修　士　論　文

Distributional User Preference Modeling for Cross-Domain Recommendation under Non-Overlapping Settings

非重複設定下におけるクロスドメイン推薦のための確率分布に基づくユーザ嗜好モデリング

指導教員　　　鈴村 豊太郎 教授

東京大学大学院 情報理工学系研究科
電子情報学専攻

氏 名　　48-246453 肖 子吟

提 出 日　　　2026 年 1 月 22 日

Abstract

Recommendation systems play an important role in helping users discover relevant items in many applications, such as e-commerce platforms, streaming services, and social media. As online content continues to grow rapidly, building effective recommendation systems has become increasingly important.

However, traditional recommendation systems often suffer from data sparsity and cold-start problems, which limit their performance. Cross-domain recommendation has therefore been proposed as a way to improve recommendation quality by utilizing information from related domains. Most existing cross-domain recommendation methods, however, assume that users or items overlap across domains during training, an assumption that is often unrealistic in practice. In addition, user preferences are usually represented as point embeddings, which makes it difficult to capture the uncertainty and diversity of real user interests.

In this thesis, we propose a cross-domain recommendation framework that models user preferences as Gaussian Mixture Models (GMMs) under a non-overlapping training setting. By representing user preferences as probability distributions, the proposed framework can better capture diverse and uncertain user interests. Based on these distributional representations, optimal transport is applied at the distribution level to align user preferences across domains and enable knowledge transfer at inference time when cross-domain user correspondence is available. Importantly, the proposed method does not rely on overlapping users or items during training, but can flexibly incorporate overlapping user information at inference time when such information is available.

Extensive experiments on multiple Amazon Review datasets show that the proposed approach consistently outperforms strong single-domain baselines and representative cross-domain recommendation methods. Additional ablation studies further confirm the effectiveness of distribution-based user preference modeling and the benefits of leveraging source-domain information for improving target-domain recommendation performance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1   Background

Information overload has become a critical challenge for users on modern online platforms, including e-commerce websites, streaming services, and social media [1, 2, 3]. Recommender systems play a crucial role in alleviating this problem by providing personalized guidance and selecting items with the highest predicted relevance from a large pool of candidates [4, 5, 6]. By analyzing historical user–item interactions, recommender systems enable users to efficiently discover relevant content, products, or services, thereby improving user satisfaction and engagement.

Beyond enhancing user experience, recommender systems also generate substantial business value by increasing sales, customer retention, and overall platform activity [7]. As the scale, diversity, and dynamics of online content continue to grow, the importance of effective and robust recommender systems is increasingly recognized in both academia and industry.

Despite their success, recommender systems face several fundamental challenges that limit their performance. One major challenge is data sparsity, where users interact with only a small fraction of available items, making it difficult to accurately infer user preferences. This issue is particularly severe in emerging platforms or newly established services with limited historical data. Another critical challenge is the cold-start problem, which arises when new users or items enter the system and insufficient interaction history is available to support reliable recommendations. These challenges arise from the limited availability of user–item interaction data and can significantly degrade recommendation quality.

To address these challenges, cross-domain recommendation has emerged as a promising paradigm that leverages information from a related source domain to improve recommendation performance in a target domain. By transferring knowledge across domains, cross-domain recommendation can alleviate data sparsity and cold-start issues, particularly in data-scarce settings. However, most existing cross-domain recommendation methods rely on the assumption that users or items overlap between the source and target domains, and such overlapping entities are used as bridges for knowledge transfer during training. In many real-world scenarios, this assumption does not hold, as domains may be entirely disjoint due to privacy constraints, system isolation, or independent platform design.

Moreover, traditional recommender systems typically represent user preferences as point estimates in a latent space, such as embedding vectors. Such representations are widely used and can be effective in many scenarios. However, point embeddings may be often insufficient to capture the uncertainty, heterogeneity, and multi-faceted nature of real user behavior, especially under sparse data conditions. This limitation further motivates the exploration of more expressive preference representations for cross-domain recommendation.

Motivated by these challenges, this thesis proposes a novel cross-domain recommendation framework that models user preferences as probabilistic distributions, specifically, Gaussian Mixture Models (GMMs), and integrates optimal transport techniques to enable knowledge transfer under non-overlapping training settings. By representing user preferences as probability distributions rather than point estimates, the proposed approach captures both the diversity and uncertainty of user interests. And the multi-peaked structure of GMMs allows for modeling multiple distinct user preferences effectively. Furthermore, optimal transport provides a principled mechanism for aligning user preference representations across domains, facilitating effective knowledge transfer without relying on overlapping users or items during the training phase.

The proposed framework consists of two key components: (1) modeling user preferences as Gaussian mixture distributions to capture the complex structure of user behavior within each domain, and (2) employing optimal transport to align these preference representations between the source and target domains. Together, these components enable robust cross-domain recommendation and improve recommendation accuracy in the target domain under strictly non-overlapping training settings.

Training-time non-overlap and inference-time overlap.   It is important to clarify our assumptions regarding cross-domain user information. In this work, we explicitly distinguish between its availability during training and inference. Specifically, we assume a *non-overlapping setting during training*, where no explicit user or item correspondences between domains are available. This assumption reflects realistic deployment constraints, as it is often infeasible to obtain user-level cross-platform identity information for all users in advance due to privacy regulations, system isolation, or the lack of prior user consent.

At inference time, however, the situation can be different. When users actively interact with a client application, platforms may request explicit authorization from individual users to link their accounts across domains. If such authorization is granted, cross-domain user correspondence becomes available for those users, enabling the system to leverage their historical preferences from multiple domains. In this case, the proposed framework can incorporate overlapping user information at inference time to enhance recommendation performance.

Importantly, the proposed DUP-OT framework does not rely on overlapping users or items during training. User preferences are modeled as structured distributions to enhance expressive capacity, while cross-domain knowledge transfer under non-overlapping settings is achieved through optimal transport based on semantic alignment. When overlapping user information becomes available at inference time, it can be incorporated as an optional enhancement rather than a prerequisite, allowing the model to flexibly adapt to different real-world deployment scenarios.

1.2   Contributions

The main contributions of this thesis are summarized as follows:

- We propose a novel cross-domain recommendation framework designed for non-overlapping training settings, where no shared users or items are available between domains.

- We model user preferences as Gaussian Mixture Models (GMMs), enabling a distribution-based representation that captures uncertainty and multi-interest user preferences more effectively than conventional point-embedding approaches.

- We integrate optimal transport techniques to align user preference representations across domains, enabling principled and interpretable cross-domain knowledge transfer without requiring overlapping entities during training.

- We conduct extensive experiments on multiple Amazon Review datasets to evaluate the proposed framework under various cross-domain settings. The results demonstrate that the proposed method consistently outperforms both traditional single-domain recommenders and representative cross-domain baselines.

- We perform ablation studies to analyze the effects of distribution-based preference modeling and cross-domain knowledge transfer, providing insights into the contributions of each component of the framework.

1.3   Thesis Organization

The remainder of this thesis is organized as follows:

- Chapter 2 introduces the fundamental concepts and techniques relevant to this work, including recommender systems, cross-domain recommendation, Gaussian Mixture Models, and optimal transport.

- Chapter 3 reviews related work in recommender systems, cross-domain recommendation, distributional user preference modeling, and optimal transport for representation alignment.

- Chapter 4 presents the proposed DUP-OT framework in detail, including shared preprocessing stage, user GMM weights learning stage, and cross-domain rating prediction stage.

- Chapter 5 describes the experimental setup, datasets, baseline methods, and evaluation metrics used to assess the performance of the proposed framework. Also, implementation details are provided to facilitate reproducibility.

- Chapter 6 reports and analyzes the experimental results, including comparisons with baseline methods and ablation studies.

- Chapter 7 concludes the thesis and discusses limitations and directions for future research.

Chapter 2

Preliminaries

2.1   Recommender Systems

2.1.1   Problem Definition

The core objective of recommender systems is to predict user preferences for items that they have not yet interacted with. And based on these preference predictions, the system generates personalized recommendations for each user.

Mathematically, recommendation tasks are often formulated as a mapping function that maps user–item interaction pairs to ratings or preference scores.

Definition of Basic Symbols   We first define the following basic sets and symbols:

- $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$: the set of users, where M is the total number of users.

- $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$: the set of items, where N is the total number of items.

- $\mathcal{R} \subseteq \mathbb{R}$: the space of possible ratings (in explicit feedback) or preference scores (in implicit feedback).

In explicit feedback scenarios, $\mathcal{R}$ is usually a set of finite discrete values, e.g., $\{1, 2, 3, 4, 5\}$ for a 5-star rating system. In implicit feedback settings, user preferences are inferred from observed behavioral signals rather than explicit ratings. Such feedback is typically represented as abstract preference indicators or scores, and is commonly modeled within a non-negative real-valued space, i.e., $\mathcal{R} \subseteq \mathbb{R}_{\geq 0}$.

Formulated Recommender System   The target of a recommender system can be formulated as learning a prediction function:

$$f : \mathcal{U} \times \mathcal{V} \to \mathcal{R}, \tag{2.1}$$

where $f(u, v)$ predicts the rating or preference score of user $u \in \mathcal{U}$ for item $v \in \mathcal{V}$. In explicit feedback scenarios, $\hat{r}_{uv} = f(u, v)$ represents an estimation of the true rating r_{uv}.

At test time, for each user $u \in \mathcal{U}$, the recommender system generates a ranked list of top-K items based on the predicted scores to generate personalized recommendations.

2.1.2   User–Item Interaction Matrix

In most recommender systems, user–item interactions can be represented as a user–item interaction matrix:

$$\mathbf{R} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{V}|}, \quad R_{uv} = \begin{cases} r_{uv}, & \text{if } (u,v) \text{ is observed,} \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

This matrix typically has following characteristics:

- **Highly Sparse**: In real-world applications, users usually interact with only a small fraction of available items, leading to a sparse interaction matrix.

- **Noise and Uncertainty**: User interactions can be noisy and uncertain due to various factors such as changing preferences, context, and external influences.

Intuitively, the essential task of recommender systems is: how to reasonably fill the unobserved entries in the user–item interaction matrix based on the observed ones in such a sparse and noisy matrix.

2.1.3   Prediction vs. Recommendation Objectives

In recommender systems, there are two primary objectives: rating prediction and top-K recommendation.

(1) Rating Prediction   The rating prediction task focuses on accurately estimating the numerical ratings to make the predictions as close as possible to the actual user ratings. The task can be formulated as:

$$\min_{\Theta} \sum_{(u,v) \in \mathcal{D}_{\text{train}}} \mathcal{L}\big(r_{uv}, \hat{r}_{uv}\big), \tag{2.3}$$

where Θ denotes the model parameters, and $\mathcal{L}(\cdot, \cdot)$ usually represents the Mean Squared Error (MSE) loss or Mean Absolute Error (MAE) loss.

This objective is commonly used in rating datasets where explicit user ratings are available, such as the MovieLens, Amazon Reviews, etc. And the evaluation metrics typically include Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) in rating prediction tasks.

(2) Top-K Recommendation   In many real-world applications, the primary goal of recommender systems is not to predict exact ratings but to generate a ranked list of top-K items for each user:

$$\pi_u = \text{rank}\big(\{\hat{r}_{uv}\}_{v \in \mathcal{V}}\big), \tag{2.4}$$

where π_u denotes the ranked list of items for user u based on the predicted scores $\hat{r}_{uv}$.

This objective is typically optimized using ranking-based loss functions, such as pairwise losses (e.g., Bayesian Personalized Ranking) or listwise ranking losses.

The evaluation metrics for top-K recommendation tasks often include Precision@K, Recall@K, Normalized Discounted Cumulative Gain (NDCG)@K, and Mean Average Precision (MAP)@K.

2.1.4   Latent Representation Learning

The core idea of modern recommender systems is mapping users and items into a shared low-dimensional latent space, where user preferences can be effectively captured through interactions in this space. This process is known as latent representation learning or embedding learning.

Latent Vector Representations   In latent representation learning, each user and item is represented by a dense vector in a low-dimensional space:

$$\mathbf{p}_u \in \mathbb{R}^d, \quad \mathbf{q}_v \in \mathbb{R}^d. \tag{2.5}$$

And the prediction function is often defined as:

$$\hat{r}_{uv} = g(\mathbf{p}_u, \mathbf{q}_v), \tag{2.6}$$

where $g(\cdot, \cdot)$ is a similarity function, can be:

- **Dot Product**: $g(\mathbf{p}_u, \mathbf{q}_v) = \mathbf{p}_u^\top \mathbf{q}_v$.

- **MLP**: $g(\mathbf{p}_u, \mathbf{q}_v) = \mathrm{MLP}([\mathbf{p}_u; \mathbf{q}_v])$.

- **Other Non-linear Functions**: e.g., cosine similarity, neural attention mechanisms, etc.

The Intuitive Understanding of Representation Learning   The users' latent vectors $\mathbf{p}_u$ can be interpreted as the user's preferences over various latent factors, while the items' latent vectors $\mathbf{q}_v$ represent the item's attributes in the same latent factor space.

Similar users or items are expected to have similar latent representations, which means their representation vectors should be close in the latent space.

However, this modeling approach has an implicit assumption that users' preferences can be fully captured by a single point in the latent space. This assumption may not hold in practice, as user preferences are often complex, diverse, and uncertain. That's the motivation for us to explore distributional representations of user preferences in this thesis.

2.1.5   Challenges of Recommender Systems

Despite the significant advancements in recommender systems, several challenges remain that hinder their effectiveness and user satisfaction:

- **Data Sparsity**: The user–item interaction matrix is often extremely sparse, making it difficult to learn accurate user preferences.

- **Cold Start Problem**: New users or items with little to no interaction history pose challenges for recommendation accuracy.

- **Preference Uncertainty and Diversity**: Users' preferences can be diverse and uncertain, making it challenging to capture their true interests with point embeddings.

2.1.6   Connections to This Thesis

The recommendation problem focused in this thesis can be seen as an extension of the conventional recommender systems, which:

- Does not model user preferences as deterministic point embeddings, but as probability distributions to capture the uncertainty and diversity of user preferences.

- Conducts structured alignment and transfer of user preference distributions across different domains to address data sparsity and cold-start issues.

So, in the following sections, we will introduce Cross-Domain Recommendation Systems, and formally define the fundamental settings of cross-domain recommendation tasks.

2.2   Cross-Domain Recommendation Systems

The goal of Cross-Domain Recommendation (CDR) systems is to improve recommendation performance in a target domain by leveraging knowledge from one or more source domains.

Compared to traditional single-domain recommender systems, CDR systems explicitly consider the relationships and knowledge transfer between different domains, and enable knowledge transfer across domains to alleviate data sparsity and cold-start problems.

2.2.1   Definition of Domains

To formally define CDR, we first introduce the concept of a domain. A domain $\mathcal{D}$ can be defined as a tuple:

$$\mathcal{D} = (\mathcal{U}, \mathcal{V}, \mathcal{R}, \mathcal{X}), \tag{2.7}$$

where $\mathcal{U}$ is the set of users, $\mathcal{V}$ is the set of items, $\mathcal{R}$ is the user–item interaction matrix within that domain, and $\mathcal{X}$ represents any auxiliary information (e.g., user profiles, item attributes, contextual data) available in that domain.

In CDR, we consider at least two domains:

- **Source Domain** $\mathcal{D}_s = (\mathcal{U}_s, \mathcal{V}_s, \mathcal{R}_s, \mathcal{X}_s)$: The domain from which knowledge is transferred.

- **Target Domain** $\mathcal{D}_t = (\mathcal{U}_t, \mathcal{V}_t, \mathcal{R}_t, \mathcal{X}_t)$: The domain where we want to improve recommendation performance.

The primary objective of CDR is to leverage the information from the source domain $\mathcal{D}_s$ to enhance the recommendation quality in the target domain $\mathcal{D}_t$, especially when the target domain suffers from data sparsity or cold-start issues.

2.2.2   Learning Objectives in CDR

Given observed user–item interactions in both source and target domains, cross-domain recommendation systems aim to learn a prediction function for the target domain:

$$f_t : \mathcal{U}_t \times \mathcal{V}_t \rightarrow \mathcal{R}_t, \tag{2.8}$$

such that the learned function f_t benefits from the knowledge transferred from the source domain $\mathcal{D}_s$.

In general, this process can be formulated as a joint learning problem:

$$\min_{\Theta_s, \Theta_t} \mathcal{L}_t(\mathcal{R}_t; \Theta_t) + \lambda\, \mathcal{L}_{\text{transfer}}(\mathcal{R}_s, \mathcal{R}_t; \Theta_s, \Theta_t), \tag{2.9}$$

where Θ_s and Θ_t denote the model parameters for the source and target domains, respectively.

