Comparison between TF-DCon and Prior Methods in Other Domains.

To bridge this gap, we investigate how to effectively condense the dataset for the textual content based recommendation. Existing condensation approaches devised in other domains [150, 53, 52, 40] face two main challenges in the context of CBR: (1) *How to generate discrete textual data?* Current condensation methods are designed for continuous data (i.e., images and text embeddings), following a formulation of nested bi-level optimization. Under this formulation, the data is synthesized via the outer gradient, which cannot be utilized to generate discrete textual data. Therefore, a solution for text synthesis is essential. (2) *How to preserve the preference information between users and items?* In recommendation tasks, user and item data, along with their interactions, are critical for inferring user preferences. However, previously proposed methods mostly involve classification tasks, limiting their ability to handle interaction data and retain preference information.

Large Language Models (LLMs), such as ChatGPT developed by OpenAI, have shown a strong capacity to distill textual information [3, 12, 64], and its general expertise across various fields such as medicine [1, 2], recommendation [153, 18], and law [15, 11]. With their versatility and extensive world knowledge, LLMs exhibit emergent abilities such as davanced text comprehension and language generation [3, 12]. Leveraging these capabilities [3, 12], we propose to leverage ChatGPT for dataset condensation, processing textual content to address one of the aforementioned challenges.

While LLMs are highly effective at processing text, they lack specific domain knowledge for recommendation scenarios and the ability to capture personalized user preferences. Naïvely applying LLMs for recommendation data condensation risks losing both this domain knowledge and essential preference information. To this end, we propose a ChatGPT-powered **T**raining-**F**ree **D**ataset **Con**densation method for content-based recommendation, abbreviated as **TF-DCon**. TF-DCon is devised in a two-level manner: *content-level* and *user-level*. At content-level, we curate a prompt-evolution module to optimize prompts, enabling ChatGPT to adapt to the specific recommendation domain and condense each item's information into an informative title. At user-level, to capture the preference information of users, we propose a clustering-based synthesis module to simultaneously generate fake users and their corresponding historical interactions, based on user interests extracted by ChatGPT and user embeddings. Consequently, our approach offers several advantages: (1) TF-DCon omits the conventional paradigm of formulating condensation as a bi-level optimization problem [150, 110, 53]. Instead, we formulate condensation as a forward process, eliminating iteratively training on datasets and greatly enhancing the condensation efficiency (see Figure 4.1); (2) TF-DCon can generate text, which makes the condensed dataset more generalizable and flexible to train different architectures of models for CBR. (3) TF-DCon seamlessly embeds user preferences into the condensed data through the clustering-based synthesis module. Our contributions are summarized as follows:

- **Objective.** This work is the first to explore dataset condensation for textual CBR. We introduce a forward condensation paradigm distinct from traditional bi-level optimization and aim to effectively condense the information of items and user preference.
- **Methodology.** We propose a ChatGPT-driven dataset condensation method featuring a prompt-evolution module for item content condensation and a clustering-based synthesis module to generate synthetic users and interactions, thereby

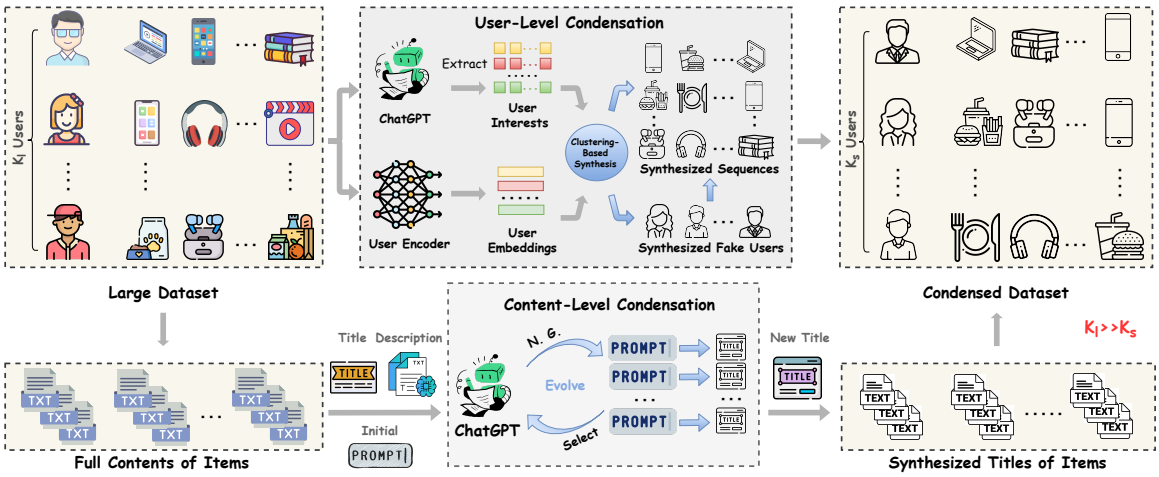


Figure 4.2: The Proposed Method in a Nutshell. The outline of this method is presented in Algorithm 3 and the details of the prompt's evolution are outlined in Algorithm 4. "N. G." denotes "Next Generated Prompt Candidates". Given a prompt, the content-level condensation and user interest extraction are instanced in Figure 4.3 and Figure 4.4.

preserving both item and preference information effectively.

- **Experiment.** Extensive experimental results demonstrate the efficacy of TF-DCon. Particularly, we can approximate up to 97% of the original performance while reducing the dataset size by 95% (i.e., on the dataset MIND). The model training on the condensed datasets is significantly faster (i.e., over 7× speedup compared to TextE. in Table 4.6).

4.2 Preliminaries

4.2.1 Content-based Recommendation

Let us first introduce key concepts and formulation of content-based recommendation. CBR aims to infer the preference of users based on historical interactions and then

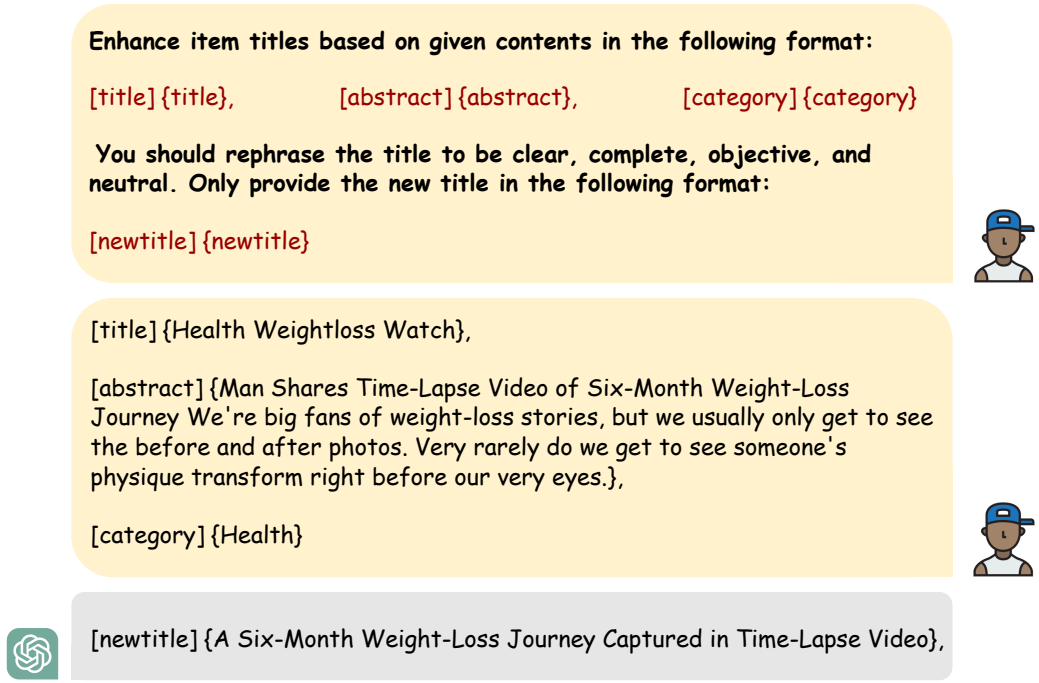


Figure 4.3: Instance of Content-level Condensation.

recommend the contents with potential interest. In the context of CBR, the dataset $\mathcal{D}$ mainly consists of item set $\mathcal{N}$, user set $\mathcal{U}$, and click history set $\mathcal{H}$. Each item $n \in \mathcal{N}$ contains various contents, such as title, abstract, and category. Each user $u \in \mathcal{U}$ is associated with a clicking history of items $h^{(u)} \subset \mathcal{N}$. We denote the set of history $h^{(u)}$ by $\mathcal{H}$. Let $\mathcal{C}$ denote the set of clicks, where each click $c \in \mathcal{C}$ is defined as a tuple (u, n) indicating that user u has clicked on item n. The task of CBR is to infer the preference of users on given candidate items based on their historical interactions.

4.2.2 Dataset Condensation

The aim of dataset condensation is to synthesize a small yet informative dataset $\mathcal{S}$, trained on which the recommendation model can achieve close performance to the model trained on full dataset $\mathcal{D}$. Here, the synthesized dataset $\mathcal{S}$ also consists of its corresponding item set $\mathcal{N}^{\mathcal{S}}$, user set $\mathcal{U}^{\mathcal{S}}$ and click history set $\mathcal{H}^{\mathcal{S}}$. Formally, the

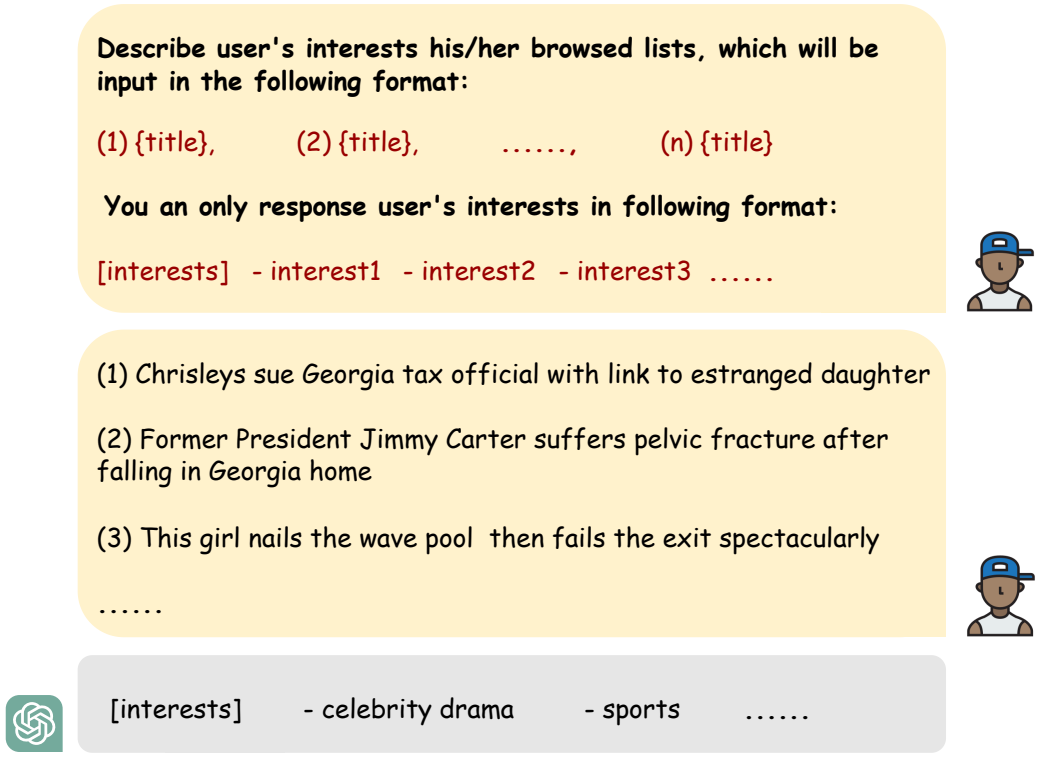


Figure 4.4: Instance of User Interest Extraction.

objective of the condensation could be formulated as follows:

$$\min_{\mathcal{S}} \mathcal{L}(f_{\boldsymbol{\theta}^{\mathcal{S}}}(\mathcal{D})), \text{ s.t. } \boldsymbol{\theta}^{\mathcal{S}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(f_{\boldsymbol{\theta}}(\mathcal{S})), \tag{4.1}$$

where $\boldsymbol{\theta}$ is the parameter of recommendation model f and $\mathcal{L}$ is the loss function. As we can see in Equation (4.1), the condensation is a bi-level problem, where the outer optimization is to synthesize the condensed dataset and the inner optimization is to train the model on dataset $\mathcal{S}$.