The first term $\mathcal{L}_t$ represents the loss function for the target domain, which measures the prediction error on the target domain interactions. In explicit feedback scenarios, this can be the Mean Squared Error (MSE) loss between the predicted ratings and the true ratings in the target domain. In implicit feedback settings, ranking-based loss functions such as Bayesian Personalized Ranking (BPR) loss can be employed.

The second term $\mathcal{L}_{\text{transfer}}$ captures the knowledge transfer between the source and target domains, encouraging the model to leverage useful information from the source domain to improve performance in the target domain. This term can be seen as a regularization that aligns the representations or distributions of users/items across domains.

The hyperparameter λ controls the balance between fitting the target domain data and transferring knowledge from the source domain.

The specific design of the transfer loss $\mathcal{L}_{\text{transfer}}$ can vary depending on the chosen transfer learning strategy, such as shared latent factors, domain adaptation techniques, or distributional alignment methods.

2.2.3   Overlapping Cross-Domain Recommendation

The overlapping CDR scenario is widely assumed in many relative research works. In this setting, the source and target domains share some common entities, which can be either users or items.

Shared Users or Items   Typical overlapping settings include:

- **User-Overlapping**: $\mathcal{U}_s \cap \mathcal{U}_t \neq \emptyset$

- **Item-Overlapping**: $\mathcal{V}_s \cap \mathcal{V}_t \neq \emptyset$

In these scenarios, the shared users or items serve as natural bridges for knowledge transfer. Leveraging these overlapping entities can facilitate the alignment of user preferences or item characteristics across domains, thereby enhancing the recommendation performance in the target domain.

For example, shared users provide direct supervision signals for cross-domain preference consistency, enabling the alignment of user latent representations via contrastive learning, or serving as bridging nodes that connect multiple domains in GNN-based models.

Characteristics of Overlapping CDR   Overlapping cross-domain recommendation typically exhibits the following properties:

- Knowledge transfer is facilitated by explicitly shared entities.

- Cross-domain alignment is often achieved by parameter sharing or regularization on shared embeddings.

- Performance strongly depends on the number and quality of overlapping users or items.

While effective in many applications, these approaches are limited when overlapping entities are scarce or entirely unavailable.

2.2.4   Non-Overlapping Cross-Domain Recommendation

In contrast, **non-overlapping cross-domain recommendation** considers a more challenging setting where no users or items are shared across domains:

$$\mathcal{U}_s \cap \mathcal{U}_t = \emptyset, \quad \mathcal{V}_s \cap \mathcal{V}_t = \emptyset. \tag{2.10}$$

This setting naturally arises in practical scenarios such as:

- Recommendation across different platforms or services,

- Cold-start deployment of a new domain,

- Privacy-preserving or federated environments.

Challenges in the Non-Overlapping Setting   Without shared entities, cross-domain knowledge transfer becomes significantly more difficult due to:

- The absence of explicit correspondence between users or items,

- Domain-specific item distributions and feature spaces,

- Inherent discrepancies in user behavior patterns across domains.

As a result, non-overlapping cross-domain recommendation requires transferring *higher-level structural knowledge*, rather than relying on direct entity matching.

2.2.5   Distribution-Level Knowledge Transfer

To address the lack of overlapping entities, recent research has shifted from instance-level transfer to **distribution-level** or **structure-level** knowledge transfer.

Instead of aligning individual users or items, these approaches focus on:

- Modeling user preferences as distributions over the item space,

- Capturing global or population-level structures shared across domains,

- Aligning preference distributions under appropriate metrics.

This perspective naturally motivates the use of probabilistic models and distance measures between distributions, which form the foundation of the methods introduced in the subsequent sections.

2.2.6   Connection to This Thesis

This thesis focuses on the non-overlapping cross-domain recommendation setting, where neither users nor items are shared across domains. To enable effective knowledge transfer under this challenging scenario, we adopt a distributional view of user preferences and seek principled methods to align such distributions across domains.

In the following sections, we introduce Gaussian Mixture Models for modeling user preference distributions, and Optimal Transport-based distance measures for aligning these distributions in a mathematically grounded manner.

2.3   Gaussian Mixture Models for User Preference Modeling

Gaussian Mixture Models (GMMs) provide a flexible probabilistic framework for modeling complex and multi-modal data distributions. In the context of recommender systems, GMMs are particularly suitable for capturing users' diverse interests and the uncertainty arising from sparse interaction data. In this thesis, GMMs are employed to model user preference distributions over the item embedding space, serving as the foundation for cross-domain knowledge transfer.

2.3.1   Motivation: From Point Embeddings to Distributional Preferences

Conventional recommender systems typically represent each user by a single deterministic latent vector. While effective in dense settings, this point-based representation implicitly assumes that a user's preference can be summarized by a single mode in the latent space.

However, in real-world scenarios, user preferences are often:

- **Multi-faceted**: users may exhibit interest in multiple distinct types of items;

- **Uncertain**: limited interactions lead to ambiguous preference estimation;

- **Domain-dependent**: preference expressions may vary across domains.

These characteristics motivate modeling user preferences as probability distributions rather than point embeddings.

2.3.2   Gaussian Mixture Model Definition

Let $\mathbf{v} \in \mathbb{R}^d$ denote an item embedding in the latent space. A Gaussian Mixture Model represents a probability distribution over $\mathbf{v}$ as a weighted sum of K Gaussian components:

$$p(\mathbf{v}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{v} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.11}$$

where:

- K is the number of mixture components,

- $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$ are mixture weights,

- $\boldsymbol{\mu}_k \in \mathbb{R}^d$ denotes the mean vector,

- $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d \times d}$ denotes the covariance matrix.

Each Gaussian component captures a latent preference mode, while the mixture weights reflect the relative importance of different interests.

2.3.3   User Preference Modeling with GMMs

In this thesis, the preference of a user u is modeled as a Gaussian mixture distribution over the item embedding space:

$$p(\mathbf{v} \mid u) = \sum_{k=1}^{K} \pi_{u,k} \, \mathcal{N}(\mathbf{v} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.12}$$

where $\{\pi_{u,k}\}_{k=1}^{K}$ denotes the user-specific mixture weights.

Notably, the Gaussian components $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ are shared across users within the same domain, while only the mixture weights are personalized. This design enables consistent semantic interpretation of preference components and facilitates cross-domain alignment.

2.3.4   Discrete Approximation of Preference Distributions

In practice, user preference distributions are supported by a finite set of item embeddings. Thus, $p(\mathbf{v} \mid u)$ can be approximated by a discrete measure:

$$p(\mathbf{v} \mid u) \approx \sum_{k=1}^{K} \pi_{u,k} \, \delta\big(\mathbf{v} = \boldsymbol{\mu}_k\big), \tag{2.13}$$

where $\delta(\cdot)$ denotes the Dirac delta function.

This discrete representation plays a crucial role in enabling efficient optimal transport computation between user preference distributions.

2.3.5   Parameter Estimation and Practical Considerations

The Gaussian components can be learned by fitting a GMM to item embeddings within each domain, using standard likelihood-based estimation methods. Once the component parameters are fixed, user-specific mixture weights can be estimated based on observed interactions.

This separation between global component estimation and user-level weight learning provides a stable and scalable framework for modeling user preferences, especially in sparse recommendation settings.

2.3.6   Connection to Optimal Transport

Modeling user preferences as GMMs allows cross-domain alignment to be formulated as a distribution matching problem. Given two user preference distributions from different domains, their discrepancy can be naturally measured using Optimal Transport.

In particular, when preference distributions are represented as discrete mixtures over Gaussian components, the optimal transport problem reduces to transporting mass between Gaussian components under an appropriate cost function. This formulation directly motivates the use of Wasserstein distance and Optimal Transport techniques introduced in the next section.

2.4   Optimal Transport for Distribution Alignment

Optimal Transport (OT) provides a principled framework for comparing probability distributions by explicitly modeling the cost of transporting probability mass. In this thesis, OT is employed as a distribution-level alignment mechanism for user preference distributions modeled by Gaussian Mixture Models.

2.4.1   General Formulation

Let $\mu, \nu \in \mathcal{P}(\mathcal{X})$ be two probability measures defined on a metric space $(\mathcal{X}, d)$. The Kantorovich formulation of optimal transport seeks a transport plan $\gamma \in \Pi(\mu, \nu)$ that minimizes the expected transportation cost:

$$\min_{\gamma \in \Pi(\mu, \nu)} \int_{\mathcal{X} \times \mathcal{X}} d(x, y) \, d\gamma(x, y), \tag{2.14}$$

where $\Pi(\mu, \nu)$ denotes the set of joint distributions whose marginals are μ and ν.

Based on this formulation, the p-Wasserstein distance between μ and ν is defined as:

$$W_p(\mu, \nu) = \left(\min_{\gamma \in \Pi(\mu, \nu)} \int_{\mathcal{X} \times \mathcal{X}} d(x, y)^p \, d\gamma(x, y) \right)^{\frac{1}{p}}. \tag{2.15}$$

2.4.2   Discrete Optimal Transport

In practice, user preference distributions are represented by discrete measures. Let:

$$\mu = \sum_{i=1}^{m} a_i \, \delta_{x_i}, \quad \nu = \sum_{j=1}^{n} b_j \, \delta_{y_j}, \tag{2.16}$$

where $a_i, b_j \geq 0$ and $\sum_i a_i = \sum_j b_j = 1$.

The discrete optimal transport problem can be written as:

$$\min_{\mathbf{T} \in \mathbb{R}_+^{m \times n}} \sum_{i=1}^{m} \sum_{j=1}^{n} T_{ij} \, c(x_i, y_j), \tag{2.17}$$

subject to:

$$\mathbf{T}\mathbf{1}_n = \mathbf{a}, \quad \mathbf{T}^\top \mathbf{1}_m = \mathbf{b}, \tag{2.18}$$

where $\mathbf{T}$ is the transport matrix and $c(x_i, y_j)$ denotes the ground cost.

2.4.3 Optimal Transport Between Gaussian Mixture Models

In this thesis, user preferences are modeled as Gaussian Mixture Models (GMMs). For a user u, the preference distribution is given by:

$$p(\mathbf{v} \mid u) = \sum_{k=1}^{K} \pi_{u,k} \mathcal{N}(\mathbf{v} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{2.19}$$

A GMM can be viewed as a discrete distribution over Gaussian components, where each component represents a latent preference mode. Accordingly, the optimal transport problem between two GMMs can be formulated at the component level.

Specifically, given two GMMs with mixture weights $\boldsymbol{\pi}^{(s)}$ and $\boldsymbol{\pi}^{(t)}$, the optimal transport problem reduces to:

$$\min_{\mathbf{T} \in \mathbb{R}_+^{K \times K}} \sum_{k=1}^{K} \sum_{l=1}^{K} T_{kl}\, c\Big(\mathcal{N}(\boldsymbol{\mu}_k^{(s)}, \boldsymbol{\Sigma}_k^{(s)}), \mathcal{N}(\boldsymbol{\mu}_l^{(t)}, \boldsymbol{\Sigma}_l^{(t)})\Big), \tag{2.20}$$

subject to:

$$\mathbf{T}\mathbf{1} = \boldsymbol{\pi}^{(s)}, \quad \mathbf{T}^\top \mathbf{1} = \boldsymbol{\pi}^{(t)}, \tag{2.21}$$

where $c(\cdot, \cdot)$ denotes the cost between Gaussian components.

A common choice for $c(\cdot, \cdot)$ is the squared 2-Wasserstein distance between Gaussian distributions, which admits a closed-form expression. The resulting transport matrix $\mathbf{T}$ characterizes how preference mass is aligned across mixture components.

2.4.4 Connection to This Thesis

By formulating user preference alignment as an optimal transport problem between Gaussian mixture models, this thesis enables distribution-level knowledge transfer across domains. This approach avoids reliance on overlapping users or items and provides a geometrically meaningful mechanism for cross-domain preference alignment.

Chapter 3

Related Work

3.1   Recommender Systems

Recommender systems aim to alleviate information overload by modeling user preferences and predicting the relevance of items from large candidate sets. Over the past decades, a wide range of recommendation techniques have been proposed, which can be broadly categorized into collaborative filtering-based methods, content-based methods, and hybrid approaches. These paradigms differ in how user preferences are represented, how item relevance is estimated, and how auxiliary information is exploited. In this section, we briefly review representative recommender system techniques that form the foundation of modern recommendation models.

3.1.1   Collaborative Filtering

Collaborative Filtering (CF) is one of the most widely studied paradigms in recommender systems [8, 9]. CF methods rely solely on historical user-item interaction data to infer user preferences, under the assumption that users with similar interaction patterns tend to exhibit similar tastes. By abstracting away explicit item content, CF techniques are particularly effective in capturing collective behavioral patterns and have been extensively applied in rating prediction and top-K recommendation tasks.

3.1.1.1   Memory-based CF

By user or item similarities, memory-based CF can be further categorized into user-based and item-based approaches. User-based CF recommends items to a user based on the preferences of similar users, while item-based CF suggests items similar to those the user has previously liked.

GroupLens [10] is one of the earliest and most influential memory-based CF systems, which introduced user-based collaborative filtering using Pearson correlation to compute user similarities. It demonstrated the effectiveness of CF in providing personalized recommendations and laid the groundwork for subsequent research in the field.

Sarwar et al. [11] points out that when the number of users and items is very large, user-based CF can be computationally expensive and the sparsity of the user-item interaction matrix can lead

to poor recommendation quality. Noticing that users' preferences change quickly over time but items' characteristics are relatively stable, they proposed an item-based CF approach that computes item similarities based on user interactions. The merit of this method is that item similarities can be precomputed and stored, allowing for efficient recommendation generation. And the number of items is usually much smaller than the number of users, which helps alleviate the data sparsity issue. Also, one important attribution of this paper is that it proposed a new similarity measure called adjusted cosine similarity, which accounts for individual user rating biases when computing item similarities. This method has since become a standard technique in item-based CF and has been widely adopted in various recommender systems.

3.1.1.2   Probabilistic Latent Factor Models

Early collaborative filtering methods are dominated by probabilistic latent factor models, which represent users and items as low-dimensional latent vectors and model observed interactions through their inner products. These models typically assume that user preferences and item characteristics are generated from latent variables, and interaction data are modeled via probabilistic distributions. Representative approaches include probabilistic matrix factorization and its extensions, which offer interpretable formulations and solid theoretical foundations for preference modeling.

Breese et al. [12] conducted a comprehensive analysis of both memory-based and model-based CF methods. This paper is the first to systematically distinguish between memory-based and model-based CF approaches, providing a detailed comparison of their strengths and weaknesses. The model-based methods mentioned in this paper include Bayesian Clustering and Bayesian Networks. Bayesian Clustering groups users into clusters based on their preferences, while Bayesian Networks model the probabilistic relationships between users and items.

Ungar and Foster [13] pointed out that traditional clustering-based collaborative filtering methods suffer from instability and poor generalization when interaction data are highly sparse, as approaches based on KNN or simple K-means clustering rely heavily on local similarity patterns. To address this issue, they proposed Gibbs clustering for collaborative filtering, a probabilistic co-clustering approach that jointly clusters users and items into latent classes and models their interactions through class-level link probabilities. By explicitly formulating a generative model and employing Gibbs sampling for inference, their method enforces global consistency in user and item assignments and provides a principled alternative to heuristic clustering.

Koren et al. [14] systematically review matrix factorization techniques for recommender systems, demonstrating that latent factor models with bias, implicit feedback, and temporal dynamics achieve consistently superior accuracy and scalability over neighborhood-based methods, and establishing matrix factorization as a dominant model-based collaborative filtering paradigm.

Salakhutdinov and Mnih introduce Probabilistic Matrix Factorization as a scalable latent factor model for large, sparse recommender systems, and further extend it to a fully Bayesian framework using MCMC, significantly improving robustness and generalization—especially for infrequent users—while establishing PMF as a foundational model-based collaborative filtering paradigm [15, 16].

3.1.1.3   Deep Learning-based CF

With the advancement of deep learning, neural network-based collaborative filtering methods have been proposed to overcome the limited expressiveness of linear latent factor models. These approaches leverage multilayer perceptrons, autoencoders, or neural interaction functions to capture complex, nonlinear user-item relationships. By learning high-capacity representations from interaction data, deep CF models have achieved strong empirical performance and become a dominant paradigm in modern recommender systems.