4.3 Method

In this section, we detail the proposed method (TF-DCon) and the overview is shown in Figure 4.2. Content-level condensation reduces the textual data load for each item (Section 4.3.1), while user-level condensation synthesizes fake users and interactions,

Algorithm 3 TF-DCon

Input: dataset $\mathcal{D} = (\mathcal{H}, \mathcal{N})$ where $\mathcal{H}$ is the set of users' historical interactions and $\mathcal{N}$ is the set of items, number of synthesized users in the condensed dataset K.

1: Utilize ChatGPT to extract user interests:

$$\mathcal{I} \leftarrow \text{ChatGPT}(\mathcal{H}, \mathcal{N})$$

2: Encode user interests and user histories:

$$H \leftarrow \text{UserEncoder}(\mathcal{D}),\ H' \leftarrow \text{TextEncoder}(\mathcal{I})$$

3: Cluster to generate K synthetic users and their interactions:

$$\mathcal{H}^{con} \leftarrow \text{Clustering}(K, H, H')$$

4: Utilize ChatGPT to condense the content of items:

$$\mathcal{N}^{con} \leftarrow \text{ChatGPT}(\mathcal{N})$$

5: **return** condensed dataset $\mathcal{D}^{con} = (\mathcal{H}^{con}, \mathcal{N}^{con})$

capturing primary interests and merging users with similar preferences (Section 4.3.2). Section 4.3.3 compares the computational complexity of different methods.

4.3.1 Content-level Condensation

Content Condensation

In the content-level condensation, we aim to condense all the information of each item into a succinct yet informative title. Recent studies [17, 99] reveal the exceptional power of large language models in processing textual contents. Therefore, we propose to utilize ChatGPT to condense the contents. Specifically, we design the guiding prompt in the following format:

> Hints on the format of input:
> *[title]*{*title*}*, [abs]*{*abs*}*,...*
> Instructions on the format of output:
> *[new_title]*{*new_title*}

After facilitating ChatGPT's comprehension on the input, the information of items will be fed into ChatGPT to generate the condensed title. Formally, the process can be described as follows:

$$n_s = ChatGPT(n), \tag{4.2}$$

where $n \in \mathcal{N}$ is the original contents of item, consisting of [*title*], [*abstract*] and [*category*], and $n_s \in \mathcal{N}^{\mathcal{S}}$ is the condensed title of the item, which only contain the [*title*]. An instance of the content-level condensation is depicted in Figure 4.3, where an instance of condensing the information of a movie into a new title is presented.

Prompt Evolution

To efficiently guide ChatGPT to adapt to the recommendation scenario and perform thorough content condensation for each item, we propose a curated prompt evolution method, outlined in Algorithm 4. Given an initial prompt, EvoPro evolves the prompts over a pre-defined times. During i-th iteration of evolving, EvoPro first instructs ChatGPT to generate N next-generation prompts $\{prompt_n\}$ based on gen_{i-1}-*prompt*. Then, these prompts are utilized to instruct ChatGPT to condense item contents for TF-DCon. The similarity scores between the embeddings of condensed contents and the original contents will be assigned to each prompt in $\{prompt_n\}$, outlined in Algorithm 5. The highest-scoring prompt is selected as gen_i-*prompt* for i-th generation. When the evolution finished,

the prompt with the hightest score in the latest generation will be selected as the prompt for content condensation. The prompt selection is based on the rationale that a condensed title with the highest similarity to the original content retains the most information.

Algorithm 4 EvoPro

Input: gen_0-*prompt*, item contents $\mathcal{C} = \{content_i\}$, evolving times E, number of prompts per generation N.

Initialize prompts set $\mathcal{P} \leftarrow \{\}$ and counter $e \leftarrow 0$
while $e < E$ **do**
    $e \leftarrow e + 1$
    Use ChatGPT to generate gen_{e-1}-*prompt*'s N next generation prompts $\{prompt_n\}$
    Initialize the score set $\mathcal{S} \leftarrow \{\}$
    for all *prompt* in $\{prompt_n\}$ **do**
        Calculate score for *prompt*:
        $s \leftarrow \text{CalScore}(prompt, \mathcal{C})$
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{s\}$
    end for
    Select the highest-scoring prompt in $\mathcal{S}$ to be gen_e-*prompt*
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{gen_e\text{-}prompt\}$
end while
return evolved prompts $\mathcal{P}$

4.3.2 User-level Condensation

In user-level condensation, we aim to condense the number of users, reducing the total size of the interactions. First, we introduce the interest extraction for each user and user encoder. Then, we propose the clustering-based synthesis module to synthesize fake users and their corresponding historical sequences. In terms of user synthesis, we apply the clustering method to the user embeddings and each cluster corresponds to a synthetic user. To synthesize interactions for each user, we merge the interactions of users, whose embedding distance and interest distance to their cluster centroids rank top-m while in descending order.

Algorithm 5 CalScore

Input: *prompt*, item contents $\mathcal{C}$.

$s \leftarrow 0$
for all c_i in $\mathcal{C}$ **do**
    Condense item content: $con_i \leftarrow \text{LLM}(prompt, c_i)$
    Encode contents the condensed contents:
        $h_i \leftarrow \text{TextEncoder}(c_i)$, $h_i' \leftarrow \text{TextEncoder}(con_i)$
    Calculate the similarity score: $s \leftarrow s + \text{Sim}(h_i, h_i')$
end for
return s

1) Interests Extraction and User Encoder:

Interests Extraction: Each user u is associated with a click history $h^{(u)}$, based on which we will utilize ChatGPT to extract the set of interests $\mathcal{I}^{(u)}$ for this user. Specifically, we design the guiding prompt in the following format:

> Hints on the format of input:
> *(1){title}, (2){title}, (3){title}, ...*
> Instructions on the format of output:
> *[interests] -interest1, -interest2, ...*

Then, the click history $h^{(u)}$ will be fed into ChatGPT to generate the interests of user u, which could be formulated as follows:

$$\mathcal{I}^{(u)} = ChatGPT(h^{(u)}), \tag{4.3}$$

where $\mathcal{I}^{(u)}$ is the set of interests for user u.

User Encoder: For each user u, given the associated click history $h^{(u)}$, we first we obtain the user's embedding:

$$z_u = f_\theta(h^{(u)}, \mathcal{N}), \tag{4.4}$$

where $\mathcal{N}$ is item set and f_θ is user encoder [118, 117, 119].

2) Clustering-based Synthesis:

User Synthesis: Given the users' embeddings, we apply K-means algorithm over all of them to obtain their cluster centroids $\{c_i\}_{i=1}^{K_s}$, where K_s is the number of synthesized users in the condensed dataset.

Historical Sequence Synthesis: Given that each cluster corresponds to a fake user u_s, we need to synthesize the corresponding historical interactions. We devise a scoring mechanism using user interests and user embeddings to select the historical interactions for fake users.

First, we calculate the distance between the embedding of each real user u and its corresponding prototype c_i as follows:

$$d_u^{emb} = Dis(z_u, c_i), \tag{4.5}$$

where $Dis(\cdot, \cdot)$ is the distance function.

Given a set of interests $\mathcal{I}^{(u)}$ for each user u, we encode those interests by pretrained language model (PLM) as follows:

$$e_u = f_{pool}\left(f_{PLM}(\mathcal{I}^{(u)})\right), \tag{4.6}$$

where f_{pool} is the pooling function and $f_{PLM}(\cdot)$ is the pretrained language model. Based on the clusters of user embeddings, we calculate the corresponding cluster centroids of user interests $\{c_i'\}_{i=1}^{K_s}$ by averaging the user interests in each cluster. Given the interest embeddings and their centroids, we can calculate the distance between them as follows:

$$d_u^{int} = Dis(e_u, c_i'). \tag{4.7}$$

Combining Equation (4.5) and Equation (4.7), we have:

$$d_u = d_u^{emb} + \alpha \cdot d_u^{int}, \tag{4.8}$$

where α is a hyperparameter and d_u is defined as the *selection score*. Within each cluster, we employ an ascending ordering of users based on their selection scores. We

Table 4.1: Comparison of Computational Complexity.

Category	Methods	Inner	Outer	Total
Kernel-based	Distill-CF [91]	1	τ_o	τ_o
Matching-Based	DC [150]	τ_i	τ_o	$\tau_i \cdot \tau_o$
	Doscond [52]	1	τ_o	τ_o
Training-Free	TF-DCon (Ours)	1	1	1

subsequently curate the historical interactions for a synthetic user by merging the interaction histories of the top-m users, which could be formalized as follows:

$$h_s^{(u_s)} = \cup_{i=1}^{m} h^{(u_i)}, \tag{4.9}$$

where d_{u_i} ranks top-m within its corresponding cluster and $h^{(u)} \in \mathcal{H}$ is the interactions of u in the original dataset.

4.3.3 Computational Complexity Comparison.

In this section, we discuss about the time complexity of TF-DCon and previous dataset condensation methods. The complexity in Table 4.1 is calculated by counting the times of generation/training based on the whole dataset. Previous condensation methods are categorized into two types [16]: matching-based and kernel-based. The first type generates synthetic datasets via gradient matchings between the model trained on small-condensed dataset and those trained on the large-real dataset [150, 52], which typically involve a complex bi-level optimization. In the inner optimization, the models are trained while the datasets are optimized via outer gradients. In contrast, the kernel-based methods reduce the problem into a single-level optimization, which is always free from model training during condensation. If we denote the number of outer-iteration as τ_o and the number of inner-iteration as τ_i,

Table 4.2: Dataset Statistics. "avg. tok." denotes the average number of tokens in the content of each item and "avg. his." denotes the average length of user histories.

Datasets	MIND	GoodReads	MovieLens
#Items	65,238	16,833	1,682
avg. tok.	45.42	35.37	34.80
#Users	94,057	23,089	943
avg. his.	14.98	7.81	4.94
#pos	347,727	273,888	48,884
#neg	8,236,715	485,233	17,431
Density	0.0057%	0.0705%	3.0820%

we can respectively obtain the computational complexity for each and we compare our proposed method with several representative condensation approaches. As shown in Table 4.1, we can find that our proposed method is the most efficient since it follows a forward paradigm, thus free from iterative optimization on the synthesized dataset.

4.4 Experiments

In this section, we conduct experiments to evaluate TF-DCon, mainly aiming to answer the following research questions: **RQ1**: How well the performance can be achieved when trained on datasets condensed by TF-Con? **RQ2**: How generalizable is TF-DCon and the condensed datasets across different recommendation models? **RQ3**: How efficient is TF-DCon when compared to previous condensation methods? **RQ4**: Can TF-DCon work well with other less powerful LLM and how the evolutionary prompts affect the condensing quality?

Table 4.3: Overall Performance Comparison on Condensed Datasets. "Quality" denotes the achieved ratio of performance when compared to those trained on original datasets. The detailed condensation ratio could be found in Table 4.4.