Neural Interaction Modeling. Neural interaction modeling methods extend traditional latent factor approaches by learning flexible, non-linear user-item interaction functions through neural networks. Rather than assuming a fixed inner product in the latent space, these models parameterize interaction mechanisms using multi-layer architectures, enabling richer expressiveness and improved modeling capacity. By directly learning interaction patterns from data, neural interaction models overcome the representational limitations of linear factorization methods and form a foundational branch of deep learning-based collaborative filtering.

He et al. [17] proposed the Neural Collaborative Filtering (NCF) framework, which formulates collaborative filtering as a neural interaction learning problem. Instead of relying on a fixed inner product as in traditional matrix factorization, NCF employs neural networks to learn flexible and non-linear user-item interaction functions directly from data. In this framework, users and items are embedded into low-dimensional latent spaces and their representations are combined through neural architectures—such as generalized matrix factorization (GMF), multi-layer perceptrons (MLP), and their fusion model NeuMF—to capture complex interaction patterns beyond linear similarity measures.

Sequential Recommendation. Sequential recommendation focuses on modeling the temporal order and sequential dependencies within users' interaction histories to capture the dynamic evolution of user preferences. By explicitly incorporating sequence information, these methods aim to predict future user behaviors based on past interaction patterns rather than treating interactions as independent events. Sequential recommender systems have become a core component of modern recommendation pipelines, particularly in scenarios where user preferences exhibit strong temporal dynamics.

Hidasi et al. [18] proposed a session-based recommender system that applies recurrent neural networks (RNNs) to model user behavior within individual sessions. By representing a session as a sequence of item interactions and maintaining a recurrent hidden state, their approach captures both short-term and long-term dependencies in session data. Furthermore, the authors introduced ranking-oriented loss functions tailored to recommendation tasks, enabling the model to significantly outperform traditional item-to-item and neighborhood-based baselines.

The mainstream sequential recommender systems now are Transformer-based models. Kang and McAuley [19] proposed SASRec, a self-attentive sequential recommender system based on the Transformer encoder architecture. By employing self-attention mechanisms, SASRec adaptively weighs historical items in a user's interaction sequence, enabling the model to capture long-range dependencies while remaining efficient on sparse data. Unlike recurrent models that summarize sequences through a single hidden state, SASRec directly attends to relevant past interactions, leading to improved recommendation accuracy and scalability in sequential recommendation tasks.

Sun et al. [20] proposed BERT4Rec, a Transformer-based sequential recommender system that employs bidirectional self-attention to model user behavior sequences. Unlike unidirectional sequential models such as RNN-based methods and SASRec, BERT4Rec leverages bidirectional contextual

information by predicting masked items within a sequence using a Cloze-style training objective. This design enables each item representation to incorporate both preceding and succeeding context, leading to more expressive sequence modeling and consistently improved recommendation performance across multiple benchmark datasets.

Graph-based Collaborative Filtering. Graph-based collaborative filtering models user-item interactions as graph-structured data and leverages graph neural networks to learn representations through neighborhood aggregation. By propagating information along interaction edges, these methods capture high-order collaborative signals that are difficult to model with point-wise interaction functions. Graph-based frameworks also naturally support the incorporation of side information and heterogeneous relations, enabling more expressive modeling of user preferences and item characteristics.

Van den Berg et al. [21] proposed Graph Convolutional Matrix Completion (GCMC), which formulates collaborative filtering as a link prediction problem on a bipartite user-item interaction graph. By employing a graph convolutional auto-encoder architecture, GCMC learns user and item representations through message passing on the interaction graph and reconstructs ratings via a bilinear decoder. This approach effectively captures high-order collaborative signals and naturally incorporates side information, leading to improved recommendation performance on benchmark datasets.

Ying et al. [22] proposed PinSage, a graph-based recommender system designed for web-scale applications. PinSage combines graph neural networks with efficient random-walk-based neighborhood sampling to learn item representations that incorporate both graph structure and rich side information. By addressing the scalability limitations of conventional GCNs, PinSage was successfully deployed in large-scale industrial systems such as Pinterest, demonstrating the effectiveness of graph-based recommendation models in real-world production environments.

Wang et al. [23] proposed Neural Graph Collaborative Filtering (NGCF), which explicitly integrates graph neural networks into collaborative filtering by modeling user-item interactions as a bipartite graph. NGCF refines user and item embeddings through recursive message passing on the interaction graph, enabling the explicit modeling of high-order connectivity and collaborative signals, which leads to significant improvements in recommendation performance.

He et al. [24] introduced LightGCN, a simplified graph convolutional network tailored for recommender systems. Unlike prior GNN-based models that incorporate feature transformations and nonlinear activations, LightGCN argues that these components contribute little to collaborative filtering performance when only user and item IDs are available as input. Accordingly, LightGCN retains only the neighborhood aggregation operation to propagate embeddings over the user-item interaction graph, significantly simplifying the model architecture while preserving the ability to capture high-order connectivity. Extensive experiments demonstrate that LightGCN not only reduces computational complexity but also achieves superior recommendation accuracy compared to more complex GNN-based methods such as NGCF, making it a widely adopted and strong baseline in graph-based recommender system research.

Wu et al. [25] proposed SR-GNN, a session-based recommender system that models user interaction sequences as graph-structured data and applies graph neural networks to learn item representations within sessions. By constructing a directed session graph for each interaction sequence, SR-GNN is able to capture complex transition patterns among items that go beyond simple sequential dependencies. This work represents an early attempt to integrate graph-based modeling with sequential recommen-

dation, effectively combining the strengths of GNNs in capturing high-order relational information and sequential models in characterizing short-term user intent.

Generative Recommendation Models. Generative recommendation models formulate the recommendation task from a probabilistic generative perspective, aiming to model the underlying data generation process of user-item interactions. Instead of directly optimizing discriminative prediction objectives, these approaches learn likelihood-based or distributional representations that can capture uncertainty and variability in user preferences. By enabling principled sampling and probabilistic inference, generative models provide a complementary modeling paradigm that has gained increasing attention in recommender systems research.

Liang et al. [26] proposed a variational autoencoder (VAE)-based collaborative filtering framework for implicit feedback data. By modeling user preferences as latent random variables and employing a multinomial likelihood, this approach provides a probabilistic formulation that captures uncertainty and multi-modal structures in user behavior. Compared with deterministic autoencoder-based models, the VAE framework enables more expressive preference modeling and demonstrates strong empirical performance under sparse interaction settings.

Wang et al. [27] proposed IRGAN, a generative adversarial framework that unifies generative and discriminative models for information retrieval and recommendation tasks. By formulating the learning process as a minimax game, the generative model aims to approximate the underlying relevance distribution over items, while the discriminative model learns to distinguish relevant from non-relevant user-item pairs. Through adversarial training, IRGAN effectively improves recommendation performance, particularly under implicit feedback settings.

Chae et al. [28] proposed CFGAN, a generic collaborative filtering framework based on generative adversarial networks. Unlike prior GAN-based recommender systems that generate discrete item indices, CFGAN adopts vector-wise adversarial training, where the generator produces real-valued preference vectors and the discriminator distinguishes them from ground-truth interaction vectors. This design effectively stabilizes adversarial learning and improves recommendation accuracy, especially under sparse implicit feedback settings.

Beyond VAEs and GANs, diffusion models have recently been introduced to recommender systems as a new class of generative models. Wang et al. [29] proposed DiffRec, a diffusion-based recommender system that models the user-item interaction generation process through iterative denoising. By gradually corrupting user interaction histories and learning to recover the original interactions step by step, DiffRec provides a flexible and expressive framework for modeling complex preference distributions. This work demonstrates the potential of diffusion models to overcome the limitations of traditional generative approaches and further improve recommendation performance under noisy and sparse interaction settings.

3.1.2   Content-based and Hybrid Recommendation Methods

Content-based recommendation methods exploit item attributes and user profiles to model user preferences based on item content similarity rather than collective behavior. While such approaches are effective in cold-start scenarios, they often suffer from limited generalization due to narrow user interest modeling. To address the limitations of both collaborative filtering and content-based methods, hybrid recommendation approaches combine interaction data with auxiliary information, aiming to achieve robust performance across diverse recommendation settings.

Salton et al. [30] proposed the vector space model for information retrieval, in which both documents and queries are represented as weighted term vectors. By assigning importance weights to terms (e.g., inverse document frequency) and computing similarity scores between query and document vectors, the model enables ranked retrieval based on relevance. This representation and similarity-matching paradigm laid the foundation for content-based recommender systems, where user profiles and item content are similarly modeled in a shared feature space.

Pazzani and Billsus [31] proposed a content-based recommender system that learns user profiles from explicit user feedback on item content. Their method represents items using content features and employs a naive Bayesian classifier to incrementally learn and revise user preference profiles, enabling the system to predict the interestingness of unseen items.

He et al. [32] developed a large-scale recommender system for Facebook Ads based on gradient boosting decision trees. By modeling user preferences from rich user, item, and contextual features, their approach significantly improved ad targeting effectiveness and user engagement, demonstrating the practicality of supervised learning methods in real-world industrial recommendation scenarios.

Furthermore, advanced gradient boosting frameworks such as XGBoost [33] and LightGBM [34] have been widely applied in recommender systems to enhance both accuracy and scalability. These methods leverage efficient tree-based boosting strategies to model high-order feature interactions, making them particularly suitable for large-scale recommendation tasks with sparse and high-dimensional feature spaces.

Burke [5] presented a comprehensive survey of hybrid recommender systems, systematically categorizing hybridization strategies such as weighted, switching, mixed, and feature combination approaches. The survey analyzed how different hybrid designs integrate multiple recommendation techniques to balance their respective strengths and weaknesses, demonstrating that hybrid methods can effectively improve recommendation accuracy and alleviate issues such as data sparsity and cold-start problems.

Pazzani [35] proposed a unified framework for integrating collaborative, content-based, and demographic filtering methods in recommender systems. By exploiting multiple sources of information, including user – item interactions, item content, and user profiles, the framework combines recommendations from different models to improve precision. Experimental results demonstrated that hybrid approaches within this framework consistently outperform single-method recommenders.

Melville et al. [36] proposed a content-boosted collaborative filtering (CBCF) framework that integrates content-based prediction into the collaborative filtering process. Specifically, a content-based predictor is first used to generate pseudo ratings for unrated items, producing a dense pseudo user – item matrix on which collaborative filtering is subsequently applied. By alleviating sparsity and the first-rater problem, this approach achieves significantly improved recommendation accuracy compared to pure collaborative, pure content-based, and naive hybrid methods.

Billsus et al. [37] developed a hybrid news recommender system for adaptive news access that integrates collaborative filtering and content-based filtering techniques. Their system learns personalized user models from both explicit and implicit user feedback, and combines short-term and long-term interest representations to adapt to users' evolving information needs. Deployed in a real-world news delivery environment, this work demonstrated the practical effectiveness of hybrid recommender systems in improving personalization quality without requiring additional user effort.

Basilico and Hofmann [38] proposed a unified supervised learning framework that integrates collaborative filtering and content-based filtering within a single prediction model. Their approach formulates

19

recommendation as a learning problem over user-item pairs by designing joint feature representations and kernel functions that enable simultaneous generalization across both user and item dimensions. By incorporating user-item interaction data together with item and user attributes, the framework achieves improved recommendation accuracy compared to traditional collaborative or content-based methods.

Rendle [39] introduced Factorization Machines (FMs), a supervised learning model that generalizes matrix factorization by modeling pairwise feature interactions through factorized parameters. By representing user – item interactions, item attributes, and contextual information as sparse feature vectors, FMs can efficiently capture interactions in high-dimensional and highly sparse settings. This unified formulation subsumes several state-of-the-art factorization models and has demonstrated superior performance over traditional collaborative and content-based approaches in various recommendation tasks.

Burges et al. [40] proposed a learning-to-rank framework that directly optimizes the ordering of items rather than predicting absolute preference scores. Their approach formulates ranking as a pairwise learning problem and introduces RankNet, which models ranking preferences using a probabilistic cost function optimized via gradient descent. By focusing on ranking quality, this framework significantly improves recommendation effectiveness in scenarios where the relative order of items is more important than precise rating prediction.

Later, Burges [41] provided a comprehensive overview of learning-to-rank methods, including RankNet, LambdaRank, and LambdaMART. RankNet formulates ranking as a pairwise probabilistic learning problem optimized via gradient descent, while LambdaRank introduces the concept of lambda gradients to directly optimize ranking metrics such as NDCG. By combining LambdaRank with gradient-boosted decision trees, LambdaMART further improves ranking performance and has become a widely adopted approach in large-scale recommendation and information retrieval systems.

3.2   Cross-Domain Recommender Systems

Despite the success of single-domain recommender systems, challenges such as data sparsity and cold-start users remain difficult to address when interaction data are limited. Cross-Domain Recommendation (CDR) tackles these issues by transferring knowledge from a source domain with abundant user-item interactions to a target domain where data are scarce.

The core objective of CDR is to leverage auxiliary information across domains to improve recommendation performance, under the assumption that user preferences or item characteristics exhibit certain transferable patterns. By exploiting correlations between domains, CDR methods aim to enhance recommendation accuracy and robustness, especially in cold-start and sparse-data scenarios.

Based on whether domains share common entities, existing CDR approaches can be broadly categorized into *overlapping* and *non-overlapping* settings. Overlapping CDR methods assume the existence of shared users or items across domains and utilize this overlap as a bridge for knowledge transfer in the training stage. Typical techniques include joint matrix factorization, co-clustering, and graph-based models that align user preferences or item representations across domains.

In contrast, non-overlapping CDR methods address more challenging scenarios where no users or items are shared between domains during training. These approaches generally rely on content features, latent representations, or learned mappings between domains to enable preference transfer

without explicit entity overlap. Non-overlapping CDR is particularly relevant in practical applications, but remains challenging due to the lack of direct correspondence between domains.

3.2.1   Overlapping CDR

Overlapping cross-domain recommendation assumes the existence of shared entities, such as users or items, across domains. These shared entities provide explicit bridges that enable direct knowledge transfer through shared latent representations or alignment constraints. Many existing methods exploit overlapping users or items to jointly learn cross-domain embeddings, often using multi-task learning, graph-based modeling, or contrastive objectives to enforce consistency across domains.

Singh and Gordon [42] proposed Collective Matrix Factorization (CMF), a multi-relational matrix factorization framework that jointly factorizes multiple user – item interaction matrices across domains. By sharing latent factors for overlapping users or items, CMF enables effective knowledge transfer between related domains and improves recommendation performance under data sparsity. This work is widely regarded as a foundational approach for overlapping cross-domain recommendation.

Li et al. [43] proposed a transfer learning framework for collaborative filtering based on a rating-matrix generative model. Their approach captures shared latent rating patterns across domains by learning a common cluster-level rating structure, enabling knowledge transfer even under severe data sparsity. This framework provides a principled probabilistic formulation for overlapping cross-domain recommendation and significantly improves recommendation performance in the target domain.

Pan et al. [44] proposed a transfer learning framework for collaborative filtering that alleviates data sparsity by transferring knowledge from auxiliary domains. Their method discovers shared latent structures of users and items in auxiliary data through matrix factorization and adapts these structures to the target domain via a principled regularization scheme. By exploiting overlapping users or items across domains, this approach effectively captures transferable preference patterns and improves recommendation performance in sparse target domains.

Zhu et al. [45] proposed WCDR, a dual-target cross-domain recommendation framework that models users and items as probability distributions to explicitly capture uncertainty across domains. Unlike prior embedding-based overlapping CDR methods that rely on deterministic representations, WCDR represents user and item behaviors as elliptical Gaussian distributions and decouples them into local-intrinsic and global-domain components. To facilitate effective knowledge transfer between overlapping domains, WCDR further adopts a shared-specific paradigm for global-domain distributions and introduces a Masked Domain-aware Sub-distribution Aggregation (MDSA) module in the Wasserstein space. This design enables adaptive aggregation of domain-shared knowledge while alleviating negative transfer caused by domain imbalance. Extensive experiments demonstrate that modeling domain-level uncertainty and distributional relationships can significantly improve recommendation performance in both source and target domains. Their proposed method can be illustrated as in Figure 3.1.