Datasets		MIND[1], r=5%				Goodreads, r=27%				MovieLens, r=26%			
Rec Model	Metrics	Random	Majority	TF-DCon	*Original*	Random	Majority	TF-DCon	*Original*	Random	Majority	TF-DCon	*Original*
NAML	N@1	0.2871	0.2854	0.3071	*0.3176*	0.5197	0.5057	0.5411	*0.6462*	0.8241	0.8367	0.8484	*0.8280*
	N@5	0.3470	0.3466	0.3691	*0.3783*	0.7943	0.7884	0.8033	*0.8475*	0.8251	0.8397	0.8494	*0.8310*
	R@1	0.4016	0.4002	0.4377	*0.4534*	0.4520	0.4326	0.4704	*0.5635*	0.1785	0.1886	0.1873	*0.1752*
	R@5	0.5670	0.5697	0.6150	*0.6270*	0.9983	0.9986	0.9984	*0.9989*	0.7274	0.7323	0.7475	*0.7399*
	Quality	90.28%	90.15%	**97.22%**	*100.00%*	88.57%	87.01%	**90.49%**	*100.00%*	99.75%	102.18%	**103.15%**	*100.00%*
NRMS	N@1	0.2625	0.2631	0.2997	*0.3009*	0.5399	0.5094	0.5453	*0.6439*	0.8105	0.8294	0.8149	*0.8178*
	N@5	0.3225	0.3225	0.3597	*0.3608*	0.8017	0.7901	0.8054	*0.8476*	0.8227	0.8334	0.8371	*0.8253*
	R@1	0.3750	0.3793	0.4279	*0.4325*	0.4704	0.4344	0.4751	*0.5629*	0.1729	0.1821	0.1798	*0.1757*
	R@5	0.5414	0.5445	0.6000	*0.6042*	0.9982	0.9990	0.9985	*0.9993*	0.7325	0.7339	0.7446	*0.7321*
	Quality	88.23%	88.66%	**99.38%**	*100.00%*	90.47%	87.37%	**91.01%**	*100.00%*	99.31%	**101.57%**	101.28%	*100.00%*
Fastformer	N@1	0.2815	0.2736	0.3022	*0.3057*	0.5420	0.5165	0.5548	*0.6556*	0.7886	0.8105	0.8251	*0.7915*
	N@5	0.3425	0.3350	0.3637	*0.3645*	0.8028	0.7934	0.8092	*0.8529*	0.8254	0.8318	0.8429	*0.8145*
	R@1	0.3944	0.3804	0.4334	*0.4365*	0.4725	0.4414	0.4851	*0.5745*	0.1738	0.1799	0.1836	*0.1660*
	R@5	0.5631	0.5515	0.6096	*0.6144*	0.9983	0.9991	0.9986	*0.9990*	0.7390	0.7373	0.7465	*0.7279*
	Quality	92.01%	89.58%	**99.29%**	*100.00%*	89.74%	87.16%	**90.97%**	*100.00%*	101.80%	103.55%	**105.22%**	*100.00%*

[1] For the dataset MIND, we report $N@5, N@10$ and $R@5, R@10$, which is also the same in the subsequent tables of ablation studies.

4.4.1 Experimental Settings

Datasets

We condense the training set of three real-world datasets: MIND [120], Goodreads [103], and MovieLens [42] to conduct evaluation on our method. The statistics of these datasets are shown in Table 4.2. We split the datasets into 80%/10%/10% for training/validation/test, respectively. The condensation is mainly conducted on the training sets of datasets and during the testing phase, we still use the users and contents of items from the original datasets.

Baselines

Due to the limited studies on condensation for CBR, we compare our proposed methods with three baselines implemented by ourselves: (i) *Random*. In the setting of

overall performance comparison (Table 4.3), we randomly select a certain portion of users and randomly select a certain portion of tokens to be the contents for each item. In the setting of user-level condensation (Table 4.9), it is implemented via randomly selecting certain ratios of users while the contents of items remain the same as those in original datasets. In the setting of content-level condensation (Table 4.7), we implement it by randomly selecting tokens for each item. (ii) *Majority.* In the overall performance comparison, we implement this method by sampling the users with a large number of interactions and compressing the contents of items by randomly sampling the tokens. (iii) *TextEmbed.* For the efficiency comparison, we adapt this text-embedding-based condensation method [65], originally designed for classification, by replacing the cross-entropy loss with BPR loss. Due to its limited performance in our context, TextEmbed is included solely in the efficiency analysis and excluded in the main performance comparison. For fair comparison, we maintained an equivalent number of synthesized users as employed in our method, aiming to learn a representation for each user as well as a representation for the corresponding candidate item.

Evaluation Methods

To assess the effectiveness of our condensation approach, we evaluate the downstream recommendation performance of models trained *solely* on the condensed datasets. We consider three widely used content-based recommendation models, i.e., NAML [117], NRMS [118], and Fastformer [119]. These methods share a hierarchical architecture: a *content (item) encoder* is first applied to transform each interacted item (e.g., a news article or an item with textual attributes) in a user's history into an item representation, and a *user encoder* then aggregates the sequence of historical item representations into a unified user representation for matching and ranking candidate items. Our evaluation pipeline consists of three stages. **(1) Dataset condensation.** Given the original training set, we apply a condensation/selection method to

Table 4.4: Comparison between Condensed/Sampled datasets and Original Datasets. We use "ORI.", "RD." and "MJ." to represent the statistics of original, random sampled, and majority-selected dataset. "Item" and "User" denote the condensation information of item contents and users, respectively.

Datasets		MIND				Goodreads				MovieLens			
Methods		OR.	RD.	MJ.	Ours	OR.	RD.	MJ.	Ours	OR.	RD.	MJ.	Ours
Item	avg. tok.	45.42	16.73	16.73	16.73	35.37	16.01	16.73	16.01	34.80	15.26	16.73	15.26
	Size (KB)	10,384	4,095	4,095	4,095	2,189	993	993	993	232	121	121	121
	Ratio	100%	39%	39%	39%	100%	45%	45%	45%	100%	52%	52%	52%
User	#users	94,057	9,405	9,405	9,405	23,089	4,617	2,350	4,617	943	47	47	47
	Size (KB)	106,614	2,124	5,733	2,139	8,704	1,791	1,977	1,957	532	68	232	81
	Ratio	100%	2%	5%	2%	100%	21%	23%	22%	100%	13%	44%	15%
Overall	Size (KB)	116,998	6,219	9,828	6,234	10,893	2,784	2,970	2,950	764	189	353	202
	Ratio	100%	5%	8%	5%	100%	26%	27%	27%	100%	25%	46%	26%

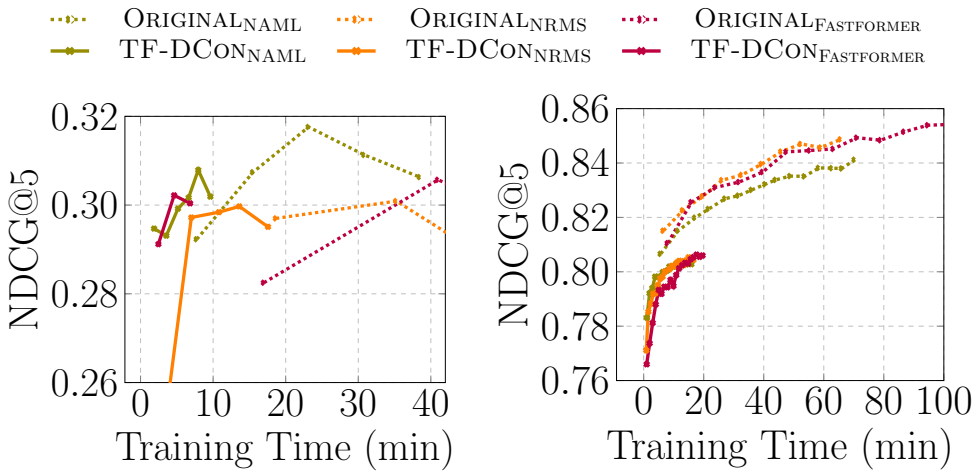
produce a compact training set under the same data budget (e.g., a fixed condensation ratio). The resulting condensed dataset is used as a drop-in replacement for the full training set. **(2) Model training on condensed data.** We train each recommendation model using only the condensed dataset, while keeping the model architecture and training recipe fixed across different condensation methods to ensure a fair comparison. Specifically, we use the same optimization settings (e.g., learning rate, batch size, and training epochs) and the same early-stopping criterion, and we tune hyper-parameters on the validation split following the standard practice of the corresponding backbone models. **(3) Performance evaluation.** After training, we evaluate the learned model on the test split with a standard top-K ranking protocol: for each user, the model scores candidate items and produces a ranked recommendation list, where the held-out test item(s) are treated as positives. We then compute ranking metrics at different cutoffs and report the averaged performance over all test

users.

For evaluation metrics, we follow the common practice in content-based recommendation [118, 117] and adopt *NDCG@K* and *Recall@K*, abbreviated as *N@K* and *R@K* in the tables. We set $K = 1, 5$ for Goodreads and MovieLens, and $K = 5, 10$ for MIND, in accordance with the typical evaluation settings of these datasets. Finally, to directly compare the effectiveness of condensed data against the original training data, we report a normalized score termed *Quality*. Concretely, for each model and each metric, we compute the percentage of the metric value achieved by training on the condensed dataset relative to training on the original dataset, and then average these percentages over the reported metrics (and cutoffs) to obtain the final *Quality* score. Higher *N@K*, *R@K*, and *Quality* indicate better recommendation performance and higher fidelity of the condensed dataset.

Implementation Details

During dataset condensation, we utilize GPT-3.5 for condensing. During training, we employ Adam [56] operator with a learning rate of 5e-3 for three datasets. We select title, abstract, and category feature of each piece of news as the item content for the MIND dataset, and select book or movie name and description feature of each book or movie as the item content for the Goodreads and MovieLens dataset. We concatenate multiple item contents into a single sequence before feeding them to the recommenders. For all three backbone models, we set the embedding dimension of content encoders to 256, the embedding dimension of user encoders to 64, and the negative sampling ratio to 4. We tune the hyperparameters of all base models to attain optimal performance. We average the results of five independent runs for each model. All experiments are conducted on a single NVIDIA GeForce RTX 3090 device.



(a) MIND (b) Goodreads

Figure 4.5: Training Efficiency on Original and Condensed Data.

4.4.2 Overall Comparison (RQ1)

To validate the effectiveness of TF-DCon, we measure the recommendation performance of CBR models trained in the condensed datasets and their training efficiency. Specifically, we train the recommendation models (i.e., NAML, NRMS, Fastformer) on the condensed datasets and evaluate their performance on the test set of the original datasets. The results are reported in Table 4.3 and the detailed condensation ratios are shown in Table 4.4. Further, we also evaluate the training efficiency on the condensed datasets and the results are summarized in Figure 4.5.

Overall Performance Comparison

The proposed TF-DCon achieves better performance than the two baselines on three datasets. We utilize "'Quality" to evaluate the quality of condensed dataset, which is computed by dividing each of the four metrics by the original performance, and then averaging them. Notably, as shown in Table 4.3, we approximate over 97% of the original performance with only 5% data on the MIND dataset across three different recommender models. From Table 4.4, we observe that the amazing con-

Table 4.5: Generalization of TF-DCon with Different Recommender Models. The datasets condensed by different recommender models exhibit similar training performance across different recommender models, verifying the versatility of TF-DCon. "Fast." denotes "Fastformer".

Train Set	Metric	NAML	NRMS	Fast.
TF-DCon$_{\text{NAML}}$	N@5	**0.3071**	0.2997	0.3022
	R@5	**0.4377**	0.4279	0.4334
TF-DCon$_{\text{NRMS}}$	N@5	0.3066	0.2905	**0.3094**
	R@5	0.4396	0.4256	**0.4409**
TF-DCon$_{\text{Fast.}}$	N@5	**0.3060**	0.2924	0.3056
	R@5	**0.4368**	0.4184	0.4343
Original	N@5	**0.3176**	0.3009	0.3057
	R@5	**0.4534**	0.4325	0.4365

densation ratio comes from two parts: reducing the size of contents by 61% (item side) and reducing the size of historical interactions and users (user side) by 98%. We observe similar performance on datasets Goodreads and MovieLens. Interestingly, when trained on condensed MovieLens, we may achieve a slightly better performance than those trained on the original large dataset. We attribute this to the small size of the original MovieLens dataset since the condensed datasets may serve as a refined and denoised alternative to the original one.

From Table 4.4, we can find that in each dataset, the size of the user part (i.e., users' historical interactions) is much larger than the size of the item part (contents of items). Consequently, the predominant size reduction stems from the condensation of the user part. We attribute this promising reduction to the common interests among different users, thus rendering that method of generating representative fake users

Table 4.6: Efficiency and Performance Comparison with Embedding-based Method. "Con. T." denotes "condense time (min)" and "TextE." denotes "TextEmbed".

Dataset	Methods	Con. T.	N@1	N@5	R@1	R@5
MIND	TextE.	450.02	0.2678	0.3147	0.3815	0.5198
	TF-DCon	**67.46**	**0.3071**	**0.3691**	**0.4377**	**0.6150**
Goodreads	TextE.	126.20	0.5013	0.7794	0.4301	0.9953
	TF-DCon	**6.15**	**0.5411**	**0.8033**	**0.4704**	**0.9984**
MovieLens	TextE.	64.58	0.7128	0.7569	0.1474	0.6884
	TF-DCon	**0.47**	**0.8484**	**0.8494**	**0.1873**	**0.7475**

to replace a variety users reasonably and effective. The condensation ratio of user-side on Goodreads and MovieLens dataset is less promising than the ratio on MIND. This discrepancy is ascribed to the fact that smaller datasets inherently exhibit fewer redundant interests among distinct users, consequently leading to a less pronounced reduction ratio. Specifically, within the MIND dataset, M users may share analogous preferences and interaction histories. In contrast, this number diminishes to N within the Goodreads and MovieLens datasets ($M >> N$). Consequently, in the condensed dataset derived from MIND, the synthesis of a single user effectively encapsulates the information equivalent to that of M users in the original dataset. In comparison, the synthesis of one user in the condensed datasets derived from Goodreads and MovieLens covers the information of N users in each dataset, respectively.

Training Efficiency

Here, we evaluate the training time and performance of NAML, NRMS and Fastformer when trained on the original datasets and condensed datasets by TF-DCon. As shown in Figure 4.5, the solid lines represent the training curves on condensed datasets and the dotted line denotes original datasets, where we can observe that the training

Table 4.7: Ablation Study on Content-level Condensation.

Datasets		MIND, $r = 39\%$			Goodreads, $r = 45\%$			MovieLens, $r = 52\%$		
Rec Model	Metrics	Random	TF-DCon	Original	Random	TF-DCon	Original	Random	TF-DCon	Original
NAML	N@1	0.3059	0.3126	*0.3176*	0.5807	0.6359	*0.6462*	0.7959	0.8382	*0.8280*
	N@5	0.3678	0.3725	*0.3783*	0.8204	0.8434	*0.8475*	0.8208	0.8367	*0.8310*
	R@1	0.4396	0.4457	*0.4534*	0.5053	0.5543	*0.5635*	0.1665	0.1811	*0.1752*
	R@5	0.6167	0.6169	*0.6270*	0.9988	0.9989	*0.9989*	0.7341	0.7374	*0.7399*
	Quality	97.39%	**98.39%**	*100%*	95.06%	**99.23%**	*100%*	97.79%	**100.75%**	*100%*
NRMS	N@1	0.2902	0.3050	*0.3009*	0.6320	0.6668	*0.6439*	0.8017	0.8207	*0.8178*
	N@5	0.3525	0.3626	*0.3608*	0.8421	0.8561	*0.8476*	0.8250	0.8285	*0.8253*
	R@1	0.4205	0.4387	*0.4325*	0.5521	0.5832	*0.5629*	0.1724	0.1759	*0.1757*
	R@5	0.5998	0.6041	*0.6042*	0.9991	0.9990	*0.9993*	0.7329	0.7354	*0.7321*
	Quality	97.23%	**99.30%**	*100%*	97.43%	**98.34%**	*100%*	99.26%	**100.38%**	*100%*
Fastformer	N@1	0.2988	0.3025	*0.3057*	0.6296	0.6499	*0.6556*	0.7828	0.7974	*0.7915*
	N@5	0.3599	0.3643	*0.3645*	0.8408	0.8506	*0.8529*	0.8086	0.8159	*0.8145*
	R@1	0.4321	0.4359	*0.4365*	0.5495	0.5701	*0.5745*	0.1653	0.1699	*0.1660*
	R@5	0.6067	0.6129	*0.6144*	0.9991	0.9991	*0.9990*	0.7223	0.7266	*0.7279*
	Quality	98.63%	**99.67%**	*100%*	97.96%	**99.60%**	*100%*	99.16%	**100.40%**	*100%*

on condensed datasets converges much faster than the training on original datasets (i.e., up to 5× speedup). Typically, the model convergences on condensed datasets are reached with around 6 minutes while the training on original datasets needs around or more than 30 minutes. For the performance, we find that those trained on condensed datasets can exhibit comparability with regard to the metric NDCG@5.

4.4.3 Generalization of TF-DCon and Condensed Datasets Across Recommender Models (RQ2)

Generalization of Condensed Datasets

We test the performance of CBR models with different architectures trained on the datasets condensed by one CBR model. Specifically, we employ NAML [117] as the backbone model for condensation and test the performance of NAML [117],

NRMS [118] and Fastformer [119] trained on the condensed datasets. The results are reported in Table 4.3. Furthermore, we also test the generalizability of content-level and user-level condensation in the ablation study for TF-DCon, as shown in Table 4.7 and Table 4.9. From the results, we can observe that the datasets condensed by TF-DCon exhibit good generalization on different CBR models. We ascribe this capability to textual data, as texts exclusively encapsulate semantic information while excluding architectural details found in text embeddings.

Generalization of TF-DCon

The proposed TF-DCon is highly flexible since we can adopt different CBR models as the user encoder in the condensation method. We investigate the performances of different models on datasets condensed by TF-DCon with different recommendation models. We utilize NAML, NRMS, and Fastformer as the user encoder for both condensation and evaluation, respectively. We choose the MIND dataset to report results in Table 4.5, where we can observe the strong generalization ability of condensed datasets on other models. We can also observe that NAML can outperform the other two models when trained on the most condensed dataset as well as the original dataset.

4.4.4 Condensation Efficiency (RQ3)

Since TF-DCon is a training-free condensation method, we examine its efficiency and compare it with previous condensation method in this section. Specifically, we compare the performance and running time of TF-DCon with TextEmbed [65], which follows the most prevalent optimization paradigm [150, 53].We report the results on three datasets with NAML as the recommendation model, as shown in Table 4.6. We can observe that TF-DCon is much faster than the iterative TextEmbed and the performance is also more superior. The efficiency of TF-DCon originates from its

Table 4.8: Generalization of TF-DCon with Open-sourced LLM and Evolutionary Prompting for Condensation.

Dataset	Metric	-evo-0	-evo-1	-evo-2	TF-DCon
MIND, r=5%	N@1	0.2902	0.2934	0.2993	**0.3071**
	N@5	0.3494	0.3526	0.3545	**0.3691**
	R@1	0.4278	0.4315	0.4329	**0.4377**
	R@5	0.5919	0.6016	0.6021	**0.6150**
Goodreads, r=27%	N@1	0.5098	0.5122	0.5258	**0.5411**
	N@5	0.7903	0.7951	0.7981	**0.8033**
	R@1	0.4419	0.4546	0.4642	**0.4704**
	R@5	0.9978	0.9979	0.9982	**0.9984**
MovieLens, r=26%	N@1	0.8330	0.8378	0.8423	**0.8480**
	N@5	0.8359	0.8410	0.8464	**0.8494**
	R@1	0.1802	0.1862	0.1857	**0.1873**
	R@5	0.7312	0.7382	0.7416	**0.7475**

training-free design, which is free from the nested optimization in TextEmbed.

4.4.5 Generalization Across Language Models and Prompt Generations (RQ4)

In this section, we aim to examine:

- The generalization of TF-DCon when equipped with other language models.
- The impact of evolutionary prompt generations on condensation performance.

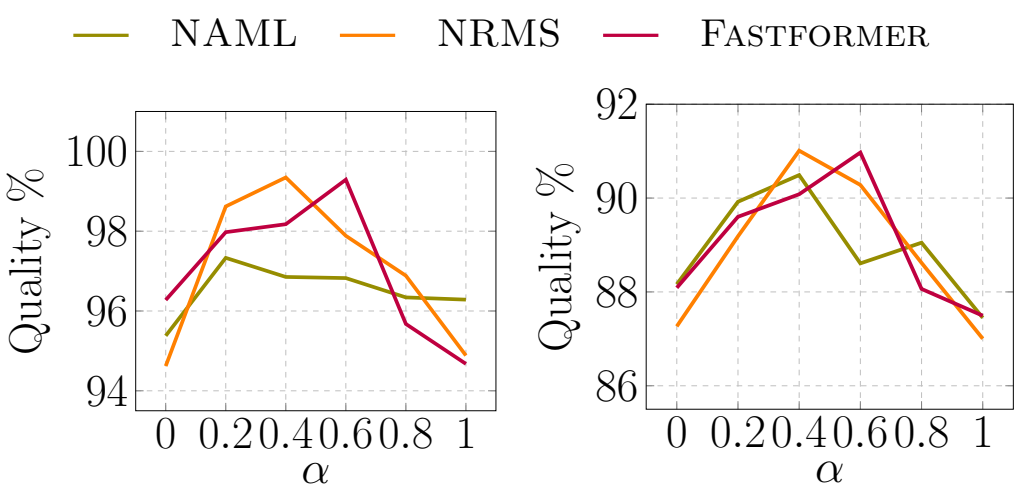
Specifically, we equip TF-DCon with Llama-2-7b [98] and utilize different generation of prompts from EvoPro (Algorithm 4) for the content-level condensation.

To examine the performance of TF-DCon with Llama-2 and the influence of Evo-Pro, we conduct experiments on three datasets with recommendation model NAML. The results are reported in Table 4.8, where "-evo-0" denotes the method with the

Table 4.9: Ablation Study on User-level Condensation.

Datasets		MIND, $r = 2\%$			Goodreads, $r = 22\%$			MovieLens, $r = 15\%$		
Rec Model	Metrics	Random	TF-DCon	Original	Random	TF-DCon	Original	Random	TF-DCon	Original
NAML	N@1	0.2886	0.2964	*0.3176*	0.5259	0.5391	*0.6462*	0.8154	0.8338	*0.8280*
	N@5	0.3474	0.3528	*0.3783*	0.7970	0.8034	*0.8475*	0.8350	0.8470	*0.8310*
	R@1	0.3995	0.4273	*0.4534*	0.4570	0.4736	*0.5635*	0.1776	0.1879	*0.1752*
	R@5	0.5627	0.5887	*0.6270*	0.9983	0.9983	*0.9989*	0.7457	0.7468	*0.7399*
	Quality	89.97%	**93.75%**	*100%*	90.91%	**92.09%**	*100%*	99.98%	**101.61%**	*100%*
NRMS	N@1	0.2702	0.2834	*0.3009*	0.5488	0.5559	*0.6439*	0.8132	0.8207	*0.8178*
	N@5	0.3281	0.3449	*0.3608*	0.8066	0.8099	*0.8476*	0.8208	0.8338	*0.8253*
	R@1	0.3796	0.4159	*0.4325*	0.4793	0.4851	*0.5629*	0.1738	0.1802	*0.1757*
	R@5	0.5413	0.5927	*0.6042*	0.9984	0.9984	*0.9993*	0.7291	0.7373	*0.7321*
	Quality	89.45%	**96.38%**	*100%*	92.78%	**93.31%**	*100%*	99.45%	**100.83%**	*100%*
Fastformer	N@1	0.2880	0.2978	*0.3057*	0.5564	0.5614	*0.6556*	0.8047	0.8251	*0.7915*
	N@5	0.3469	0.3568	*0.3645*	0.8094	0.8113	*0.8529*	0.8260	0.8412	*0.8145*
	R@1	0.3931	0.4255	*0.4365*	0.4860	0.4902	*0.5745*	0.1740	0.1815	*0.1660*
	R@5	0.5575	0.5952	*0.6144*	0.9985	0.9983	*0.9990*	0.7359	0.7460	*0.7279*
	Quality	92.12%	**97.34%**	*100%*	92.48%	**92.84%**	*100%*	101.63%	**103.76%**	*100%*

initial handcrafted prompt and "-evo-i" denotes the method with gen_i*-prompt*. **On the one hand**, we can observe that the method "TF-DCon" with ChatGPT outperforms all the variants with Llama-2, which is reasonable since the capability of text comprehension and generation of ChatGPT is superior to Llama-2-7b. However, the performance with Llama-2-7b ("-evo-i") remains close to that achieved with ChatGPT ("TF-DCon"), indicating that our method is effective even with smaller-scale language models. **On the other hand**, we can observe that the next generation of prompt can consistently outperform its "parent": "-evo-1" is better than "-evo-0" while "-evo-2" is better than "-evo-1". This demonstrates that the evolution of the prompts is beneficial in adapting LLM for condensation.