WCDR models users and items as probability distributions to capture uncertainty in overlapping cross-domain recommendation. By contrast, our work considers a more challenging non-overlapping CDR scenario without any shared users or items, where direct entity-level alignment is unavailable in the training stage and cross-domain transfer must rely on distribution-level modeling.
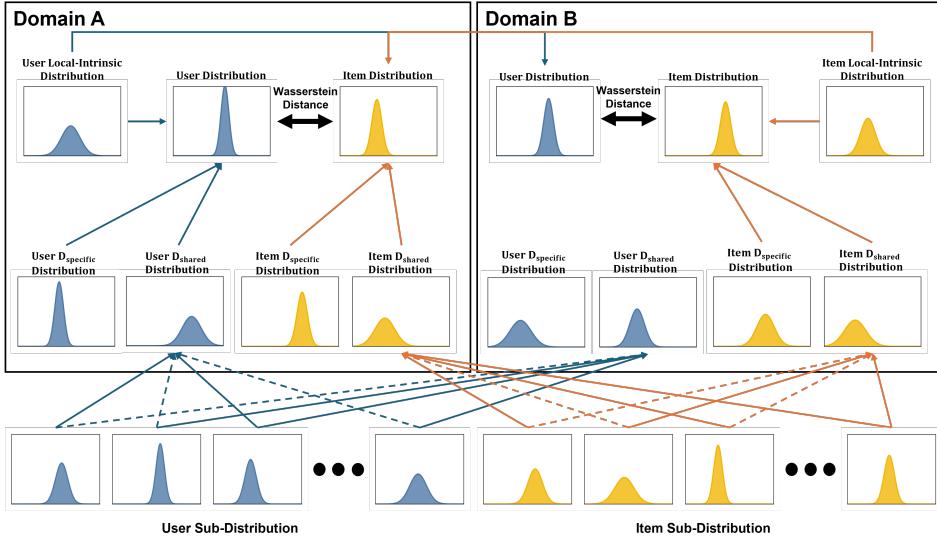
Figure 3.1: Illustration of the WCDR framework [45], which models users and items as probability distributions to capture uncertainty in overlapping cross-domain recommendation.

3.2.2   Non-overlapping CDR

In contrast, non-overlapping cross-domain recommendation addresses a more challenging setting where no shared users or items exist between domains. Without explicit entity-level correspondences, knowledge transfer must rely on higher-level structural, semantic, or distributional similarities across domains. This setting better reflects real-world scenarios involving independent platforms or privacy constraints, and has recently attracted increasing research attention due to its practical significance and methodological difficulty.

Man et al. [46] proposed EMCDR, an embedding-and-mapping framework for cross-domain recommendation. The framework first learns latent user and item representations independently in each domain, and then learns a cross-domain mapping function to project embeddings from the source domain into the target domain. By transferring preferences through latent space mapping, EMCDR enables effective knowledge transfer in non-overlapping and cold-start scenarios.

Beyond latent embedding mapping, several works investigate cross-domain knowledge transfer by aligning heterogeneous feature spaces. Kusano and Oyamada [47] proposed ATP, which formulates non-overlapping cross-domain recommendation as an attribute transportation problem. ATP focuses on scenarios where neither users nor attributes are shared across domains, and aims to estimate cross-domain user similarity by aligning heterogeneous attribute spaces. Specifically, ATP leverages optimal transport to learn a soft correspondence between source and target domain attributes, enabling users to be re-described in a unified attribute space for similarity computation and recommendation. By operating at the attribute level, ATP provides an interpretable and feature-driven solution for non-overlapping cross-domain recommendation.

Liu et al. [48] proposed JPEDET, a joint preference exploration and dynamic embedding transportation framework for non-overlapping cross-domain recommendation. JPEDET addresses the absence of overlapped users and items by learning transferable user representations from both rating and review information, and further aligning user embeddings across domains through a dynamic transportation mechanism.

Specifically, JPEDET first employs a joint preference exploration module to fuse rating-based and review-based user representations, aiming to capture more robust and generalizable user preferences within each domain. To bridge the embedding discrepancy across domains, the model introduces a dynamic embedding transportation module based on neural ordinary differential equations (Neural ODEs), which enables continuous and bidirectional transformation between source and target user embedding spaces. Moreover, JPEDET incorporates optimal transport-based matching regularization and barycentric Wasserstein interpolation to guide the transportation trajectories, leading to smoother and more stable cross-domain alignment.

By modeling cross-domain adaptation as a dynamic and invertible embedding transformation process, JPEDET advances non-overlapping CDR beyond static embedding mapping or adversarial alignment methods, and demonstrates superior performance on multiple real-world benchmark datasets. The overall framework of JPEDET can be illustrated as in Figure 3.2.
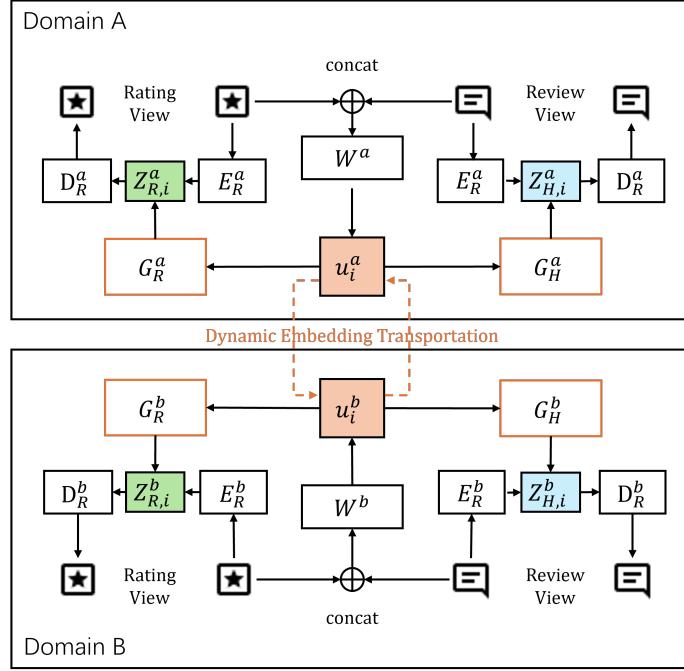

Figure 3.2: Illustration of the JPEDET framework [48], which employs joint preference exploration and dynamic embedding transportation for non-overlapping cross-domain recommendation.

In addition to embedding-level alignment methods, several recent works explore cross-domain rec-

ommendation from a distributional perspective. Liu et al. [49] proposed GWCDR, which addresses non-overlapping cross-domain recommendation by aligning user-item interaction patterns at the distribution level.

Instead of relying on overlapped users or items, GWCDR constructs rating pattern distributions based on observed positive interactions, where each interaction is represented by the element-wise product of the corresponding user and item embeddings. To bridge the discrepancy between domains without assuming a shared embedding space, the model employs Gromov–Wasserstein optimal transport to align the structural relationships within the interaction distributions of the source and target domains.

By preserving the relative geometry of interaction patterns across domains, GWCDR enables effective knowledge transfer under strict non-overlapping settings. This distribution-level alignment paradigm provides a complementary perspective to embedding mapping, attribute transportation, and dynamic embedding transportation methods, and demonstrates strong performance across multiple real-world cross-domain recommendation benchmarks.

3.3   Distributional User Preference Modeling

Recent studies have recognized that user preferences are often uncertain and multi-modal, reflecting diverse and evolving interests. However, most traditional recommender systems represent user preferences as deterministic point embeddings in a latent space, which implicitly assume a single dominant preference pattern.

To better capture preference variability, distributional user preference modeling has been proposed as an alternative paradigm that represents user interests as probability distributions. By modeling preferences at the distribution level, these approaches provide a more expressive representation that can capture uncertainty, preference diversity, and complex user-item interaction patterns.

3.3.1   Multi-interest User Modeling

Multi-interest user modeling aims to capture the fact that a single user may exhibit multiple distinct preference patterns across different contexts or item categories. Instead of learning a single latent representation per user, these methods explicitly model multiple interest vectors or components, allowing recommender systems to better reflect diverse and dynamic user behaviors. Such approaches have been shown to improve recommendation accuracy, especially in scenarios with heterogeneous user preferences.

Zhou et al. [50] proposed the Deep Interest Network (DIN), which models users' diverse interests through a target-aware attention mechanism. Instead of compressing all historical behaviors into a fixed-length representation, DIN dynamically aggregates user behavior embeddings conditioned on the target item, allowing different interests to be activated for different recommendation candidates. This adaptive representation enables DIN to capture the multi-faceted nature of user preferences and has been shown to achieve strong performance in large-scale industrial recommender systems.

Li et al. [51] proposed the Multi-Interest Network with Dynamic Routing (MIND), which explicitly represents each user with multiple interest vectors. MIND employs a dynamic routing mechanism to cluster user behaviors into distinct interest representations, enabling the model to capture diverse

and heterogeneous user preferences. By decoupling interest extraction from item matching, MIND is suitable for large-scale retrieval scenarios and has been successfully deployed in industrial recommender systems.

Cen et al. [52] proposed ComiRec, a controllable multi-interest recommender framework that explicitly models users with multiple interest representations. ComiRec employs capsule networks or self-attention mechanisms to extract diverse user interests from behavior sequences, and introduces an aggregation module to balance recommendation accuracy and diversity. By jointly considering multi-interest extraction and controllable aggregation, ComiRec extends prior multi-interest methods and has demonstrated strong effectiveness in large-scale industrial recommender systems.

3.3.2   Distribution-based Preference Modeling

Beyond discrete multi-interest representations, distribution-based preference modeling represents user preferences using continuous probability distributions. By modeling uncertainty, variance, and multi-modality in user interests, these approaches provide a richer and more flexible representation framework. Distributional modeling techniques, such as Gaussian-based or mixture-based representations, have been explored to capture complex preference structures and improve robustness in sparse or noisy recommendation settings.

Hofmann [53] proposed probabilistic latent semantic analysis (PLSA) for collaborative filtering, formulating user – item interactions as a latent class mixture model. In PLSA, each user is associated with a probability distribution over latent topics (or communities), and each interaction is generated by first sampling a latent topic and then drawing an item conditioned on that topic. As a result, user preferences are represented as distributions over latent semantic factors, allowing different interactions of the same user to be explained by different latent causes, rather than being tied to a single latent representation.

Marlin [54] proposed the User Rating Profile (URP) model, a probabilistic latent variable approach for rating-based collaborative filtering that explicitly models uncertainty in user preferences. URP represents each user as a mixture over latent user attitudes, where the mixture proportions are drawn from a Dirichlet distribution. For each item, a latent attitude is sampled and the corresponding rating is generated according to an attitude-specific rating distribution. By modeling users as distributions over latent preference patterns rather than fixed point representations, URP enables different items rated by the same user to be explained by different latent factors and allows direct inference of rating distributions for unseen items.

Blei, Ng, and Jordan [55] proposed Latent Dirichlet Allocation (LDA), a hierarchical generative probabilistic model that represents each document as a mixture over latent topics, where the topic proportions are drawn from a Dirichlet prior. By introducing a document-level latent variable, LDA provides a fully generative framework that enables principled inference for previously unseen data. Although originally developed for text modeling, LDA has been extended to collaborative filtering by treating users as documents and items as words. Under this formulation, user preferences are modeled as probability distributions over latent topics, allowing each user – item interaction to be explained by different latent factors. This distributional representation enables LDA to capture heterogeneous user interests more flexibly than single-vector latent representations.

Salakhutdinov and Mnih [16] further advanced this direction by introducing Bayesian Probabilistic Matrix Factorization (BPMF), which models user and item latent factors as Gaussian distributions

and performs Bayesian inference to explicitly account for uncertainty in preference estimation.

VAE-based methods provide another probabilistic framework for modeling uncertainty in user preferences. Liang et al. [26] extended variational autoencoders to collaborative filtering by modeling user preferences as latent random variables and performing variational inference. By representing users with posterior distributions in the latent space, VAE-based recommender systems are able to capture both the variability and uncertainty inherent in user behavior, leading to more robust recommendation performance under sparse interaction data.

Beyond recommender system-specific models, some studies investigate representing entities as probability distributions in general representation learning. Vilnis and McCallum [56] proposed Gaussian embeddings, which represent words as Gaussian distributions in the embedding space, enabling the modeling of both semantic uncertainty and asymmetric relationships. Although not originally designed for recommender systems, this distributional representation paradigm provides important methodological insights and has inspired subsequent work on modeling users and items as distributions in recommendation tasks.

3.4 Optimal Transport for Representation Alignment

Optimal Transport (OT) provides a principled mathematical framework for measuring and aligning probability distributions. Originally developed in probability theory and operations research, OT has recently gained significant attention in machine learning for representation alignment and domain adaptation. By computing minimal-cost transformations between distributions, OT enables structured and interpretable alignment across heterogeneous representation spaces.

3.4.1 OT-based Alignment in Cross-Domain Recommendation

In the context of cross-domain recommendation, OT has been employed to align user or item representations across domains by treating them as probability distributions. OT-based alignment offers a natural way to model distribution-level discrepancies while preserving intrinsic geometric structures. This property makes OT particularly suitable for cross-domain scenarios, where direct entity correspondences are unavailable or insufficient, and motivates its adoption in recent cross-domain recommendation frameworks.

Villani [57] systematically developed the theoretical foundations of optimal transport, formalizing the Kantorovich relaxation of the Monge problem and introducing Wasserstein distances as principled metrics between probability measures. By framing distribution comparison as a cost-minimizing mass transportation problem, this work established a rigorous mathematical basis for measuring and aligning probability distributions, which later enabled the adoption of OT as a core tool in machine learning tasks such as distribution alignment, domain adaptation, and representation learning.

Peyré and Cuturi [58] provided a comprehensive survey of optimal transport from a computational and machine learning perspective, systematically bridging OT theory with practical algorithms. Their work reviewed a wide range of OT-based applications, including domain adaptation, generative modeling, and deep learning, and emphasized scalable solutions such as entropic regularization and Sinkhorn iterations. By demonstrating how OT can be efficiently integrated into modern learning pipelines, this

survey established OT as a practical and versatile tool for distribution comparison and representation alignment in machine learning.

Cuturi and Marco [59] introduced the Sinkhorn algorithm, which incorporates an entropic regularization term into the optimal transport formulation to obtain a smooth and strictly convex objective. This regularization allows the resulting OT problem to be solved efficiently via iterative matrix scaling, leading to orders-of-magnitude speedups compared to classical linear programming solvers. As a result, Sinkhorn-based OT distances make optimal transport computationally feasible for large-scale machine learning applications and have become a cornerstone for OT-based representation learning and distribution alignment methods.

Genevay et al. [60] proposed Sinkhorn divergences as a differentiable and computationally tractable OT-based loss for training generative models. By introducing entropic regularization and leveraging automatic differentiation through Sinkhorn iterations, their approach enables stable optimization and efficient gradient computation at scale. This work demonstrated that OT-based losses can effectively improve the quality of generated samples, highlighting the potential of optimal transport as a powerful tool for generative modeling.

Courty et al. [61] proposed a seminal optimal-transport-based framework for unsupervised domain adaptation, in which feature distributions from the source and target domains are aligned by learning an optimal transportation plan. By minimizing the Wasserstein distance between empirical distributions, their method explicitly addresses domain shift at the distribution level and enables effective knowledge transfer from the source domain to the target domain. Moreover, the proposed regularized OT formulation allows the incorporation of class and structural information, leading to improved adaptation performance and establishing OT as a principled tool for representation alignment in domain adaptation.

Frogner et al. [62] introduced a deep learning framework that incorporates the Wasserstein distance as a loss function for representation learning and domain adaptation. By embedding OT-based losses into neural networks, their method enables end-to-end training while explicitly accounting for the geometric structure of the output space. Through an entropically regularized formulation, the proposed Wasserstein loss can be efficiently optimized via gradient-based methods, allowing effective alignment of feature distributions across domains and demonstrating the feasibility of OT within deep learning frameworks.

Chapter 4

Distributional Preference Modeling for Cross-Domain Recommendation

4.1   Background

Recommender systems have achieved remarkable success in a wide range of real-world applications, such as e-commerce, online media platforms, and social networks. However, despite their effectiveness, many existing recommender systems continue to face fundamental challenges, including data sparsity and the cold-start problem, particularly when user – item interactions are limited or unevenly distributed. These challenges often lead to suboptimal recommendation performance, as traditional models struggle to accurately capture user preferences under such constraints.

Most traditional recommender systems model user preferences as fixed point embeddings in a latent space. While such representations are computationally efficient, they implicitly assume that a user's interests can be captured by a single deterministic vector. This assumption overlooks two important characteristics of real user behavior. First, user preferences are inherently uncertain, especially in sparse settings where limited observations are available. Second, user interests are often multi-faceted, spanning multiple latent aspects that cannot be adequately represented by a single point estimate. As a result, point-based representations may fail to capture the diversity and ambiguity of user preferences, leading to suboptimal recommendation performance.

Cross-domain recommendation (CDR) aims to alleviate cold-start and data sparsity issues by transferring knowledge from a data-rich source domain to a data-sparse target domain. A common strategy adopted by existing CDR methods is to exploit overlapping users or items between domains as explicit bridges for knowledge transfer. Through shared embeddings, adversarial alignment, or joint training objectives, these methods assume that at least part of the user or item space is shared across domains during training. However, such assumptions are often unrealistic in practical scenarios. In many real-world applications, user identities or item catalogs are domain-specific due to privacy constraints, platform isolation, or delayed synchronization, resulting in entirely non-overlapping users and items across domains during the training phase.

In the absence of overlapping entities, effective knowledge transfer across domains becomes sub-

stantially more challenging. Without explicit correspondences, aligning latent representations across domains requires modeling higher-level structural or distributional similarities rather than instance-level matches. Existing non-overlapping CDR methods typically rely on shared latent spaces or distribution matching techniques, but most of them still represent user preferences as discrete vectors, limiting their expressiveness and robustness under severe data sparsity.

Motivated by these limitations, we propose to model user preferences from a distributional perspective and perform cross-domain knowledge transfer at the distribution level. Specifically, we introduce **DUP-OT** (Distributional User Preference with Optimal Transport), a novel framework for cross-domain recommendation under strictly non-overlapping settings. The core idea of DUP-OT is to represent each user's preference as a probability distribution over latent preference components, rather than a single point embedding. This distributional representation explicitly captures both the uncertainty and the multi-aspect nature of user interests.

To enable effective cross-domain alignment between such distributional preferences, DUP-OT leverages optimal transport (OT) theory as a principled mechanism for measuring and aligning probability distributions across domains. By aligning distributional representations in a shared latent space, OT allows preference knowledge to be transferred from the source domain to the target domain without requiring any overlapping users or items during training. This design enables DUP-OT to bridge domain gaps at a structural level, making it particularly suitable for realistic cross-domain recommendation scenarios where explicit correspondences are unavailable.

In the following sections, we introduce the detailed architecture of DUP-OT, including its shared preprocessing stage for constructing a unified latent space, the distributional user preference modeling module based on Gaussian Mixture Models, and the optimal-transport-based alignment mechanism for cross-domain preference transfer.

4.2   Methods

4.2.1   Overview

The overall architecture of the proposed DUP-OT framework is illustrated in Figure 4.1. DUP-OT is designed for cross-domain recommendation under strictly non-overlapping settings and consists of three main stages: (1) Shared Preprocessing Stage, (2) User GMM Weights Learning Stage, and (3) Cross-Domain Rating Prediction Stage. These stages jointly enable distributional user preference modeling and cross-domain knowledge transfer via optimal transport.

The scenario setup of DUP-OT involves two domains: a source domain $\mathcal{D}_S$ and a target domain $\mathcal{D}_T$. Each domain contains its own set of users and items, with **no overlapping users or items** between the two domains during training. The training set of source domain should happen before the valid and test set of target domain in time to avoid information leakage. The goal of DUP-OT is to leverage the abundant user – item interaction data in the source domain to enhance recommendation performance in the target domain, particularly under data sparsity conditions.

A core design principle of DUP-OT is to model user preferences as probability distributions rather than point embeddings. Specifically, we represent each user's preference as a Gaussian Mixture Model (GMM) in a shared latent space across both source and target domains. To make distribution-level alignment computationally feasible, we introduce the following assumption: within each domain, all
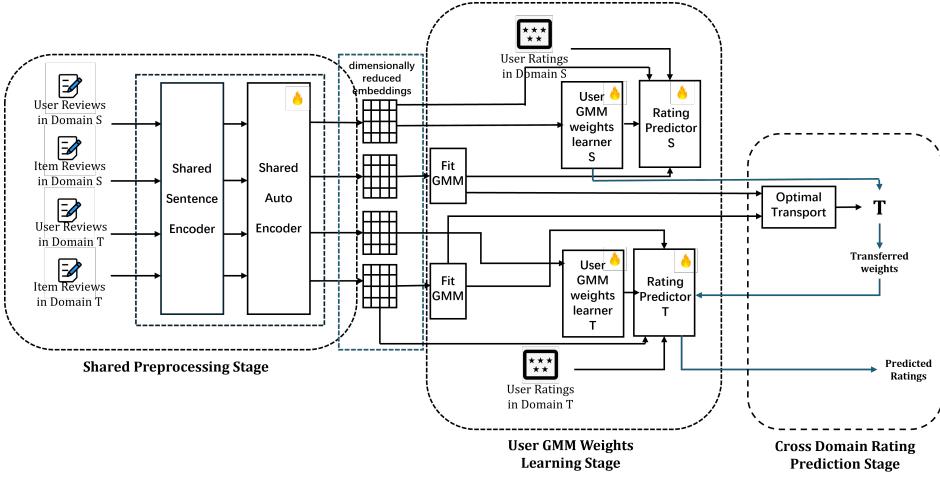
Figure 4.1: Architecture of the DUP-OT Framework

users share a fixed set of GMM components, while only the mixture weights vary across users. The shared components capture the main latent preference aspects of the domain, whereas the user-specific weights reflect individual preferences over these aspects. This assumption is reasonable in practice, as users within the same domain often exhibit similar underlying preference structures, and it allows us to perform cross-domain alignment at the component level instead of the instance level.

The Shared Preprocessing Stage aims to construct a unified latent space for both domains. Given user − item interaction data accompanied by review texts, ratings, and timestamps, we first extract semantic features from review texts using a shared pre-trained BERT model. User and item embeddings are obtained by aggregating review-level features, where a time-decay function is applied to assign higher weights to more recent reviews during user embedding aggregation. As for item embeddings, we simply average all associated review embeddings. This preprocessing pipeline is shared across domains to ensure consistency in representation. Since the resulting embeddings are high-dimensional, a shared autoencoder is trained on data from both domains to reduce dimensionality and produce compact embeddings in a common latent space, which serves as the basis for subsequent preference modeling.

Based on the reduced item embeddings, we then determine the fixed GMM components for each domain. Specifically, a Gaussian Mixture Model is fitted to all item embeddings within a domain, and the learned mixture components are treated as domain-level latent preference aspects. This design is motivated by the observation that items in a domain naturally reflect its major semantic and preference dimensions.

In the User GMM Weights Learning Stage, DUP-OT learns personalized preference distributions for individual users. For each domain, we train a user-specific GMM weight learner and a rating prediction model using only data from that domain. The weight learner, implemented as a multi-layer perceptron (MLP), maps a user's reduced embedding to a set of mixture weights over the fixed GMM components. These weights define the user's preference distribution. A separate rating prediction

MLP then estimates user – item ratings based on the weighted Mahalanobis distances between item embeddings and the GMM components. This stage produces expressive distributional representations of user preferences within each domain.

Finally, the Cross-Domain Rating Prediction Stage enables knowledge transfer from the source domain to the target domain via optimal transport. Since GMM components are fixed within each domain, we formulate optimal transport at the component level to align source-domain and target-domain GMMs. The resulting transport plan specifies how preference mass should be transferred between components across domains. Using this transport plan, user-specific GMM weights learned in the source domain can be mapped to the target domain, yielding adapted preference distributions. These transferred distributions are optionally fused with original target-domain user distributions to enhance preference modeling. The final user preference distributions are then used by the target-domain rating prediction model to generate improved rating predictions.

4.2.2   Shared Preprocessing Stage

The Shared Preprocessing Stage aims to extract consistent user and item embeddings from raw review data across both source and target domains. This stage consists of three main steps: (1) Review Text Embedding, (2) User and Item Embedding Aggregation, and (3) Dimensionality Reduction via Autoencoder. As our settings involve two domains with entirely non-overlapping users and items, it is crucial to ensure that the extracted embeddings are comparable and lie in a shared latent space. Also, the potential connections between the two domains can only be established through semantic similarities, so leveraging review texts is essential for capturing meaningful representations.

4.2.2.1   Review Text Embedding

To extract semantic features from review texts, we utilize a shared pre-trained BERT model across both domains. Specifically, we are using all-MiniMLM-L6-v2 model from Sentence-Transformers library [63], which is a lightweight variant of BERT optimized for generating sentence embeddings. Given a review text, we tokenize it and feed it into the BERT model to obtain a fixed-length embedding vector that captures its semantic content. By using a shared BERT model, we ensure that review embeddings from both domains are generated in the same semantic space, facilitating cross-domain alignment later on.

4.2.2.2   User and Item Embedding Aggregation

Usually, the recent reviews of a user are more indicative of its current preferences or characteristics. To account for this temporal aspect, we apply a time-decay function when aggregating review embeddings into user embeddings. For each user, we aggregate review-level embeddings into an initial user representation using a time-aware weighted pooling strategy.

Given a set of reviews with timestamps $\{t_i\}_{i=1}^N$ and corresponding embeddings $\{\mathbf{e}_i\}_{i=1}^N$, we define the reference time as the most recent review timestamp $t_{\mathrm{ref}} = \max_i t_i$. The temporal distance of each review is measured in months and truncated by a maximum value :

$$\Delta_i = \min\left(\frac{t_{\mathrm{ref}} - t_i}{T}, \Delta_{\max}\right),$$

where $T = 30 \times 86400$. Each review is assigned an exponentially decayed weight:

$$w_i = \exp(-\lambda \Delta_i),$$

which is further normalized as $\tilde{w}_i = w_i / \sum_j w_j$. The final user embedding is computed as a weighted sum of review embeddings:

$$\mathbf{u} = \sum_{i=1}^{N} \tilde{w}_i \mathbf{e}_i.$$

For item embedding aggregation, we assume that item attributes are relatively stable over time. Therefore, we simply compute the item representation by averaging all associated review embeddings. Formally, the item embedding is computed as

$$\mathbf{v} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{e}_i,$$

where $\{\mathbf{e}_i\}$ are the embeddings of reviews associated with the item.

An example of item embeddings after aggregation is illustrated in Figure 4.2. It can be observed that item embeddings from different domains exhibit distinct distributional patterns, indicating a substantial domain discrepancy. Addressing this discrepancy is critical for effective cross-domain recommendation and constitutes the main focus of the following sections.
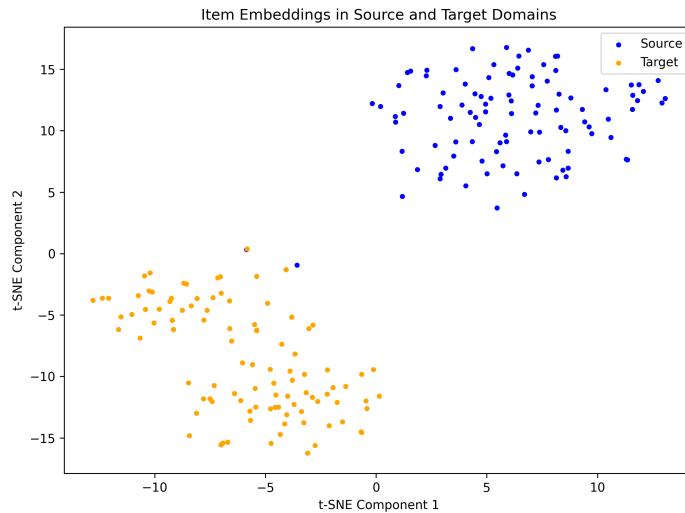

Figure 4.2: Item Embeddings after Aggregation

4.2.2.3   Dimensionality Reduction via Autoencoder

The aggregated user and item embeddings are typically high-dimensional, which can lead to increased computational costs and potential overfitting in subsequent modeling stages. To address this issue, we employ a shared autoencoder to reduce the dimensionality of embeddings from both domains. The autoencoder consists of an encoder network that maps input embeddings to a lower-dimensional latent space and a decoder network that reconstructs the original embeddings from the latent representations. The structure of the autoencoder is illustrated in Figure 4.3.
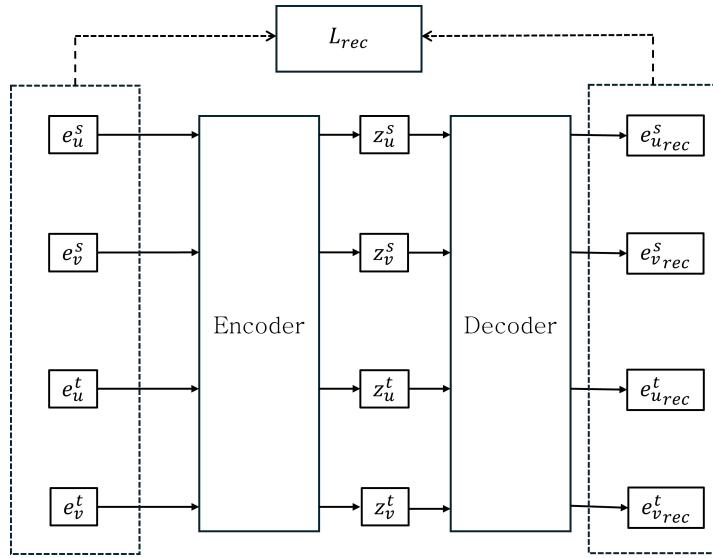
Figure 4.3: Autoencoder Structure

The autoencoder is trained on a combined dataset of user and item embeddings from both source and target domains. The training objective is to minimize the reconstruction loss, defined as the mean squared error between the original embeddings and their reconstructions. By sharing the autoencoder across domains, we ensure that the resulting reduced embeddings lie in a common latent space, which is essential for following distributional preference modeling and cross-domain alignment.

After training, we obtain reduced user and item embeddings by passing the original embeddings through the encoder network. These reduced embeddings serve as the basis for subsequent GMM-based preference modeling and optimal transport alignment in the DUP-OT framework.

4.2.3   User GMM Weights Learning Stage

In the User GMM Weights Learning Stage, we aim to learn expressive distributional representations of user preferences in the shared latent space constructed by the Shared Preprocessing Stage. Specifically, each user's preference is modeled as a Gaussian Mixture Model (GMM), which enables the representation of multi-aspect user interests in a probabilistic manner. This stage consists of two key components:

(1) determining a fixed set of domain-level GMM components, and (2) learning user-specific mixture weights together with a corresponding rating prediction model.

4.2.3.1   Fixed GMM Component Determination

To capture the major latent preference aspects within each domain, we first construct a domain-level Gaussian Mixture Model (GMM) by fitting it to the reduced item embeddings obtained from the Shared Preprocessing Stage. The resulting GMM components are treated as fixed and shared by all users within the same domain, serving as latent preference components for subsequent user-specific preference modeling.

To build intuition for Gaussian Mixture Models, we first present a one-dimensional (1D) example in Figure 4.4, where the roles of individual Gaussian components and mixture weights can be easily visualized. We then illustrate a two-dimensional (2D) GMM in Figure 4.5 to demonstrate how mixture models generalize from scalar to vector-valued representations.
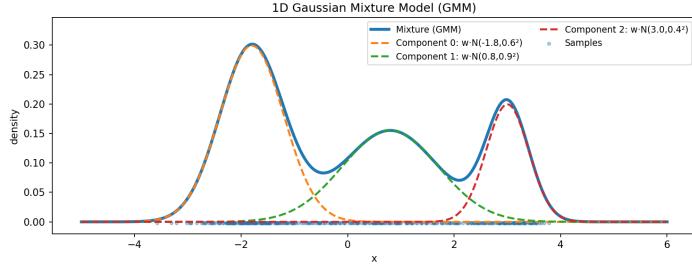
Figure 4.4: 1D Gaussian Mixture Model Example

To this end, we adopt the `BayesianGaussianMixture` implementation from the scikit-learn library [64] to fit GMMs on item embeddings. Unlike conventional GMMs that require manually specifying the number of mixture components, this approach is based on variational Bayesian inference with a Dirichlet prior over mixture weights. Such a formulation allows mixture components that are not sufficiently supported by the data to be automatically suppressed during training, enabling the effective number of active components to be inferred directly from the data.

Formally, given a set of item embeddings $\{\mathbf{x}_n\}_{n=1}^N$, the Bayesian Gaussian Mixture Model assumes the following generative process:

$$\boldsymbol{\pi} \sim \mathrm{Dirichlet}(\boldsymbol{\alpha}), \qquad z_n \sim \mathrm{Categorical}(\boldsymbol{\pi}), \qquad \mathbf{x}_n \mid z_n = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where $\boldsymbol{\pi}$ denotes the mixture weights, z_n is the latent component assignment, and $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ represent the mean and covariance of the k-th Gaussian component.