(a) MIND (b) Goodreads

Figure 4.6: Analysis on Hyperparameter α in Eq 4.8.

4.4.6 Ablation Study

To examine how **user-level condensation** and **content-level condensation** affect the overall performance, we performed an ablation study on them, respectively. We examine the performance of models trained on datasets generated by only user-level condensation or content-level condensation.

Specifically, content-level condensation only condense the contents of items and the historical interactions. The results are reported in Table 4.7, where we observe that only condensing the item contents can also achieve comparable performance. The condensation ratio is relatively not promising and we attribute this to the information capacity of the contents which restricts the size of textual data. For user-level condensation, we only condense the users into fake users and their corresponding interactions while the contents are preserved. As shown in Table 4.9, we can find that the condensation ratio is astonishingly good while we can still achieve good performance. Especially, on the MIND dataset, we can achieve over 97% performance of the original datasets with 98% size reduction of the user-side dataset (i.e., the historical

Table 4.10: Analysis on the Number of Cluster K.

Dataset	K	N@1	N@5	R@1	R@5
MIND	1154	0.2846	0.3478	0.4014	0.5686
	2308	0.2913	0.3560	0.4135	0.5997
	4617	0.3071	0.3691	0.4377	0.6150
	5000	**0.3088**	**0.3701**	**0.4397**	**0.6204**
Goodreads	1154	0.5005	0.7864	0.4344	0.9984
	2308	0.5220	0.7959	0.4541	0.9983
	4617	0.5411	**0.8033**	0.4704	0.9984
	5000	**0.5467**	0.8017	**0.4804**	**0.9986**
MovieLens	1154	0.8196	0.8213	0.1779	0.7304
	2308	0.8345	0.8399	0.1801	0.7336
	4617	**0.8484**	**0.8494**	0.1873	0.7475
	5000	0.8474	0.8486	**0.1893**	**0.7620**

interactions of users) when tested on Fastformer. When tested on NAML and NRMS, the performance can also exceed over 93% of those trained on the original dataset. Besides, the condensation ratios on Goodreads and MovieLens are also good with a comparable performance. We attribute the promising condensation ratios to similar interests among different users. Therefore, when we synthesize a fake user to represent a group of users with common interests, we can provide adequate information for model training.

Moreover, by comparing the performance of different methods in content-level condensation (Table 4.7) to the performance in user-level condensation (Table 4.9) and both-level condensation (Table 4.3), we can find that TF-DCon consistently maintains superior performance while the baseline methods suffer a large performance drop while reducing the user number. We attribute this to the ability of TF-DCon

Table 4.11: Analysis on User Interests.

Dataset	Rec Model	Metric	N@1	N@5	R@1	R@5
MIND	NAML	TF-DCon$_{RI}$	0.2978	0.3647	0.4203	0.5889
		TF-DCon	**0.3071**	**0.3691**	**0.4377**	**0.6150**
	NRMS	TF-DCon$_{RI}$	0.2916	0.3479	0.4010	0.5890
		TF-DCon	**0.2997**	**0.3597**	**0.4279**	**0.6000**
	Fastformer	TF-DCon$_{RI}$	0.2898	0.3589	0.4197	0.5938
		TF-DCon	**0.3022**	**0.3637**	**0.4334**	**0.6096**
Goodreads	NAML	TF-DCon$_{RI}$	0.5257	0.7988	0.4626	0.9982
		TF-DCon	**0.5411**	**0.8033**	**0.4704**	**0.9984**
	NRMS	TF-DCon$_{RI}$	0.5294	0.7993	0.4679	**0.9986**
		TF-DCon	**0.5453**	**0.8054**	**0.4751**	0.9985
	Fastformer	TF-DCon$_{RI}$	0.5496	0.8055	0.4602	**0.9989**
		TF-DCon	**0.5548**	**0.8092**	**0.4851**	0.9986
MovieLens	NAML	TF-DCon$_{RI}$	0.8361	0.8402	0.1813	0.7432
		TF-DCon	**0.8484**	**0.8494**	**0.1873**	**0.7475**
	NRMS	TF-DCon$_{RI}$	0.8126	0.8344	0.1764	0.7340
		TF-DCon	**0.8149**	**0.8371**	**0.1798**	**0.7446**
	Fastformer	TF-DCon$_{RI}$	0.8085	0.8352	0.1821	0.7437
		TF-DCon	**0.8251**	**0.8429**	**0.1836**	**0.7465**

in preserving the preference information for most users.

4.4.7 Further Investigation

Analysis on hyperparameter α

In this section, we further investigate how different values of hyperparameter α affect the quality of the condensed datasets.

The hyperparameter α is utilized to balance the influence of user interests and user embeddings on the user synthesis. The larger value of α denotes larger influence of user interests on the synthesis of fake users. As shown in Figure 4.6, we can find that in most cases, the best choice for α with three different recommendation models is smaller than 1 but larger than 0, which verifies that the effectiveness of selecting/merging users with similar interests for the synthesis. Besides, the selection of α varies among three recommendation models, influenced by their respective interest-capturing ability. A smaller optimal α indicates the model's diminished capacity to capture information from limited texts, such as user interests. We can observe that Fastformer possesses the best interest-capturing ability, which is also in line with the results in previous study [119].

Analysis on the number of Cluster K

This section investigates the influence of the cluster parameter K, which directly corresponds to the cardinality of synthesized users in the condensed dataset. To evaluate how dataset condensation quality scales with K, experiments were conducted on three benchmark datasets (MIND, Goodreads, MovieLens) using the NAML recommendation model. Key results in Table 4.10 reveal a general performance improvement across all datasets as K increases from 1,154 to 5,000, suggesting that larger condensed datasets better preserve latent user-item interaction patterns.

Notably, the performance gains exhibit dataset-dependent characteristics: MIND demonstrates progressive enhancement in all metrics (N@1: +8.5%, R@5: +9.1%), aligning with its inherent user diversity. Goodreads achieves near-perfect R@5 (0.9986) even with smaller K, reflecting its dense interaction patterns, while N@1 improves significantly (+9.2%) with larger K. MovieLens shows diminishing returns in N@5 (0.8486 at K= 5000 K=5000 vs 0.8494 at K = 4617 K=4617), suggesting an optimal cluster-density threshold for precision-oriented metrics.

Table 4.12: Moderation Check.

Dataset	MIND		Goodreads		MovieLens	
Violation	Mean	Var	Mean	Var	Mean	Var
sexual	4.94E-03	2.13E-03	8.45E-04	1.43E-05	3.08E-04	1.13E-06
hate	6.72E-05	3.97E-08	3.33E-05	1.90E-08	2.11E-05	3.75E-09
harassment	1.57E-04	2.64E-07	4.55E-05	6.64E-09	4.64E-05	4.29E-08
self-harm	2.46E-06	8.65E-11	3.04E-06	1.77E-10	2.71E-06	8.68E-11
minors	1.05E-05	4.26E-09	1.55E-05	1.09E-08	2.68E-06	5.28E-11
hate	1.18E-06	2.12E-11	1.01E-06	1.70E-11	1.11E-06	2.10E-11
graphic	4.77E-04	4.31E-06	7.49E-05	9.46E-08	5.03E-05	3.11E-08
suicide	2.43E-07	1.43E-12	7.80E-08	1.18E-13	1.38E-07	3.61E-13
instructions	7.96E-07	6.16E-12	3.07E-08	3.02E-14	9.53E-08	2.98E-13
threatening	2.13E-05	1.36E-08	1.75E-06	2.60E-11	4.18E-06	4.96E-10
violence	3.40E-03	2.57E-04	4.79E-03	3.50E-04	3.42E-03	1.10E-04

Analysis on User Interests

In this section, to examine the effectiveness of the user interests in assisting the user condensation, conduct an ablation study on the MovieLens dataset by designing a variant of TF-DCon, denoted by "TF-DCon$_{RI}$". We implement "TF-DCon$_{RI}$" by replacing the user interests with tokens randomly sampled from his/her historical items and restricting the number of the tokens to the same number of user interests. The results are shown in Table 4.11. We can observe that randomly sampled tokens under-perform TF-DCon, which demonstrates the effectiveness of our design of ChatGPT-enhanced interests extraction module.

Moderation Check

To check whether the condensed dataset contain toxic contents, we conduct moderation check via the endpoints provided by OpenAI[1]. Specifically, we randomly sample 100 items from each dataset to assess their contents. The results are presented in

[1]https://platform.openai.com/docs/guides/moderation

table 4.12, where the "Mean" signifies the mean violation score of the contents in items encompassing the corresponding toxic words and "Var" denotes the associated variance. The violation score represents the model's confidence in input violations of the OpenAI category policy. This value ranges between 0 and 1, with higher values indicating higher confidence that the input contains the corresponding toxic words. Therefore, lower values indicate better quality. We can observe from the table that the scores on the condensed datasets are extremely low, indicating that they do not incorporate any of the toxic contents.

4.4.8 Case Study

In this section, we conduct case study to examine the reasonability of the condensed data by investing the correlation between the history of the synthesized users and their corresponding real users. The following example is randomly sampled from the MovieLens dataset. We show the history of the synthesized user and three related real users. In the following, the history of synthesized user consists of short introduction of the interacted movies while the histories of real users consist of the title and description of the interacted movies. We can observe that the preference information of those real users can be roughly covered by this synthesized user, which may be the reason why models trained on the condensed data since it provides adequate information for sufficient model training.

Synthesized User

Interaction History:

- My Best Friend's Wedding is about a romantic comedy. //**from: Real User 1**
- Back to the Future is about a high school student who time travels to the past and must ensure his parents fall in love to preserve his own existence. //**from: Real User 1**

- Grosse Pointe Blank is about a hitman facing unexpected dilemmas at a high school reunion. //**from: Real User 3**
- "Robin Hood: Men in Tights" is about a skilled archer leading a funny rebellion against a corrupt ruler, with the help of his misfit Merry Men. //**from:Real User 1 & Real User 2 & Real User 3**
- Mad City is about a hostage situation in a museum, delving into the power struggle between a security guard and a journalist, while examining media ethics and the blurred boundaries of truth. //**from:Real User 2**
- The Wizard of Oz is about Dorothy's journey in the magical land of Oz, where she seeks the help of the Great Wizard to return home, facing challenges and learning valuable lessons along the way. //**from:Real User 2**
- This Is Spinal Tap is a mockumentary comedy released in 1984. //**from:Real User 2**

Real User 1

Interaction History:

- My Best Friend's Wedding (1997), with description:
  - "My Best Friend's Wedding is a romantic comedy film released in 1997."
- Snow White and the Seven Dwarfs (1937), with description:
  - "Snow White and the Seven Dwarfs is a timeless animated film released in 1937. It tells the story of a young princess named Snow White who finds herself in the care of seven lovable dwarfs after escaping from her evil stepmother. With its enchanting tale, memorable songs, and captivating animation, this classic movie continues to captivate audiences of all ages."
- Die Hard (1988), with description:
  - "Die Hard is an action film released in 1988."
- Robin Hood: Men in Tights (1993), with description:
  - "In Robin Hood: Men in Tights (1993), a comedic retelling of the classic Robin

Hood tale, a skilled archer named Robin must outwit the corrupt Prince John and win the heart of the beautiful Maid Marian. With a band of misfit Merry Men by his side, Robin leads a hilarious and daring rebellion against the unjust ruler, using wit, skill, and plenty of laughs along the way."