Exact posterior inference in this model is intractable. Therefore, variational Bayesian inference is employed by maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] - \mathbb{E}_q[\log q(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})],$$

where $q(\cdot)$ denotes the variational posterior distribution.
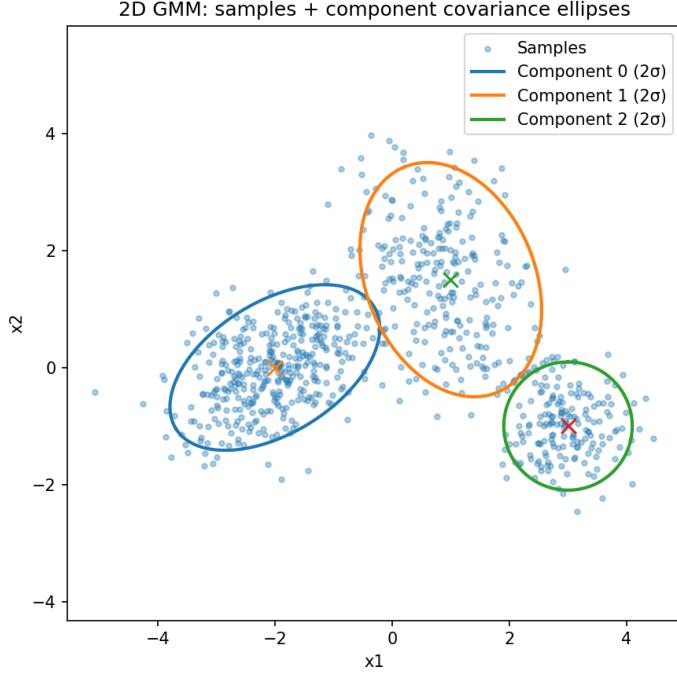
Figure 4.5: 2D Gaussian Mixture Model Example

Due to the Dirichlet prior imposed on the mixture weights, components that are weakly supported by the data are assigned negligible posterior mass, effectively pruning redundant components. As a result, the Bayesian GMM provides a principled and flexible mechanism for determining the effective number of domain-level latent preference components.

4.2.3.2   User-Specific GMM Weight Learning and Rating Prediction

With the fixed GMM components determined for each domain, we proceed to learn user-specific mixture weights that characterize individual preference distributions in the shared latent space. For each domain, DUP-OT employs two neural modules trained solely on the data from that domain: a user-specific GMM weight learner and a rating prediction network. These two modules are optimized jointly to ensure that the learned preference distributions are directly aligned with the rating prediction objective.

User-specific GMM weight learning. Let $\mathbf{u}_i \in \mathbb{R}^d$ denote the reduced embedding of user i obtained from the Shared Preprocessing Stage. Given the fixed set of domain-level GMM components $\{(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}_{k=1}^{K}$, we learn personalized mixture weights via a multi-layer perceptron (MLP):

$$\mathbf{s}_i = f_\theta(\mathbf{u}_i) \in \mathbb{R}^K, \qquad \boldsymbol{\pi}_i = \mathrm{softmax}(\mathbf{s}_i),$$

where $\boldsymbol{\pi}_i = [\pi_{i1}, \ldots, \pi_{iK}]^\top$ represents the user-specific mixture weights satisfying $\sum_{k=1}^{K} \pi_{ik} = 1$ and $\pi_{ik} \geq 0$. The resulting weights define a personalized preference distribution over the domain-level latent preference components.

Distribution-aware user–item representation.   For an item j with reduced embedding $\mathbf{v}_j \in \mathbb{R}^d$, we compute its Mahalanobis distance to each GMM component:

$$d_{ijk} = \sqrt{(\mathbf{v}_j - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{v}_j - \boldsymbol{\mu}_k)}.$$

These component-wise distances are combined with the user-specific mixture weights to construct a distribution-aware representation of user–item compatibility:

$$\mathbf{h}_{ij} = [\pi_{i1}d_{ij1}, \ \pi_{i2}d_{ij2}, \ \ldots, \ \pi_{iK}d_{ijK}]^\top \in \mathbb{R}^K.$$

This formulation enables user preferences and item representations to be compared in a distribution-aware manner by jointly considering latent preference components and their personalized importance.

Rating prediction and joint training.   A separate MLP is employed to predict the rating for a given user–item pair:

$$\hat{r}_{ij} = g_\phi(\mathbf{h}_{ij}),$$

where $g_\phi(\cdot)$ denotes the rating prediction network. The GMM weight learner f_θ and the rating prediction network g_ϕ are trained jointly in a supervised fashion by minimizing the rating prediction loss over the observed interactions $\mathcal{D}$:

$$\mathcal{L}_{\text{rate}}(\theta, \phi) = \frac{1}{|\mathcal{D}|} \sum_{(i,j,r_{ij}) \in \mathcal{D}} (\hat{r}_{ij} - r_{ij})^2,$$

where r_{ij} and $\hat{r}_{ij}$ denote the ground-truth and predicted ratings, respectively. Through joint optimization, the learned user-specific preference distributions are directly tailored to the recommendation task.

4.2.4   Cross-Domain Rating Prediction Stage

In the Cross-Domain Rating Prediction Stage, we leverage optimal transport (OT) theory to facilitate knowledge transfer from the source domain to the target domain at the distribution level. It is important to emphasize that the non-overlapping constraint in our setting is imposed strictly during the training stage: no overlapping users or items are exploited when learning user preference models or cross-domain alignments. During inference and evaluation, however, users may appear in both domains, which reflects realistic deployment scenarios and allows transferred knowledge to be utilized in a principled manner. This design is reasonable because, in many real-world applications, cross-domain user correspondences are unavailable or restricted during training due to privacy concerns, system isolation, or data access limitations, while such correspondences may become observable at deployment or evaluation time.

Based on the User GMM Weights Learning Stage, each user's preference in both the source and target domains is represented as a Gaussian Mixture Model (GMM) defined over fixed, domain-level

latent preference components. Given these distributional representations, we formulate cross-domain knowledge transfer as an optimal transport problem between the source-domain and target-domain GMM component sets. Since all users within a domain share the same set of GMM components, OT is performed at the component level rather than the instance level, enabling efficient and stable alignment of latent preference aspects across domains.

Optimal transport formulation and cost matrix.   Formally, let $\boldsymbol{\pi}^s \in \mathbb{R}_+^{K_s}$ and $\boldsymbol{\pi}^t \in \mathbb{R}_+^{K_t}$ denote the mixture weight vectors over the fixed GMM components in the source and target domains, respectively. We define a cost matrix $\mathbf{C} \in \mathbb{R}^{K_s \times K_t}$, where each entry C_{kl} measures the dissimilarity between the k-th source-domain component and the l-th target-domain component.

Specifically, the cost matrix is constructed using the 2-Wasserstein distance between Gaussian distributions. For the k-th source-domain component $\mathcal{N}(\boldsymbol{\mu}_k^s, \boldsymbol{\Sigma}_k^s)$ and the l-th target-domain component $\mathcal{N}(\boldsymbol{\mu}_l^t, \boldsymbol{\Sigma}_l^t)$, the cost is defined as:

$$C_{kl} = \|\boldsymbol{\mu}_k^s - \boldsymbol{\mu}_l^t\|_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}_k^s + \boldsymbol{\Sigma}_l^t - 2\left((\boldsymbol{\Sigma}_k^s)^{1/2}\boldsymbol{\Sigma}_l^t(\boldsymbol{\Sigma}_k^s)^{1/2}\right)^{1/2}\right).$$

Figure 4.6 provides an intuitive illustration of how the cost matrix is constructed by computing pairwise Wasserstein distances between GMM components in the source and target domains. Each entry in the cost matrix corresponds to the transport cost between a pair of latent preference components, serving as the fundamental unit for subsequent OT alignment.
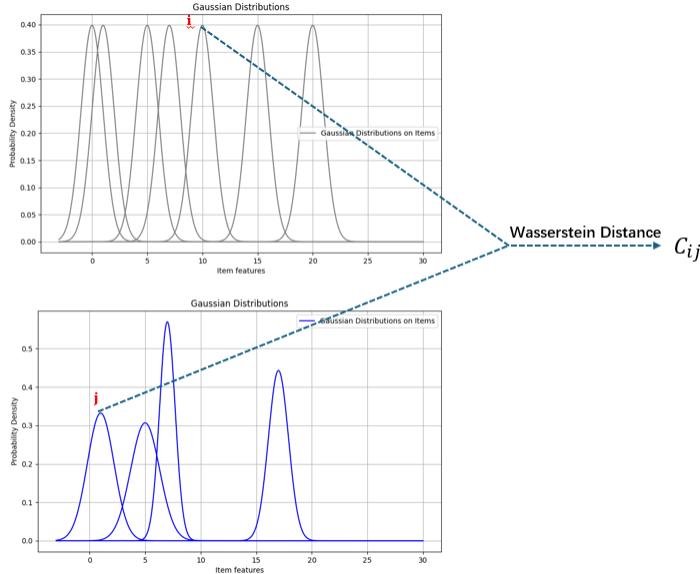

Figure 4.6: Cost Matrix Construction via Pairwise Wasserstein Distances

Given the cost matrix, the optimal transport plan $\mathbf{T}$ is obtained by solving:

$$\min_{\mathbf{T} \geq 0} \langle \mathbf{T}, \mathbf{C} \rangle \quad \text{s.t.} \quad \mathbf{T1} = \boldsymbol{\pi}^s, \; \mathbf{T}^\top \mathbf{1} = \boldsymbol{\pi}^t.$$

When no regularization is applied, this formulation corresponds to the Earth Mover's Distance (EMD), which can be solved as a linear programming problem to obtain an exact and typically sparse transport plan. Alternatively, we adopt the entropically regularized OT formulation:

$$\min_{\mathbf{T} \geq 0} \langle \mathbf{T}, \mathbf{C} \rangle + \varepsilon \sum_{k,l} T_{kl}(\log T_{kl} - 1),$$

which can be efficiently solved using the Sinkhorn algorithm. The entropy regularization yields a smooth and dense transport plan while significantly reducing computational cost. In our setting, OT is performed at the component level, where the number of GMM components is relatively small. Therefore, both EMD and Sinkhorn-based OT are computationally feasible. In practice, we primarily adopt the Sinkhorn algorithm for efficiency and numerical stability, while EMD is used for analysis when an exact transport plan is desired.

Distribution transfer and fusion. Using the learned optimal transport plan, we transfer user-specific mixture weights from the source domain to the target domain as:

$$\tilde{\boldsymbol{\pi}}_i^t = \mathbf{T}^\top \boldsymbol{\pi}_i^s.$$

At inference time, we construct the final target-domain preference distribution via a linear fusion strategy:
$$\boldsymbol{\pi}_i^{\text{final}} = \lambda_i \boldsymbol{\pi}_i^t + (1 - \lambda_i)\tilde{\boldsymbol{\pi}}_i^t,$$
where $\lambda_i \in [0,1]$ controls the contribution of the original target-domain distribution. If a user has interactions in both domains, $\lambda_i \in (0,1)$ is used to combine domain-specific evidence with transferred knowledge. If a user has interactions only in the source domain, $\lambda_i = 0$ and the final distribution relies solely on the transferred preference. If a user has interactions only in the target domain, $\lambda_i = 1$ and the original target-domain distribution is retained.

Finally, the fused user preference distributions are fed into the target-domain rating prediction model to generate enhanced rating predictions. Through this training-time non-overlapping and inference-time adaptive fusion design, DUP-OT faithfully reflects practical cross-domain recommendation scenarios while maintaining a strict non-overlapping assumption during model learning.

Chapter 5

Experiments

In this chapter, we present the detailed setup of our experiments to evaluate the proposed **DUP-OT** framework for non-overlapping cross-domain recommendation training. Specifically, we first proposed the research questions we want to solve by conducting experiments, and then we'll describe the datasets we used and how we preprocessed them, the experiments we conducted, including the baseline models we compared against, the evaluation metrics we employed. Finally, we outline the implementation details of our proposed method, including hyperparameter settings and training procedures to ensure reproducibility.

The final results and analyses will be presented in the next chapter.

5.1   Research Questions

As the proposed **DUP-OT** framework introduces two key components: (1) distribution-based user preference modeling using GMMs, and (2) cross-domain preference transfer via optimal transport, we want to design experiments to figure out the effects of these two components respectively. Moreover, we also want to compare our proposed method with existing non-overlapping cross-domain recommendation methods. Thus, the research questions we aim to answer through our experiments can be summarized as follows:

- **RQ1:** Does introducing cross-domain information enhance the recommendation performance in the target domain?

- **RQ2:** Does modeling user preferences as distributions, rather than vectors, lead to improved recommendation performance?

- **RQ3:** How does the performance of our proposed method compare with that of existing non-overlapping CDR models?

To answer **RQ1**, we conduct an ablation study comparing target-domain recommendation performance with and without leveraging source-domain information. When we do not use source-domain

data, our model essentially reduces to a single-domain recommendation model that employs GMMs for user preference modeling.

To address **RQ2**, we evaluate our target-domain model without source-domain information against some single-domain baseline models that represent user preferences as vectors. The idea is, if we don't use source-domain data, the target-domain model will reduce to a single-domain recommendation model, and thus the only difference between our model and the baselines will be the way of user preference modeling. Thus we can isolate the effect of distribution-based user preference modeling.

Finally, to answer **RQ3**, we simply compare our full model with representative non-overlapping CDR baseline models in the same experimental settings.

5.2   Datasets and Preprocessing

We set our experiment under an e-commerce scenario, where different product categories are treated as different domains. For example, Electronics and Video Games are two different domains, and our target is to leverage user interactions in the source domain (e.g., Video Games) to improve recommendation performance in the target domain (e.g., Electronics). To this end, we choose the Amazon Review dataset [65] for our experiments, specifically, we choose the 5-core version of four domains: Electronics, Digital Music, Movies & TV, and Video Games. To make sure the 5-core really holds in these four datasets, we further filter the datasets to ensure that each user and item has at least 5 interactions. After preprocessing, the statistics of the four datasets are summarized in Table 5.1.

Table 5.1: Statistics of the Amazon Review datasets after preprocessing.

Domain	Users	Items	Interactions	Density (%)
Electronics	728,489	159,729	6,737,580	0.0058
Digital Music	16,252	11,269	166,942	0.0912
Movies & TV	297,377	59,925	3,408,612	0.0191
Video Games	55,144	17,286	496,904	0.0521

Considering the interaction density and the relative time stamps of the four domains, we select Electronics as the target domain, and the other three domains as source domains in our experiments.

Although our proposed method is not a sequential recommendation model, we think it's still important to avoid data leakage when splitting the datasets into training, validation, and test sets. So we sort the interactions of each user by their time stamps, and then split target-domain interactions, in our case, Electronics, into training, validation, and test sets globally by the ratio of 80%, 10%, and 10%. Then to avoid data leakage, we need to ensure the interactions in the source domain used for training happened before the earliest interaction in the target-domain validation set. To this end, we first find the earliest time stamp t_{val} in the target-domain validation set, and then filter the source-domain interactions to only keep those happened before t_{val}. To reduce computational costs, we search for t_{val} in a step of 5% in the time-stamp sorted source-domain interactions. We don't need to split the source-domain interactions into training, validation, and test sets, as we only use source-domain data for training in our experiments.

As we split the interactions of target-domain globally, it's possible that some users only appear in the validation or test sets. When we build the Dataset object for training, we just randomly set the embeddings of the users who only appear in the validation or test sets as random vectors.

5.3   Experimental Setup

To answer the research questions proposed above, we mainly design three groups of experiments corresponding to **RQ1**, **RQ2**, and **RQ3** respectively. All experiments are conducted on the preprocessed Amazon Review datasets described in the previous section, and the target domain is always Electronics while the source domain varies among Digital Music, Movies & TV, and Video Games. In other words, we conduct experiments on three source-target domain pairs:

- Digital Music → Electronics

- Movies & TV → Electronics

- Video Games → Electronics

To answer **RQ1**, which investigates the effect of cross-domain information, we compare two variants of our proposed framework: DUP-OT (w/ source) and DUP-OT (w/o source), which means we either leverage source-domain information for target-domain recommendation or not. These two variants share identical architectures, representation learning strategies, and training protocols. The only difference lies in whether source-domain preference distributions are transferred to the target domain via optimal transport in the last stage of our framework. Therefore, performance differences can be directly attributed to the effect of cross-domain information.

To address **RQ2**, which examines the effect of distribution-based user preference modeling, we compare DUP-OT (w/o source) with representative single-domain recommendation models, namely LightGCN [24] and NeuMF [17]. In this setting, DUP-OT (w/o source) does not leverage any source-domain interactions and is trained exclusively on the target-domain training set. Despite this, the only difference between DUP-OT (w/o source) and the baselines is the way of user preference modeling. Our model represents user preferences as GMMs, while the baselines use point embeddings. Thus, performance differences can be directly attributed to the effect of distribution-based user preference modeling.

Finally, to answer **RQ3**, which compares our proposed method with existing non-overlapping CDR models, we select a representative non-overlapping CDR baseline model, TDAR [66], and compare it with our full model, DUP-OT (w/ source), in the same experimental settings.

5.3.1   Baseline Models

As discussed in the previous section, we select different baseline models to answer the proposed research questions from complementary perspectives. In particular, the selected baselines cover both single-domain recommendation models and representative non-overlapping cross-domain recommendation approaches. In total, we consider three baseline models and two variants of our proposed framework in the experiments, which are described as follows:

- **LightGCN** [24]: LightGCN is a representative graph-based collaborative filtering model that learns user and item representations by propagating embeddings on the user–item interaction graph. Unlike earlier GCN-based recommenders, LightGCN removes feature transformation and non-linear activation, and retains only neighborhood aggregation, resulting in a simple yet effective architecture. As a strong single-domain baseline, LightGCN represents user preferences as deterministic embedding vectors and relies solely on target-domain interaction data. We include LightGCN to evaluate whether distribution-based user preference modeling provides advantages over state-of-the-art vector-based graph collaborative filtering methods.

- **NeuMF** [17]: NeuMF is a classical neural collaborative filtering model that unifies matrix factorization and multi-layer perceptrons under a neural framework. By replacing the inner product with a non-linear interaction function, NeuMF aims to capture complex user–item interaction patterns in the latent space. Similar to LightGCN, NeuMF represents users and items as point-valued latent vectors and operates in a purely single-domain setting. This baseline is adopted to compare our distribution-based user preference modeling with a widely used neural vector-based recommendation paradigm.

- **TDAR** [66]: TDAR is a representative non-overlapping cross-domain recommendation model based on text-enhanced domain adaptation. In the absence of shared users or items across domains, TDAR leverages domain-invariant textual features extracted from user reviews as anchor points to align latent spaces via adversarial training. By transferring distribution patterns learned from a dense source domain to a sparse target domain, TDAR alleviates data sparsity in non-overlapping scenarios. We include TDAR as a strong non-overlapping CDR baseline to compare our optimal transport-based preference transfer mechanism with existing text-driven domain adaptation approaches.

- **DUP-OT (w/ source)**: This variant corresponds to the full version of our proposed **DUP-OT** framework. It models user preferences as Gaussian mixture distributions and transfers such distributional preferences from the source domain to the target domain via optimal transport. This setting evaluates the complete effectiveness of distribution-based preference modeling combined with cross-domain knowledge transfer.

- **DUP-OT (w/o source)**: This variant disables the cross-domain transfer component and trains the model using only target-domain data. Under this setting, DUP-OT degenerates into a single-domain recommendation model that still represents user preferences as probability distributions. By comparing this variant with vector-based single-domain baselines, we can isolate and analyze the impact of distribution-based user preference modeling independent of cross-domain information.

5.3.2   Evaluation Metrics

To quantitatively assess the recommendation performance of different models, we formulate the recommendation task as a *rating prediction* problem and employ two widely used evaluation metrics: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

We adopt rating prediction as the evaluation task for two main reasons. First, rating prediction provides a fine-grained and direct measure of how accurately a model captures user preferences, which is particularly important in cross-domain recommendation scenarios where the goal is to transfer preference knowledge from a source domain to a target domain. Second, in non-overlapping cross-domain recommendation settings, ranking-based evaluation protocols often rely on strong assumptions about negative sampling and candidate item sets, which may introduce additional bias. In contrast, rating prediction allows us to evaluate model performance on observed user–item interactions without requiring assumptions about unobserved entries.

Following prior cross-domain recommendation studies that focus on rating prediction under non-overlapping settings, we adopt RMSE and MAE as our primary evaluation metrics. Both metrics have been widely used in existing cross-domain recommendation work and provide complementary perspectives on prediction accuracy. RMSE penalizes large prediction errors more heavily and is therefore sensitive to outliers, while MAE measures the average absolute deviation and offers a more robust assessment of overall prediction quality.

Formally, given a set of test user–item interactions $\mathcal{D}_{test} = \{(u, i, r_{ui})\}$, where r_{ui} denotes the ground-truth rating of user u on item i, the RMSE and MAE are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{D}_{test}|} \sum_{(u,i,r_{ui}) \in \mathcal{D}_{test}} (\hat{r}_{ui} - r_{ui})^2} \tag{5.1}$$

$$\text{MAE} = \frac{1}{|\mathcal{D}_{test}|} \sum_{(u,i,r_{ui}) \in \mathcal{D}_{test}} |\hat{r}_{ui} - r_{ui}| \tag{5.2}$$

where $\hat{r}_{ui}$ denotes the predicted rating generated by a recommendation model. Lower values of RMSE and MAE indicate better recommendation accuracy.

5.4   Implementation Details

In this section, we provide implementation details of our proposed **DUP-OT** framework to ensure reproducibility of our experimental results.

We implement our model using PyTorch and conduct experiments on a machine with an NVIDIA RTX 3080 GPU. To retrieve user and item embeddings from the interaction data, we utilize a pre-trained BERT model to encode from review texts. Specifically, we use the `all-MiniLM-L6-v2` variant from the Sentence Transformers library [63], which produces 384-dimensional embeddings.

To reduce computational costs, we trained an auto-encoder to compress the 384-dimensional BERT embeddings into 128-dimensional vectors. Specifically, to ensure both domains' embeddings remain in the same latent space, we train the auto-encoder on both source and target domain user and item embeddings together. The auto-encoder consists of an encoder with two linear layers ($384 \rightarrow 256 \rightarrow 128$), and a decoder with two linear layers ($128 \rightarrow 256 \rightarrow 384$). And we use ReLU as the activation function between layers. For training the auto-encoder, we use the Adam optimizer with a learning rate of 0.0002 and a batch size of 128. We set the number of training epochs to 500 and use Mean Squared Error (MSE) as the reconstruction loss.

In the proposed **DUP-OT** framework, we fit Gaussian Mixture Models (GMMs) on item embeddings separately for each domain in order to capture the latent multi-modal structure of item representations.

Specifically, we employ the `BayesianGaussianMixture` class from the `sklearn.mixture` module in Scikit-learn [67]. For each domain, the maximum number of mixture components is set to the square root of the number of items in that domain, with a lower bound of 8 and an upper bound of 128. This heuristic provides sufficient modeling capacity while avoiding excessive complexity in domains with large item sets. The final effective number of components is inferred automatically through variational Bayesian inference.

We set the maximum number of iterations to 300 and adopt a diagonal covariance structure (`covariance_type = 'diag'`), which offers a good trade-off between expressive power and computational efficiency. Diagonal covariances are sufficient to capture variations along individual embedding dimensions while maintaining numerical stability and scalability.

To further ensure stable optimization, we add a small regularization term (`reg_covar = 1e-5`) to the diagonal of each covariance matrix. The convergence tolerance is set to `tol = 1e-3`. We use the default setting for the weight concentration prior, and fix `random_state = 42` to ensure reproducibility. Verbose output is disabled during model fitting.

After obtaining the Gaussian Mixture Models for both the source and target domains, we compute a pairwise Wasserstein distance matrix between the Gaussian components across domains. This distance matrix serves as the cost matrix for cross-domain alignment.

Based on this cost matrix, we formulate the optimal transport problem between source-domain and target-domain preference distributions. The optimal transport plan is computed using the Python Optimal Transport (POT) library [68].

In the next stage of our framework, we need to train two MLPs in each domain, one for learning the user preference weights over GMM components, and the other for predicting ratings based on user preference distributions and item embeddings.

We train each domain's MLPs separately using the Adam optimizer with a learning rate of 0.0003 and a batch size of 256. To make the final result more convincing, we repeated each experiment 5 times with different random seeds and report the average performance. The random seed is set to 7, 11, 17, 33, 2025 for different runs. The total epochs for training are set to 50 for both source and target domains. For the MLP architecture, both MLPs consist of three linear layers with ReLU activation functions between layers. The first MLP takes dimensional-reduced user embeddings as input and outputs preference weights for each GMM component in the corresponding domain. The shape of the first MLP is (128, (128 + number of GMM components)//2, number of GMM components). The second MLP takes as input the Mahalanobis distances between the item embedding and each Gaussian component, weighted by the corresponding user-specific mixture weights, together with a processed timestamp feature, and outputs the predicted rating. The shape of the second MLP is (number of GMM components + 1, (number of GMM components + 1)//2, 1).

For the timestamp feature, we transform the original interaction time into a normalized recency-aware scalar to capture temporal effects in user behavior. Specifically, for each domain, we use the maximum timestamp observed in the training set as a reference time. Given an interaction occurring at time t, we first compute its temporal distance to the reference time in units of months. To avoid extreme values caused by very old interactions, this temporal distance is capped at a maximum of 36

months.

We then apply an exponential decay function to model temporal recency, where the decay rate is chosen such that the weight decays to 0.5 after six months. Formally, the decayed time value is computed as $\exp(-\lambda \Delta t)$, where Δt denotes the clipped month-level time difference and λ is the decay coefficient. Finally, the resulting value is passed through a squashing transformation $x/(1+x)$ to obtain a bounded and smooth timestamp feature.

The same reference time derived from the training set is used to process validation and test interactions, ensuring temporal consistency and preventing information leakage. The processed timestamp feature is then concatenated with the distance-based representation and fed into the rating prediction MLP.

Finally, in the test phase, we consider different availability patterns of user preference information across the source and target domains. Specifically, test users are categorized into the following three cases:

- Users who appear only in the target-domain test set.

- Users who appear only in the source-domain test set.

- Users who appear in both the source-domain and target-domain test sets.

For users who appear only in the target-domain test set, we directly use the trained target-domain model to infer their preference weights over the target GMM components and make rating predictions accordingly.

For users who appear only in the source-domain test set, we first use the trained source-domain model to infer their preference weights over the source-domain GMM components. These source-domain mixture weights are then mapped to the target domain via the learned optimal transport plan, yielding transferred preference weights on the target-domain GMM components. The transferred weights are finally used by the target-domain rating predictor to make rating predictions.

For users who appear in both source and target domains, we infer their domain-specific preference weights using the corresponding trained models. Specifically, let $\mathbf{w}_u^{tgt,raw}$ denote the preference weight vector over target-domain GMM components inferred from target-domain interactions, and let $\mathbf{w}_u^{tgt,trans}$ denote the preference weight vector transferred from the source domain via optimal transport. We then combine the two weight vectors as follows:

$$\mathbf{w}_u^{tgt} = \alpha \mathbf{w}_u^{tgt,raw} + (1-\alpha)\mathbf{w}_u^{tgt,trans}, \tag{5.3}$$

where α controls the relative contribution of target-domain and source-domain information. In our experiments, we set $\alpha = 0.5$ to assign equal importance to both sources. The combined preference weights $\mathbf{w}_u^{tgt}$ are subsequently normalized and fed into the trained target-domain rating prediction MLP to produce the final rating predictions, from which RMSE and MAE are computed.

Chapter 6

Results

In this chapter, we present and analyze the experimental results obtained from the settings described in Chapter 5. The primary objective of this chapter is to empirically evaluate the effectiveness of the proposed DUP-OT framework under the non-overlapping cross-domain recommendation setting. We systematically compare DUP-OT with representative single-domain and cross-domain baseline methods, and analyze the results with respect to the research questions defined earlier. Through this analysis, we aim to provide clear evidence of the advantages brought by cross-domain information transfer and distribution-based user preference modeling.

6.1   Overall Performance

The overall results of the three groups of experiments introduced in Chapter 5 are summarized in Table 6.1. The table reports rating prediction performance on the target domain in terms of RMSE and MAE, where lower values indicate better recommendation accuracy. Boldfaced numbers denote the best-performing method under each experimental setting. To reduce the influence of randomness caused by parameter initialization and data shuffling, all results shown in the table are averaged over five independent runs with different random seeds. The complete experimental results of the proposed DUP-OT framework under different random seeds, including both the full model and its variants across multiple domain pairs, are reported in Appendix A for completeness and reproducibility.

From Table 6.1, it can be observed that the complete version of the proposed DUP-OT framework consistently achieves the best performance across all cross-domain experimental settings. This performance advantage is observed consistently across different source domains, indicating that the proposed framework is not sensitive to a particular domain pair. These results suggest that incorporating source-domain knowledge via optimal transport provides substantial benefits for improving rating prediction accuracy in the target domain.

Table 6.1: Rating prediction performance of baseline models and the proposed DUP-OT framework.

(a) Cross-domain recommendation results with Electronics as the target domain

$D_s \rightarrow D_t$	Digital Music $\rightarrow$ Electronics		Movies & TV $\rightarrow$ Electronics		Video Games $\rightarrow$ Electronics	
Model	RMSE	MAE	RMSE	MAE	RMSE	MAE
TDAR	1.5688	1.0904	1.5677	1.0896	1.5688	1.0902
DUP-OT (w/o source)	1.3699	0.9463	1.3494	0.8806	1.3716	**0.9481**
DUP-OT (w/ source)	**1.2919**	**0.8965**	**1.2907**	**0.8774**	**1.3376**	1.0032

(b) Single-domain recommendation results on the Electronics dataset

Model	RMSE	MAE
LightGCN	1.5317	1.1179
NeuMF	1.4599	1.3297
DUP-OT (w/o source, average)	**1.3636**	**0.9250**

6.2 Effect of Cross-Domain Information (RQ1)

To investigate whether leveraging cross-domain information contributes to improved recommendation performance, we compare DUP-OT with source-domain information against its ablated variant trained without any source-domain interactions. The two models share identical network architectures and training protocols, and the only difference lies in whether source-domain preference weights are transferred to the target domain during inference.

As shown in Table 6.1, DUP-OT with source-domain information consistently outperforms DUP-OT (w/o source) across all evaluated domain pairs. Both RMSE and MAE are reduced when source-domain information is introduced, indicating more accurate rating prediction. This performance gap demonstrates that source-domain user preferences provide complementary information that cannot be fully captured by target-domain data alone. The results therefore validate the effectiveness of cross-domain preference transfer in the proposed DUP-OT framework.

6.3 Effect of Distributional User Preference Modeling (RQ2)

To assess the impact of modeling user preferences as distributions rather than point embeddings, we compare DUP-OT (w/o source) with strong single-domain baselines, namely LightGCN and NeuMF. These baseline methods represent each user using a single latent vector and do not explicitly model uncertainty or multi-modal preference structures.

As reported in Table 6.1, DUP-OT (w/o source) consistently outperforms both LightGCN and NeuMF on the Electronics dataset, even though it does not leverage any source-domain data. This observation indicates that distribution-based user preference modeling provides a more expressive and flexible representation of user interests. By modeling user preferences as mixture weights over multiple

latent components, DUP-OT is able to capture heterogeneous and uncertain user behaviors that are difficult to represent using single-vector embeddings.

6.4　Comparison with Cross-Domain Baselines (RQ3)

We further compare the proposed DUP-OT framework with TDAR, a representative cross-domain recommendation method designed for non-overlapping settings. As shown in Table 6.1, DUP-OT consistently achieves lower RMSE and MAE than TDAR across all evaluated domain pairs.

These results suggest that aligning user preferences at the distribution level via optimal transport is more effective than adversarial feature alignment approaches for cross-domain recommendation. In particular, DUP-OT directly operates on preference representations and performs explicit distribution alignment, which leads to more stable and accurate knowledge transfer between domains without relying on shared users or items.

6.5　Robustness Analysis

To evaluate the robustness of the proposed DUP-OT framework, we conduct multiple independent runs of each experiment using different random seeds. The detailed results for each individual run are reported in Appendix A. Across all runs, DUP-OT exhibits consistently strong performance, with relatively small variations in RMSE and MAE across different initializations.

This consistency indicates that the proposed framework is stable with respect to random initialization and training dynamics. The robustness of DUP-OT further supports its practical applicability for cross-domain recommendation scenarios, where reliable performance across repeated runs is an important requirement.