- Back to the Future (1985), with description:
  - "Back to the Future is a science fiction film released in 1985. The movie follows the story of a high school student named Marty McFly who, with the help of a eccentric scientist, travels back in time to the year 1955. Marty must find a way to return to the present and also ensure that his parents meet and fall in love, or else he risks erasing his own existence. The film combines comedy, adventure, and a touch of romance as it explores the consequences of altering the past and the importance of family."

Real User 2

Interaction History:

- Snow White and the Seven Dwarfs (1937), with description:
  - "Snow White and the Seven Dwarfs is a timeless animated film released in 1937. It tells the story of a young princess named Snow White who finds herself in the care of seven lovable dwarfs after escaping from her evil stepmother. With its enchanting tale, memorable songs, and captivating animation, this classic movie continues to captivate audiences of all ages."
- Robin Hood: Men in Tights (1993), with description:
  - "Robin Hood: Men in Tights (1993), a comedic retelling of the classic Robin Hood tale, a skilled archer named Robin must outwit the corrupt Prince John and win the heart of the beautiful Maid Marian. With a band of misfit Merry Men by his side, Robin leads a hilarious and daring rebellion against the unjust ruler, using wit, skill, and plenty of laughs along the way."

- The Wizard of Oz (1939), with description:
  - "The Wizard of Oz is a classic fantasy film released in 1939. It tells the story of a young girl named Dorothy, who is transported to the magical land of Oz after a tornado hits her Kansas farm. In Oz, Dorothy encounters various colorful characters, including the Scarecrow, the Tin Man, and the Cowardly Lion, as she embarks on a journey to find the Great Wizard who can help her return home. Along the way, they face challenges from the Wicked Witch of the West, and learn important lessons about friendship, courage, and the importance of following your dreams. The Wizard of Oz is beloved for its stunning visuals, memorable songs, and timeless themes."
- Mad City (1997), with description:
  - "Mad City, a 1997 film, centers on a hostage crisis in a museum, delving into the power struggle between a disgruntled security guard and a manipulative journalist, each vying to control the crisis narrative. Through intense dialogue and suspense, Mad City critically examines media ethics and the blurred boundaries between truth and fiction."
- This Is Spinal Tap (1984), with description:
  - "This Is Spinal Tap is a comedy mockumentary film released in 1984."

Real User 3

Interaction History:

- Robin Hood: Men in Tights (1993), with description:
  - "Robin Hood: Men in Tights" (1993), a comedic retelling of the classic Robin Hood tale, a skilled archer named Robin must outwit the corrupt Prince John and win the heart of the beautiful Maid Marian. With a band of misfit Merry Men by his side, Robin leads a hilarious and daring rebellion against the unjust ruler, using wit, skill, and plenty of laughs along the way."

- Chasing Amy (1997), with description:
  - "*Chasing Amy* is a 1997 romantic comedy-drama film."
- In the Name of the Father (1993), with description:
  - "In the Name of the Father is a gripping drama film released in 1993. It tells the story of Gerry Conlon, a young Irishman wrongfully convicted of an IRA bombing. Set in the backdrop of the Troubles in Northern Ireland, the film explores themes of injustice, incarceration, and the quest for truth and justice. Directed by Jim Sheridan and featuring remarkable performances by Daniel Day-Lewis and Pete Postlethwaite, this emotional and thought-provoking movie delves into the complexities of the legal system and the enduring power of family bonds."
- Nightmare on Elm Street, A (1984), with description:
  - "A Nightmare on Elm Street is a horror film released in 1984. Directed by Wes Craven, the movie tells the story of a group of teenagers who are haunted in their dreams by a supernatural killer named Freddy Krueger. As they struggle to stay awake and escape his clutches, they soon realize that death in their dreams also means death in reality. Filled with suspenseful moments and unnerving visual effects, A Nightmare on Elm Street delivers a chilling and unforgettable experience for horror enthusiasts."
- Grosse Pointe Blank (1997), with description:
  - "Grosse Pointe Blank is a 1997 movie that revolves around a skilled and professional hitman who, during a high school reunion, unexpectedly finds himself confronting personal and moral dilemmas."

4.4.9 Implications

The results show that that our proposed method is effective in condense content-based recommendation datasets, achieving comparable model training performance with even 5% data. Our method designs a novel condensing paradigm, training-free,

from previous iterative-optimization based ones. Benefiting from this design, our method is significantly more efficient. Besides, we enable the generation of discrete texts via ChatGPT and preserve users' preference via user-level condensation module. Therefore, incorporating our method to condense datasets for daily model training can save lots of resources costs without performance degradation in practice.

4.5 Limitations

Although TF-DCon has been introduced to address the issue of resources demanding during model training by condensing datasets, it introduces new limitations: (1) Although the condensation process is efficient, it requires a pre-process of the dataset by ChatGPT/other LLMs, the efficiency of which can be unstable when we utilize ChatGPT for the pre-process since it is affected by the network latency. (2) Since the cost of calling ChatGPT is linear to the size of dataset, we may need to train the models based on the condensed datasets for a certain times to achieve the cost-gain balance. (3) As shown in the experiments, the performance of TF-DCon with open source LLMs is still limited.

4.6 Related Work

4.6.1 Content-Based Recommendation

Content-based recommendation (CBR) encompasses various formats of information in different formats: text [66, 127, 73], image [108, 144, 21, 26], video [113, 135], social networks [35, 72], etc. This paper focuses on textual contents, i.e., news, books, and movie reviews.

Textual CBR has gained significant attention in both industry and academia. Models

based on neural networks [106, 118] have been proposed to learn the representations of users and items via CNN [59] or attention mechanism [102]. Thereafter, the advance of pretrained language models like BERT [20] exhibited promising performance on textual content recommendation [63, 146]. However, these methods require training on large-scale datasets, resulting in high computational costs and challenges for sustainable and scalable applications. The recent emergence of investigations into dataset condensation and dataset distillation presents a promising avenue for tackling this concern [110, 150]. Dataset condensation synthesizes compact datasets for training, enabling models to perform comparably to those trained on full datasets. Nevertheless, dataset condensation has not been explored in the context of CBR. To bridge this critical gap, we introduce an innovative and efficacious framework tailored to the task of dataset condensation for CBR.

4.6.2 Dataset Condensation

As we all know, training neural networks on large datasets is extremely expensive. To mitigate this issue, plenty of works on dataset distillation/dataset condensation have been proposed to synthesize small datasets to replace the large ones, such as GCond [53], DC [150] and SFGC [155]. Those methods always formulate the condensation process as a bi-level optimization paradigm. The condensation methods have been explored on images [150], graph [53] and text embeddings [40]. Various methods have been proposed to improve the quality of condensation and optimization efficiency. For instance, DC [150] proposed gradient matching, achieving a comparable performance under a high compression ratio, and DosCond [52] proposed one-step update strategy, which significantly improves the optimization efficiency. Condensation on textual data is conducted based on the text embeddings or soft labels [82, 65, 93, 95, 40, 105, 123, 125, 124]. Further exploration extends condensation methods to image-text retrival [132]. Recently, a work [91] aims at condensing collaborative filtering dataset but it is only suitable for NTK-based (Neural Tangent

Kernel) architecture and fails to process textual data. Besides, CGM [104] follows the nested bi-level formulation to condense categorical data for CTR predictions and DConRec [122] designs an optimization-based method for ID-based recommendation, both of which cannot generate texts as well.

To fill the gap, we design a condensation method for the CBR dataset. Unlike previous methods, our approach does not require iterative training, significantly reducing condensation costs. Furthermore, our condensation procedure centers on textual content, thereby diverging from those text condensation methods that hinge on text embeddings.

4.7 Conclusion

Advanced content-based recommendation models rely on training with large datasets, which is costly. To solve this problem, in this paper, we investigate how to condense the dataset of content-based recommendation into a small synthetic dataset, enabling models to achieve performance comparable to those trained on full datasets. We propose a novel method TF-DCon where a prompt-evolving module is proposed to adapt ChatGPT to condense contents of items, and users and their corresponding historical interactions are condensed via a curated clustering-based synthesis module. This work represents the first exploration of dataset condensation for content-based recommendation and the first non-iterative synthetic data optimization approach. The experimental results on multiple real-world datasets verify the effectiveness of our proposed methods. Notably, TF-DCon achieves up to 97% of the original performance while reducing the dataset size by 95% (i.e., on the MIND dataset). The condensation process of TF-DCon is significantly faster than the method that is based on nested optimization.

Chapter 5

Conclusion and Future Directions

5.1 Conclusion

This dissertation advances the field of recommender systems by shifting the focus from traditional model-centric approaches to a data-centric paradigm. Motivated by the limitations of existing methods in capturing multi-faceted user dynamics and addressing computational inefficiencies, we have proposed novel techniques to optimize datasets and enhance data utilization for training recommendation models. The key contributions are synthesized as follows:

- Disentangled Contrastive Learning for Social Recommendation (Chapter 2): Building on social correlation theory, we introduced a framework to decouple user behaviors into collaborative (user-item) and social (user-user) domains. By employing a cross-domain contrastive learning module, our method enables the independent modeling of domain-specific personality facets while facilitating recommendation modeling via knowledge transfer between domains. This approach addresses the longstanding challenge of capturing heterogeneous user traits in social recommendation systems, achieving significant improvements in

recommendation accuracy.

- Dataset Condensation for Recommendation Data (Chapter 3): To mitigate the computational burden of training on large-scale interaction sets, we formulated the problem of dataset condensation for recommendation tasks and developed DConRec, a bilevel optimization framework. DConRec synthesizes compact yet representative interaction subsets that preserve critical user-item relationships, enabling models trained on condensed data to match the performance of those trained on full datasets. Theoretical convergence guarantees further validate its effectiveness.

- Training-free Condensation for Content-based Recommendation (Chapter 4): Extending condensation principles to content-aware scenarios, we proposed TF-DCon, which integrates large language models (LLMs) to process textual item descriptions and employs training-free modules for simultaneous condensation of user interactions and item content. By eliminating gradient-based optimization, TF-DCon achieves computationally efficient condensation while maintaining recommendation quality, bridging the gap between ID-based and content-based recommendation paradigms.

These contributions collectively demonstrate that data-centric optimization—spanning data synthesis, compression, and cross-domain modeling—can substantially enhance both the performance and efficiency of modern recommender systems.

5.2 Future Directions

While this dissertation has introduced significant advancements in data optimization for recommendation systems, several directions remain open for future research:

- Improving Data Condensation for Dynamic Data: One limitation of current

data condensation techniques is their reliance on static datasets. In dynamic environments, where user preferences and content evolve over time, future work could focus on adapting condensation methods to handle real-time data updates and changes.

- Incorporating More Complex Side Information: While this work addresses user-item interactions and item content, future research could incorporate additional types of side information, such as temporal, geographical, or contextual factors, into the condensation process. This would further enhance the ability of recommendation systems to provide personalized and context-aware recommendations.

- Exploring Other Forms of Knowledge Transfer: The cross-domain contrastive learning module introduced in this dissertation offers a promising approach for knowledge transfer between domains. Future research could explore additional forms of knowledge transfer, such as domain adaptation or transfer learning, to further improve recommendation systems involving data from various domains.