6.6　Summary

In summary, the experimental results presented in this chapter provide strong and consistent evidence for the effectiveness of the proposed DUP-OT framework. The results confirm that cross-domain preference transfer significantly improves target-domain recommendation performance, that modeling user preferences as distributions offers clear advantages over conventional vector-based representations, and that DUP-OT outperforms existing cross-domain recommendation methods under non-overlapping settings. Together, these findings demonstrate the practical value and robustness of the proposed approach.

Chapter 7

Conclusion

This thesis investigated cross-domain recommendation under non-overlapping settings, where neither users nor items are shared across domains during the training phase. Such scenarios can arise in real-world applications, yet remain challenging for conventional methods that rely on shared entities for knowledge transfer or domain adaptation. What's more, most existing approaches represent user preferences as fixed-point embeddings, which may fail to capture the inherent uncertainty and diversity of user interests.

To address these challenges, we proposed a novel framework, DUP-OT, which models user preferences as probability distributions and performs cross-domain knowledge transfer via optimal transport, without relying on overlapping users or items during training, while allowing such information to be optionally incorporated at inference time when available.

7.1   Summary of Findings

The core idea of DUP-OT is to represent user preferences as mixture weights over Gaussian components in a shared latent space, rather than as fixed-point embeddings, and to enable cross-domain knowledge transfer under a strictly non-overlapping training setting. By fitting Gaussian Mixture Models on item representations in each domain, user preferences can be expressed as distributional representations that naturally capture uncertainty and multi-modal interest structures. Optimal transport is then employed to align preference weights across domains at the component level, enabling principled and interpretable cross-domain transfer without relying on overlapping users or items during training.

Extensive experiments conducted on multiple Amazon Review datasets demonstrate the effectiveness of the proposed framework under strictly non-overlapping training settings. The experimental results show that DUP-OT consistently outperforms strong single-domain baselines as well as representative cross-domain recommendation methods under non-overlapping settings. In particular, incorporating source-domain preference information via optimal transport leads to substantial improvements in target-domain rating prediction accuracy. Even without leveraging source-domain data, the distribution-based preference modeling adopted in DUP-OT yields superior performance compared to conventional vector-based approaches, highlighting the importance of modeling uncertainty and

preference heterogeneity.

Additional analyses further indicate that the proposed framework is robust with respect to random initialization and training dynamics. The consistent performance across multiple random seeds suggests that the observed improvements are not due to incidental effects, but rather reflect the inherent advantages of the proposed modeling and transfer strategy.

7.2   Contributions and Implications

This thesis makes several contributions to the study of cross-domain recommendation. First, it introduces a distribution-based formulation of user preferences that departs from the dominant paradigm of point embeddings. By modeling preferences as mixture weights over latent components, the proposed approach provides a more expressive representation capable of capturing diverse and uncertain user behaviors.

Second, this work highlights optimal transport as a principled and flexible mechanism for cross-domain preference transfer under non-overlapping training settings. Unlike approaches that rely on adversarial alignment or shared entities during training, DUP-OT leverages optimal transport at inference time to incorporate source-domain preference distributions when such information is available. By performing explicit distribution-level transfer and fusion rather than representation-level alignment, the proposed framework enables stable, interpretable, and scenario-adaptive knowledge transfer across domains. This perspective broadens the applicability of optimal transport in recommender systems and underscores its potential for handling domain shift and data sparsity.

From a practical standpoint, the proposed framework is particularly well-suited for real-world recommendation scenarios involving cold-start users or fragmented platform ecosystems. By avoiding reliance on shared users or items during training and allowing auxiliary cross-domain user information to be optionally incorporated at inference time, DUP-OT can provide meaningful recommendations even when target-domain data are limited. This flexibility makes the framework applicable to a wide range of industrial settings, such as cross-platform content recommendation and emerging service deployment.

7.3   Limitations

Despite its effectiveness, the proposed approach has several limitations that should be acknowledged. First, the structure of the Gaussian Mixture Model is closely tied to the scale and distribution of items within each domain. When the number of items or their distribution changes substantially, the appropriate number of mixture components may also change, which in turn requires reconfiguring the second-stage neural architectures. In particular, both the multi-layer perceptrons used for modeling user-specific mixture weights and those used for rating prediction are structurally coupled to the number of mixture components. As a result, adapting the framework to domains with significantly different item scales or supporting dynamic scaling may require architectural modification, posing challenges for large-scale or rapidly evolving deployment scenarios.

Second, the proposed framework implicitly assumes a certain degree of semantic relatedness between the source and target domains. While the model does not rely on overlapping users or items during

training, effective cross-domain transfer still depends on the existence of shared or compatible latent structures across domains. When the semantic gap between domains is extremely large, learning a meaningful shared representation space becomes challenging, which may limit the effectiveness of distribution-level alignment and degrade the quality of the resulting transport plan.

Finally, the temporal dynamics of user preferences are incorporated only through a lightweight timestamp encoding. While this design choice balances model complexity and performance, it does not explicitly model long-term sequential dependencies or evolving user interests over time.

7.4   Future Work

Several directions for future research can be explored to extend this work. One promising direction is to investigate more expressive distribution families for modeling user preferences, such as mixtures with full covariance structures or non-Gaussian components, which may capture richer interaction patterns.

Another important direction is to integrate more advanced temporal modeling techniques into the framework. Incorporating sequential models or continuous-time dynamics could enable more accurate modeling of preference evolution and further improve recommendation quality.

In addition, extending the proposed approach to multi-source or multi-target cross-domain scenarios would enhance its applicability in complex real-world environments.

In conclusion, this thesis demonstrates that modeling user preferences as distributions and transferring them via optimal transport provides a powerful and flexible solution to cross-domain recommendation under non-overlapping training settings. The proposed DUP-OT framework offers both theoretical insights and practical benefits, and serves as a foundation for future research in distribution-based recommender systems.

References

[1] D. Bawden and L. Robinson. Information overload: An overview. In *Oxford Encyclopedia of Political Decision Making*. Oxford University Press, Oxford, June 2020. This is a draft of a chapter that has been published by Oxford University Press in the book Oxford Encyclopedia of Political Decision Making.

[2] Jun Lv and Xuan Liu. The impact of information overload of e-commerce platform on consumer return intention: Considering the moderating role of perceived environmental effectiveness. *International Journal of Environmental Research and Public Health*, Vol. 19, No. 13, 2022.

[3] Guihua Zhang, Junwei Cao, and Dong Liu. Examining the influence of information overload on consumers' purchase in live streaming: A heuristic-systematic model perspective. *PLOS ONE*, Vol. 18, No. 8, pp. 1–16, 08 2023.

[4] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, Vol. 40, No. 3, pp. 56–58, 1997.

[5] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, Vol. 12, No. 4, pp. 331–370, 2002.

[6] Robin Burke, Alexander Felfernig, and Mehmet H. Göker. Recommender systems: An overview. *AI Magazine*, Vol. 32, No. 3, pp. 13–18, Jun. 2011.

[7] Dietmar Jannach and Michael Jugovac. Measuring the business value of recommender systems. *ACM Trans. Manage. Inf. Syst.*, Vol. 10, No. 4, December 2019.

[8] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, Vol. 46, pp. 109–132, 2013.

[9] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, Vol. 2009, No. 1, p. 421425, 2009.

[10] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, p. 175–186, New York, NY, USA, 1994. Association for Computing Machinery.

[11] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, 2001.

[12] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering, 2013.

[13] Lyle H Ungar and Dean P Foster. Clustering methods for collaborative filtering. In *AAAI workshop on recommendation systems*, Vol. 1, pp. 114–129. Menlo Park, CA, 1998.

[14] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, Vol. 42, No. 8, pp. 30–37, 2009.

[15] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, Vol. 20. Curran Associates, Inc., 2007.

[16] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, p. 880–887, New York, NY, USA, 2008. Association for Computing Machinery.

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pp. 173–182, 2017.

[18] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

[19] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

[20] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, p. 1441–1450, New York, NY, USA, 2019. Association for Computing Machinery.

[21] Rianne Van Den Berg, N Kipf Thomas, and Max Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, Vol. 2, No. 8, p. 9, 2017.

[22] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 974–983, 2018.

[23] 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*, SIGIR'19, p. 165–174, New York, NY, USA, 2019. Association for Computing Machinery.

[24] 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*, SIGIR '20, p. 639–648, New York, NY, USA, 2020. Association for Computing Machinery.

[25] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, No. 01, pp. 346–353, Jul. 2019.

[26] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, p. 689–698, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[27] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, p. 515–524, New York, NY, USA, 2017. Association for Computing Machinery.

[28] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. Cfgan: A generic collaborative filtering framework based on generative adversarial networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, p. 137–146, New York, NY, USA, 2018. Association for Computing Machinery.

[29] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. Diffusion recommender model. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, p. 832–841, New York, NY, USA, 2023. Association for Computing Machinery.

[30] Gerard Salton, Chris Buckley, and Edward A Fox. Automatic query formulations in information retrieval. *Journal of the American society for information science*, Vol. 34, No. 4, pp. 262–280, 1983.

[31] Michael Pazzani and Daniel Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, Vol. 27, No. 3, pp. 313–331, 1997.

[32] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the eighth international workshop on data mining for online advertising*, pp. 1–9, 2014.

[33] Tianqi Chen. Xgboost: A scalable tree boosting system. *Cornell University*, 2016.

[34] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, Vol. 30, , 2017.

[35] Michael J Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, Vol. 13, No. 5, pp. 393–408, 1999.

[36] Prem Melville, Raymond J Mooney, Ramadass Nagarajan, et al. Content-boosted collaborative filtering for improved recommendations. *Aaai/iaai*, Vol. 23, pp. 187–192, 2002.

[37] Daniel Billsus and Michael J Pazzani. User modeling for adaptive news access. *User modeling and user-adapted interaction*, Vol. 10, No. 2, pp. 147–180, 2000.

[38] Justin Basilico and Thomas Hofmann. Unifying collaborative and content-based filtering. In *Proceedings of the twenty-first international conference on Machine learning*, p. 9, 2004.

[39] Steffen Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pp. 995–1000. IEEE, 2010.

[40] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pp. 89–96, 2005.

[41] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, Vol. 11, No. 23-581, p. 81, 2010.

[42] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 650–658, 2008.

[43] Bin Li, Qiang Yang, and Xiangyang Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th annual international conference on machine learning*, pp. 617–624, 2009.

[44] Weike Pan, Evan Xiang, Nathan Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 24, pp. 230–235, 2010.

[45] Xianghui Zhu, Mengqun Jin, Hengyu Zhang, Chang Meng, Daoxin Zhang, and Xiu Li. Modeling domains as distributions with uncertainty for cross-domain recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, p. 2517–2521, New York, NY, USA, 2024. Association for Computing Machinery.

[46] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Cross-domain recommendation: An embedding and mapping approach. In *Ijcai*, Vol. 17, pp. 2464–2470, 2017.

[47] Genki Kusano and Masafumi Oyamada. Cross-domain user similarity without overlapping attributes via optimal transport theory. 2023.

[48] Weiming Liu, Chaochao Chen, Xinting Liao, Mengling Hu, Yanchao Tan, Fan Wang, Xiaolin Zheng, and Yew Soon Ong. Learning accurate and bidirectional transformation via dynamic embedding transportation for cross-domain recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, No. 8, pp. 8815–8823, Mar. 2024.

[49] Xinhang Li, Zhaopeng Qiu, Xiangyu Zhao, Zihao Wang, Yong Zhang, Chunxiao Xing, and Xian Wu. Gromov-wasserstein guided representation learning for cross-domain recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, p. 1199–1208, New York, NY, USA, 2022. Association for Computing Machinery.

[50] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1059–1068, 2018.

[51] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 2615–2623, 2019.

[52] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2942–2951, 2020.

[53] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, Vol. 22, No. 1, p. 89–115, January 2004.

[54] Benjamin M Marlin. Modeling user rating profiles for collaborative filtering. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, Vol. 16. MIT Press, 2003.

[55] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, Vol. 3, No. Jan, pp. 993–1022, 2003.

[56] Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding, 2015.

[57] Cédric Villani, et al. *Optimal transport: old and new*, Vol. 338. Springer, 2008.

[58] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, Vol. 11, No. 5-6, pp. 355–607, 2019.

[59] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, Vol. 26, , 2013.

[60] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617. PMLR, 2018.

[61] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 39, No. 9, pp. 1853–1865, 2016.

[62] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. *Advances in neural information processing systems*, Vol. 28, , 2015.

[63] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[64] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, Vol. 12, pp. 2825–2830, 2011.

[65] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 188–197, Hong Kong, China, November 2019. Association for Computational Linguistics.

[66] Wenhui Yu, Xiao Lin, Junfeng Ge, Wenwu Ou, and Zheng Qin. Semi-supervised collaborative filtering by text-enhanced domain adaptation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, p. 2136–2144, New York, NY, USA, 2020. Association for Computing Machinery.

[67] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830, 2011.

[68] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *J. Mach. Learn. Res.*, Vol. 22, No. 1, January 2021.

Appendix

A   Additional Experimental Results

This section reports the complete experimental results on all domain pairs, including RMSE and MAE under different random seeds.

Table A.1: Additional experimental results of the full DUP-OT (w/ s) model under different random seeds on three domain pairs, with Electronics as the target domain.

	Digital Music		Movies & TV		Video Games	
Random Seed	RMSE	MAE	RMSE	MAE	RMSE	MAE
7	1.3467	0.7717	1.3471	0.7897	1.4132	1.2789
11	1.2291	0.9498	1.2779	0.9621	1.3148	1.0666
17	1.3111	0.8496	1.3402	0.7805	1.3023	1.0907
33	1.2683	1.0250	1.2493	0.8626	1.3610	0.7569
2025	1.3043	0.8866	1.2390	0.9920	1.2965	0.8227
Mean	1.2919	0.8965	1.2907	0.8774	1.3376	1.0032
Variance	0.0016	0.0074	0.0020	0.0075	0.0019	0.0362

Table A.2: Additional experimental results of the DUP-OT (w/o s) variant under different random seeds on three domain pairs, with Electronics as the target domain.

	Digital Music		Movies & TV		Video Games	
Random Seed	RMSE	MAE	RMSE	MAE	RMSE	MAE
7	1.3445	0.9046	1.3835	0.9677	1.3818	0.9886
11	1.3523	0.9336	1.3798	0.9763	1.3647	0.9348
17	1.3757	0.9595	1.3928	0.9615	1.3974	0.9764
33	1.3719	0.9298	1.3793	0.9335	1.3508	0.9259
2025	1.4052	1.0041	1.3668	0.9624	1.3635	0.9150
Mean	1.3699	0.9463	1.3805	0.9603	1.3716	0.9481
Variance	0.0004	0.0011	0.0001	0.0002	0.0003	0.0008

Publications

- 肖 子吟 and 鈴村 豊太郎. 非重複クロスドメイン推薦に向けたガウス混合ユーザ嗜好分布と最適輸送の統合フレームワーク — DUP-OT —. WebDB 夏のワークショップ 2025 2025

- Ziyin Xiao and Toyotaro Suzumura. Modeling User Preferences as Distributions for Optimal Transport-based Cross-domain Recommendation under Non-overlapping Settings. In the Proceedings of The 30th Pacific-Asia Conference on Knowledge Discovery and Data Mining. (Submitted)

Acknowledgment

The completion of this thesis would not have been possible without the support and guidance of many individuals. I would like to take this opportunity to express my sincere gratitude to all those who have contributed to this work.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Toyotaro Suzumura, for his continuous support, insightful guidance, and encouragement throughout my research journey. His expertise and constructive feedback have been invaluable in shaping the direction and quality of this thesis.

I am also grateful to Chen, Refaldi, and Igor for their enthusiasm for research, which has greatly inspired me during our weekly meetings. The discussions with them not only enhanced my understanding of the research topics but also broadened my academic horizons.

I would further like to thank my lab mates, Limin Wang and Yu Tomita, for sharing their experiences and formats for thesis writing, which greatly helped me in organizing and presenting my research effectively.

I would further like to thank the other members of Suzumura Lab for their helpful discussions and continuous support throughout my study. They not only provided technical assistance but also fostered a collaborative and stimulating research environment, making my study abroad experience both rewarding and memorable.

Finally, I would like to express my heartfelt gratitude to my parents for their unwavering support and encouragement throughout my life. Their belief in me has been a constant source of motivation.

Use of Generative AI Tools. Generative AI tools were used during the preparation of this thesis to assist with language refinement and clarity of presentation. All technical content, methodological design, experimental implementation, and scientific conclusions were developed and verified by the author. The author takes full responsibility for the originality and accuracy of the work.