- Scalability and Robustness: As recommendation systems continue to scale, the challenge of maintaining computational efficiency and robustness becomes more pronounced. Further studies could investigate how to scale the proposed techniques to handle extremely large datasets while ensuring the stability and robustness of the recommendation models.

Beyond methodological advances, the proposed data-centric techniques can be directly integrated into practical recommendation pipelines. DcRec is well-suited for platforms where users exhibit heterogeneous behaviors across collaborative and social domains, enabling more faithful personalization via domain-aware modeling and cross-domain knowledge transfer. DConRec and TF-DCon can serve as a lightweight *data optimization layer* that produces compact, informative training sets, making large-scale recommender training and re-training substantially cheaper for frequent

A/B testing, periodic model refresh, and deployment under limited compute budgets. In particular, TF-DCon is applicable to content-rich services (e.g., news or review platforms) by condensing verbose item texts into informative summaries and synthesizing representative user histories, thereby reducing both content-processing and interaction-training costs while remaining compatible with different downstream CBR architectures. Overall, these methods support scalable, sustainable recommendation development by improving training efficiency without requiring a complete redesign of model architectures.

In conclusion, this dissertation has demonstrated the potential of data-centric techniques to improve recommendation systems in both performance and efficiency. By focusing on optimizing the data itself—rather than solely developing new model architectures—we have proposed innovative approaches to enhancing user personalization and reducing the computational demands of training recommendation models. The promising results presented here pave the way for future advancements in the field and provide a solid foundation for the continued exploration of data-centric AI in recommendation systems.

References

[1] Fares Antaki, Samir Touma, Daniel Milad, Jonathan El-Khoury, and Renaud Duval. Evaluating the performance of chatgpt in ophthalmology: An analysis of its successes and shortcomings. In *medRxiv*, 2023.

[2] James RA Benoit. Chatgpt for clinical vignette generation, revision, and evaluation. In *medRxiv*, 2023.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901. Curran Associates, Inc., 2020.

[4] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Generalizing dataset distillation via deep generative prior. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3739–3748, 2023.

[5] Jiajia Chen, Xin Xin, Xianfeng Liang, Xiangnan He, and Jun Liu. Gdsrec: Graph-based decentralized collaborative filtering for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[6] Jingfan Chen, Wenqi Fan, Guanghui Zhu, Xiangyu Zhao, Chunfeng Yuan, Qing Li, and Yihua Huang. Knowledge-enhanced black-box attacks for recommendations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 108–117, 2022.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[8] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. In *Proceedings of the 33th Advances in Neural Information Processing Systems*, pages 22243–22255, Virtual Event, 2020. Curran Associates, Inc.

[9] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. Exploring the potential of large language models (llms) in learning on graphs. *SIGKDD Explor. Newsl.*, 2024.

[10] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2011.

[11] Jonathan H Choi, Kristin E Hickman, Amy Monahan, and Daniel Schwarcz. Chatgpt goes to law school. In *Available at SSRN (2023)*, 2023.

[12] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Se-

bastian Gehrmann, Parker Schuh, Kensen Shi, Sashank Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: scaling language modeling with pathways. *J. Mach. Learn. Res.*, 2023.

[13] Robert B Cialdini and Noah J Goldstein. Social influence: Compliance and conformity. *Annual Review of Psychology*, pages 591–621, 2004.

[14] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020.

[15] Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. Chatlaw: Open-source legal large language model with integrated external knowledge bases., 2023.

[16] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. In *Proceedings of the 40th International Conference on Machine Learning*. JMLR.org, 2023.

[17] Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Fang Zeng, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. AugGPT: Leveraging ChatGPT for Text Data Augmentation . *IEEE Transactions on Big Data*, pages 1–12, 2023.

[18] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, page 1126–1132. Association for Computing Machinery, 2023.

[19] Tyler Derr, Yao Ma, Wenqi Fan, Xiaorui Liu, Charu Aggarwal, and Jiliang Tang. Epidemic graph convolutional network. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*, pages 160–168. Association for Computing Machinery, 2020.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019.

[21] Yujuan Ding, Yunshan Ma, Wenqi Fan, Yige Yao, Tat-Seng Chua, and Qing Li. Fashionregen: Llm-empowered fashion report generation. In *Companion Proceedings of the ACM Web Conference 2024*. Association for Computing Machinery, 2024.

[22] Jiajun Fan and Changnan Xiao. Generalized data distribution iteration. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

[23] Jiajun Fan, Yuzheng Zhuang, Yuecheng Liu, Jianye HAO, Bin Wang, Jiangcheng Zhu, Hao Wang, and Shu-Tao Xia. Learnable behavior control: Breaking atari human world records via sample-efficient behavior selection. In *The Eleventh International Conference on Learning Representations*, 2023.

[24] Wenqi Fan, Tyler Derr, Yao Ma, Jianping Wang, Jiliang Tang, and Qing Li. Deep adversarial social recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019.

[25] Wenqi Fan, Tyler Derr, Xiangyu Zhao, Yao Ma, Hui Liu, Jianping Wang, Jiliang Tang, and Qing Li. Attacking black-box recommendations via copying cross-domain user profiles. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1583–1594. IEEE, 2021.

[26] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2024.

[27] Wenqi Fan, Wei Jin, Xiaorui Liu, Han Xu, Xianfeng Tang, Suhang Wang, Qing Li, Jiliang Tang, Jianping Wang, and Charu Aggarwal. Jointly attacking graph neural network and its explanations. *arXiv*, 2021.

[28] Wenqi Fan, Qing Li, and Min Cheng. Deep modeling of social relations for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2018.

[29] Wenqi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li. Graph trend filtering networks for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 112–121. Association for Computing Machinery, 2022.

[30] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference*. Association for Computing Machinery, 2019.

[31] Wenqi Fan, Yao Ma, Qing Li, Jianping Wang, Guoyong Cai, Jiliang Tang, and Dawei Yin. A graph neural network framework for social recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[32] Wenqi Fan, Yao Ma, Dawei Yin, Jianping Wang, Jiliang Tang, and Qing Li. Deep social collaborative filtering. In *Proceedings of the 13th ACM Conference on Recommender Systems*, page 305–313. Association for Computing Machinery, 2019.

[33] Junfeng Fang, Xinglin Li, and et al. EXGC: bridging efficiency and explainability in graph condensation. In *Proceedings of the ACM Web Conference 2024*. Association for Computing Machinery, 2024.

[34] Junfeng Fang, Wei Liu, Yuan Gao, Zemin Liu, An Zhang, Xiang Wang, and Xiangnan He. Evaluating post-hoc explanations for graph neural networks via robustness analysis. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[35] Mingxin Gan, Chunhua Wang, Lingling Yi, and Hao Gu. Exploiting dynamic social feedback for session-based recommendation. *Information Processing & Management*, 2024.

[36] Xinyi Gao, Tong Chen, Wentao Zhang, Junliang Yu, Guanhua Ye, Quoc Viet Hung Nguyen, and Hongzhi Yin. Rethinking and accelerating graph condensation: A training-free approach with class partition. *arXiv*, 2024.

[37] Xinyi Gao, Junliang Yu, Wei Jiang, Tong Chen, Wentao Zhang, and Hongzhi Yin. Graph condensation: A survey. *arXiv*, 2024.

[38] Shengbo Gong, Juntong Ni, Noveen Sachdeva, Carl Yang, and Wei Jin. Gc-bench: A benchmark framework for graph condensation with new insights. *arXiv*, 2024.

[39] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. Trustsvd: collaborative ciltering with both the explicit and implicit influence of user trust and of item ratings. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. AAAI Press, 2015.

[40] Xudong Han, Aili Shen, Yitong Li, Lea Frermann, Timothy Baldwin, and Trevor Cohn. Towards fair dataset distillation for text classification. In *Proceedings of The Third Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 65–72. Association for Computational Linguistics, 2022.

[41] Bowen Hao, Jing Zhang, Hongzhi Yin, Cuiping Li, and Hong Chen. Pre-training graph neural networks for cold-start users and items representation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, 2021.

[42] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 2015.

[43] Mohammad Hashemi, Shengbo Gong, Juntong Ni, Wenqi Fan, B. Aditya Prakash, and Wei Jin. A comprehensive survey on graph reduction: Spar-

sification, coarsening, and condensation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

[44] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[45] Rui He, Shengcai Liu, Shan He, and Ke Tang. Multi-domain active learning: Literature review and comparative study. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2023.

[46] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 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*. Association for Computing Machinery, 2020.

[47] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43th international ACM SIGIR conference on Research and development in Information Retrieval*. Association for Computing Machinery, 2020.

[48] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *The World Wide Web Conference*. Association for Computing Machinery, 2017.

[49] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *The World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 2017.

[50] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth*

ACM Conference on Recommender Systems. Association for Computing Machinery, 2010.

[51] Wei Jin, Tyler Derr, Haochen Liu, Yiqi Wang, Suhang Wang, Zitao Liu, and Jiliang Tang. Self-supervised learning on graphs: Deep insights and new direction. *arXiv*, 2020.

[52] Wei Jin, Xianfeng Tang, Haoming Jiang, Zheng Li, Danqing Zhang, Jiliang Tang, and Bing Yin. Condensing graphs via one-step gradient matching. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 720–730. Association for Computing Machinery, 2022.

[53] Wei Jin, Lingxiao Zhao, Shichang Zhang, Yozen Liu, Jiliang Tang, and Neil Shah. Graph condensation for graph neural networks. In *International Conference on Learning Representations*, 2022.

[54] KrishnaTeja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh K. Iyer. RETRIEVE: coreset selection for efficient and robust semi-supervised learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2021.

[55] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

[56] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[57] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[58] Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, pages 30–37, 2009.

[59] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012.

[60] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

[61] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. Disentangled contrastive learning on graphs. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2021.

[62] Hui Li, Dingming Wu, Wenbin Tang, and Nikos Mamoulis. Overlapping community regularization for rating prediction in social recommender systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*. Association for Computing Machinery, 2015.

[63] Jian Li, Jieming Zhu, Qiwei Bi, Guohao Cai, Lifeng Shang, Zhenhua Dong, Xin Jiang, and Qun Liu. MINER: Multi-interest matching network for news recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 343–352. Association for Computational Linguistics, 2022.

[64] Jiatong Li, Yunqing Liu, Wenqi Fan, Xiao-Yong Wei, Hui Liu, Jiliang Tang, and Qing Li. Empowering molecule discovery for molecule-caption translation with large language models: A chatgpt perspective. *IEEE Transactions on Knowledge & Data Engineering*, 2024.

[65] Yongqi Li and Wenjie Li. Data distillation for text classification. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in Information Retrieval*. Association for Computing Machinery, 2021.

[66] Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. Generative retrieval for conversational question answering. *Information Processing & Management*, 2023.

[67] Shuai Lin, Pan Zhou, Zi-Yuan Hu, Shuojia Wang, Ruihui Zhao, Yefeng Zheng, Liang Lin, Eric Xing, and Xiaodan Liang. Prototypical graph contrastive learning. *arXiv*, 2021.

[68] Haifeng Liu, Hongfei Lin, Wenqi Fan, Yuqi Ren, Bo Xu, Xiaokun Zhang, Dongzhen Wen, Nan Zhao, Yuan Lin, and Liang Yang. Self-supervised learning for fair recommender systems. *Applied Soft Computing*, page 109126, 2022.

[69] Haifeng Liu, Hongfei Lin, Bo Xu, Liang Yang, Yuan Lin, Yonghe Chu, Wenqi Fan, and Nan Zhao. Improving social recommendations with item relationships. In *International Conference on Neural Information Processing*, pages 763–770. Springer, 2020.

[70] Haochen Liu, Yiqi Wang, Wenqi Fan, Xiaorui Liu, Yaxin Li, Shaili Jain, Yunhao Liu, Anil K Jain, and Jiliang Tang. Trustworthy ai: A computational perspective. *arXiv*, 2021.

[71] Mengyang Liu, Shanchuan Li, Xinshi Chen, and Le Song. Graph condensation via receptive field distribution matching. *arXiv*, 2022.

[72] Peipei Liu, Gaosheng Wang, Hong Li, Jie Liu, Yimo Ren, Hongsong Zhu, and Limin Sun. Multi-granularity cross-modal representation learning for named entity recognition on social media. *Information Processing & Management*, 2024.

[73] Qijiong Liu, Hengchang Hu, Jiahao Wu, Jieming Zhu, Min-Yen Kan, and Xiao-Ming Wu. Discrete semantic tokenization for deep ctr prediction. In *Companion Proceedings of the ACM Web Conference 2024*. Association for Computing Machinery, 2024.

[74] Shengcai Liu, Ning Lu, Cheng Chen, and Ke Tang. Efficient combinatorial optimization for word-level adversarial textual attack. *IEEE ACM Trans. Audio Speech Lang. Process.*, 2022.

[75] Shengcai Liu, Ning Lu, Wenjing Hong, Chao Qian, and Ke Tang. Effective and imperceptible adversarial textual attack via multi-objectivization. *ACM Trans. Evol. Learn. Optim.*, 2024.

[76] Shengcai Liu, Zhiyuan Wang, Yew-Soon Ong, Xin Yao, and Ke Tang. Learning mixture-of-experts for general-purpose black-box discrete optimization. *arXiv*, 2024.

[77] Yugeng Liu, Zheng Li, Michael Backes, Yun Shen, and Yang Zhang. Backdoor attacks against dataset distillation. In *30th Annual Network and Distributed System Security Symposium*, 2023.

[78] Ning Lu, Shengcai Liu, Rui He, Yew-Soon Ong, Qi Wang, and Ke Tang. Large language models can be guided to evade ai-generated text detection. *Transactions on Machine Learning Research*, 2024.

[79] Ning Lu, Shengcai Liu, Zhirui Zhang, Qi Wang, Haifeng Liu, and Ke Tang. Less is more: Understanding word-level textual adversarial attack via n-gram frequency descend. *arXiv*, 2023.

[80] Hao Ma, Michael R. Lyu, and Irwin King. Learning to recommend with trust and distrust relationships. In *Proceedings of the 3rd ACM Conference on Recommender Systems*. Association for Computing Machinery, 2009.

[81] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, 2011.

[82] Aru Maekawa, Naoki Kobayashi, Kotaro Funakoshi, and Manabu Okumura. Dataset distillation with attention labels for fine-tuning BERT. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 119–127. Association for Computational Linguistics, 2023.

[83] Peter V Marsden and Noah E Friedkin. Network studies of social influence. *Sociological Methods & Research*, pages 127–151, 1993.

[84] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, pages 415–444, 2001.

[85] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2021.

[86] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[87] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *Proceedings of the 33th International Conference on Machine Learning*. JMLR.org, 2016.

[88] Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. Disentangled contrastive collaborative filtering. In *Proceedings of the 46nd international ACM SIGIR conference on Research and development in Information Retrieval*. Association for Computing Machinery, 2023.

[89] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE Computer Society, 2010.

[90] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, page 452–461, 2009.

[91] Noveen Sachdeva, Mehak Preet Dhaliwal, Carole-Jean Wu, and Julian McAuley. Infinite recommendation networks: A data-centric approach. In *Advances in Neural Information Processing Systems*, 2022.

[92] Noveen Sachdeva, Carole-Jean Wu, and Julian McAuley. On sampling collaborative filtering datasets. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, 2022.

[93] Shivam Sahni and Harsh Patel. Exploring multilingual text data distillation, 2023.

[94] Rendle Steffen, Freudenthaler Christoph, Gantner Zeno, and Schmidt-Thieme Lars. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009.

[95] Ilia Sucholutsky and Matthias Schonlau. Soft-label dataset distillation and text dataset distillation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021.

[96] Jiliang Tang, Huiji Gao, Huan Liu, and Atish Das Sarma. etrust: understanding trust evolution in an online world. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2012.

[97] Jiliang Tang, Huiji Gao, Huan Liu, and Atish Das Sarma. etrust: understanding trust evolution in an online world. In *Proceedings of the 18th ACM SIGKDD*

Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, 2012.

[98] Hugo Touvron, Louis Martin, and et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv*, 2023.

[99] Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. Zeroshotdataaug: Generating and augmenting training data with chatgpt. *arXiv:2304.14334*, 2023.

[100] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2019.

[101] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, pages 2579–2605, 2008.

[102] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017.

[103] Mengting Wan and Julian McAuley. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, page 86–94. Association for Computing Machinery, 2018.

[104] Cheng Wang, Jiacheng Sun, Zhenhua Dong, Ruixuan Li, and Rui Zhang. Gradient matching for categorical data distillation in ctr prediction. In *Proceedings of the 17th ACM Conference on Recommender Systems*. Association for Computing Machinery, 2023.

[105] Cheng Wang, Jiacheng Sun, Zhenhua Dong, Jieming Zhu, Zhenguo Li, Ruixuan Li, and Rui Zhang. Data-free knowledge distillation for reusing recommendation models. In *Proceedings of the 17th ACM Conference on Recommender Systems*. Association for Computing Machinery, 2023.

[106] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1835–1844. International World Wide Web Conferences Steering Committee, 2018.

[107] Kun Wang, Yuxuan Liang, et al. Brave the wind and the waves: Discovering robust and generalizable graph lottery tickets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 3388–3405, 2024.

[108] Lan Wang, Junjie Peng, Cangzhi Zheng, Tong Zhao, and Li'an Zhu. A cross modal hierarchical fusion multimodal sentiment analysis method based on multi-task learning. *Information Processing & Management*, 2024.

[109] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020.

[110] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

[111] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2019.

[112] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. Association for Computing Machinery, 2019.

[113] Xiwei Wang, Siguleng Wuji, Yutong Liu, Ran Luo, and Chengcheng Qiu. Study on the impact of recommendation algorithms on user perceived stress and health

management behaviour in short video platforms. *Information Processing & Management*, 2024.

[114] Zhiyuan Wang, Cheng Chen, and Ke Tang. Zero-shot knowledge graph completion for recommendation system. In *Proc. of IDEAL'2022*. Springer, 2022.

[115] Zhiyuan Wang, Shengcai Liu, Peng Yang, and Ke Tang. Domain-agnostic coevolution of generalizable parallel algorithm portfolios. *arXiv*, 2025.

[116] Zongwei Wang, Min Gao, Wentao Li, Junliang Yu, Linxin Guo, and Hongzhi Yin. Efficient bi-level optimization for recommendation denoising. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2023.

[117] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. Neural news recommendation with attentive multi-view learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. ijcai.org, 2019.

[118] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. Neural news recommendation with multi-head self-attention. 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)*, pages 6389–6394. Association for Computational Linguistics, 2019.

[119] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. Fastformer: Additive attention can be all you need. *arXiv preprint arXiv:2108.09084*, 2021.

[120] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 3597–3606. Association for Computational Linguistics, 2020.

[121] Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qing Li, and Ke Tang. Disentangled contrastive learning for social recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, 2022.

[122] Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qijiong Liu, Rui He, Qing Li, and Ke Tang. Dataset condensation for recommendation. In *arXiv*, 2023.

[123] Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qijiong Liu, Rui He, Qing Li, and Ke Tang. Condensing pre-augmented recommendation data via lightweight policy gradient estimation. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[124] Jiahao Wu, Qijiong Liu, Hengchang Hu, Wenqi Fan, Shengcai Liu, Qing Li, Xiao-Ming Wu, and Ke Tang. Leveraging large language models (llms) to empower training-free dataset condensation for content-based recommendation. *arXiv*, 2023.

[125] Jiahao Wu, Ning Lu, Zeiyu Dai, Wenqi Fan, Shengcai Liu, Qing Li, and Ke Tang. Backdoor graph condensation. *arXiv*, 2024.

[126] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 726–735. Association for Computing Machinery, 2021.

[127] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. A survey on accuracy-oriented neural recommendation: From collaborative filtering to

information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[128] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[129] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 235–244. Association for Computing Machinery, 2019.

[130] Lirong Wu, Haitao Lin, Zhangyang Gao, Cheng Tan, Stan Li, et al. Self-supervised learning on graphs: Contrastive, generative,or predictive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[131] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*. Association for Computing Machinery, 2019.

[132] Xindi Wu, Zhiwei Deng, and Olga Russakovsky. Multimodal dataset distillation for image-text retrieval, 2023.

[133] Lianghao Xia, Chao Huang, Yong Xu, and Jian Pei. Multi-behavior sequential recommendation with temporal graph transformer. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[134] Lianghao Xia, Yizhen Shao, Chao Huang, Yong Xu, Huance Xu, and Jian Pei. Disentangled graph social recommendation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023.

[135] Shuaiyong Xiao, Jianxiong Wang, Jiwei Wang, Runlin Chen, and Gang Chen. On the consensus of synchronous temporal and spatial views: A novel multi-modal deep learning method for social video prediction. *Information Processing & Management*, 2024.

[136] Yang Xu, Lei Zhu, Zhiyong Cheng, Jingjing Li, Zheng Zhang, and Huaxiang Zhang. Multi-modal discrete collaborative filtering for efficient cold-start recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[137] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 5812–5823. Curran Associates Inc., 2020.

[138] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, and Z. Huang. Self-supervised learning for recommender systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[139] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[140] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. Socially-aware self-supervised tri-training for recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 2084–2092. Association for Computing Machinery, 2021.

[141] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. Self-supervised multi-channel hypergraph con-

volutional network for social recommendation. In *Proceedings of the 32nd Web Conference*, pages 413–424. ACM / IW3C2, 2021.

[142] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, and Xia Hu. Data-centric ai: Perspectives and challenges. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, 2023.

[143] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. Data-centric artificial intelligence: A survey. *arXiv*, 2023.

[144] Bo-Jian Zhang, Guang-Hai Liu, Zuoyong Li, and Shu-Xiang Song. Image retrieval using compact deep semantic correlation descriptors. *Information Processing & Management*, 2024.

[145] David Junhao Zhang, Heng Wang, Chuhui Xue, Rui Yan, Wenqing Zhang, Song Bai, and Mike Zheng Shou. Dataset condensation via generative model. *arXiv*, 2023.

[146] Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. Unbert: User-news matching bert for news recommendation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3356–3362. International Joint Conferences on Artificial Intelligence Organization, 2021.

[147] Zhen Zhang, Guanhua Zhang, Bairu Hou, Wenqi Fan, Qing Li, Sijia Liu, Yang Zhang, and Shiyu Chang. Certified robustness for large language models with self-denoising. *arXiv*, 2023.

[148] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

[149] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2023.

[150] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021.

[151] Tong Zhao, Julian McAuley, and Irwin King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, 2014.

[152] Xiangyu Zhao, Haochen Liu, Wenqi Fan, Hui Liu, Jiliang Tang, and Chong Wang. Autoloss: Automated loss function search in recommendations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, 2021.

[153] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering*, pages 6889–6907, 2024.

[154] Xiangyu Zhaok, Haochen Liu, Wenqi Fan, Hui Liu, Jiliang Tang, Chong Wang, Ming Chen, Xudong Zheng, Xiaobing Liu, and Xiwang Yang. Autoemb: Automated embedding dimensionality search in streaming recommendations. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021.

[155] Xin Zheng, Miao Zhang, Chunyang Chen, Quoc Viet Hung Nguyen, Xingquan Zhu, and Shirui Pan. Structure-free graph condensation: From large-scale graphs to condensed graph-free data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[156] Xiao Zhou, Renjie Pi, Weizhong Zhang, Yong Lin, Zonghao Chen, and Tong Zhang. Probabilistic bilevel coreset selection. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

[157] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *The World Wide Web Conference*. Association for Computing Machinery, 2021.