NEURAL RECOMMENDER MODELS FOR SPARSE AND SKEWED BEHAVIORAL DATA

BY

ADIT KRISHNAN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Associate Professor Hari Sundaram, Chair
Professor ChengXiang Zhai
Professor Jiawei Han
Dr. Mahashweta Das

ABSTRACT

Modern online platforms offer recommendations and personalized search and services to a large and diverse user base while still aiming to acquaint users with the broader community on the platform. Prior work backed by large volumes of user data has shown that user retention is reliant on catering to their *specific* eccentric tastes, in addition to providing them popular services or content on the platform [50].

Long-tailed distributions are a fundamental characteristic of human activity, owing to the bursty nature of human attention [12]. As a result, we often observe skew in data facets that involve human interaction. While there are superficial similarities to Zipf's law in textual data [180] and other domains [66], the challenges with user data extend further. Individual words may have skewed frequencies in the corpus, but the long-tail words by themselves do not significantly impact downstream text-mining tasks. On the contrary, while sparse users (a majority on most online platforms [62]) contribute little to the training data, they are equally crucial at inference time. Perhaps more so, since they are likely to churn [229].

In this thesis, we study platforms and applications that elicit user participation in rich social settings incorporating user-generated content, user-user interaction, and other *modalities* of user participation and data generation. For instance, users on the Yelp review platform participate in a follower-followee network [1] and also create and interact with review text (two modalities of user data). Similarly, community question-answer (CQA) platforms incorporate user interaction and collaboratively authored content [2] over diverse domains and discussion threads. Since user participation is multimodal, we develop generalizable abstractions beyond any single data-modality.

Specifically, we aim to address the *distributional mismatch* that occurs with user data independent of dataset specifics; While a minority of the users generates most training samples, it is insufficient only to learn the preferences of this subset of users. As a result, the data's overall skew and individual users' sparsity are closely interlinked: sparse users with uncommon preferences are under-represented. Thus, we propose to treat these problems jointly with a skew-aware grouping mechanism that iteratively sharpens the identification of preference groups within the user population [96]. As a result, we improve user characterization; content recommendation and activity prediction (+6-22% AUC, +6-43% AUC, +12-25% RMSE over *state-of-the-art* baselines), *primarily for users with sparse activity.*

[1] https://neo4j.com/docs/graph-algorithms/current/yelp-example/
[2] https://stackoverflow.com/

The size of the item or content inventories compounds the skew problem. Recommendation models can achieve very high aggregate performance while recommending only a tiny proportion of the inventory (as little as 5%) to users. We propose a data-driven solution guided by the aggregate co-occurrence information across items in the dataset. We specifically note that different co-occurrences are not equally significant; For example, some co-occurring items are easily substituted while others are not. We develop a self-supervised learning framework where the aggregate co-occurrences guide the recommendation problem while providing room to learn these variations among the item associations. As a result, we improve coverage to 100% (up from 5%) of the inventory and increase long-tail item recall up to 25% [95].

We also note that the skew and sparsity problems repeat across data modalities. For instance, social interactions and review content both exhibit aggregate skew, although individual users who actively generate reviews may not participate socially and vice-versa [97]. It is necessary to differentially weight and merge different data sources for each user towards inference tasks in such cases. We show that the problem is inherently adversarial since the user participation modalities compete to describe a user accurately. We develop a framework to unify these representations while algorithmically tackling mode collapse, a well-known pitfall with adversarial models.

A more challenging but important instantiation of sparsity is the few-shot setting or cross-domain setting. We may only have a single or a few interactions for users or items in the sparse domains or partitions. We show that contextualizing user-item interactions helps us infer behavioral invariants in the dense domain, allowing us to correlate sparse participants to their active counterparts (resulting in 3x faster training, 19% recall gains in multi-domain settings).

Finally, we consider the multi-task setting, where the platform incorporates multiple distinct recommendations and prediction tasks for each user. A single user representation is insufficient for users who exhibit different preferences along each dimension. At the same time, it is counter-productive to handle correlated prediction or inference tasks in isolation. We develop a multi-faceted representation approach grounded on residual learning with heterogeneous knowledge graph representations, which provides us an expressive data representation for specialized domains and applications with multimodal user data. We achieve knowledge sharing by unifying task-independent and task-specific representations of each entity with a unified knowledge graph framework.

In each chapter, we also discuss and demonstrate how the proposed frameworks directly incorporate a wide range of gradient-optimizable recommendation and behavior models, maximizing their applicability and pertinence to user-centered inference tasks and platforms.

To my family and friends, for their love and support.

ACKNOWLEDGMENTS

I convey my sincere gratitude to those who have contributed to my academic and social life at the University of Illinois, Urbana-Champaign (UIUC), throughout my Ph.D.

First and foremost, I would like to thank my advisor, Professor Hari Sundaram, who has been a consistent source of support and advice on matters academic and beyond, whenever I needed it. Hari has laid the foundation for how I structure and express my thoughts and provided me a framework to evaluate ideas and look at the bigger picture beyond specific research objectives and milestones.

His encouragement and focus towards identifying fulfilling ideas and research pursuits have significantly influenced this thesis's directions and contents. Beyond that, he has been very accessible despite his incredibly busy schedules and numerous duties. For that, I am very grateful.

I would also like to convey my sincere gratitude to my other committee members, Professor ChengXiang Zhai, Professor Jiawei Han, and Dr. Mahashweta Das. Professor Zhai's well-structured course material and research summaries played a significant role in the initial years of my Ph.D. They helped me ground my initial research ideas and directions. I learned a great deal through my interactions with Professor Han and his students. My collaborations with him and his students were incredibly productive and engaging, and led to multiple impactful research contributions.

I enjoyed working with Mahashweta Das and Mangesh Bendre (former Ph.D. student at UIUC) over two summers at Visa Research. Their support and guidance enabled me to make impactful contributions at Visa. They helped me better understand the practical implications of my research pursuits beyond the academic objectives.

I also owe a lot to my colleagues at the Crowd Dynamics Lab for the fruitful research discussions and collaborations that shaped my work. Numerous interns and undergraduate collaborators have also contributed to my research efforts, and I am thankful for their dedication and meticulous work. My friends outside of research also deserve a special mention for helping me maintain an active social life and engage in non-academic pursuits that enriched my time at Urbana-Champaign.

Finally, this thesis could not have materialized without the consistent and unwavering support of my family. Their love and encouragement are the common denominator to all of my accomplishments, big or small.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 RECOMMENDATION WITH SKEWED AND SPARSE BEHAVIORAL DATA

Recommender systems are critical to a diverse set of e-commerce applications, including media recommendations (e.g., Netflix), products (e.g., Amazon, Walmart), friend recommendations (e.g., Facebook), and online advertising (e.g., Google). The recommendation task typically incorporates user inferencing as well, i.e., understanding users and predicting aspects of their behavior with historical behavioral data.

However, we need to address two ubiquitous challenges with pre-recorded user interactions to facilitate personalization, recommendations, and inference efforts. First, the distribution of user activity is highly skewed. We observe these heavy and long-tailed distributions for both user interests and interaction patterns. The presence of heavy-tailed distributions, a fundamental characteristic of human activity [12], implies that one cannot rely on data scale (algorithmic scalability is another challenge) alone to produce high-quality inference for all segments of the user population. The second is data sparsity, wherein the historical records of individual users and items lack the requisite density or volume of interactions to infer meaningful trends.

In recent times, deep neural network architectures have delivered impressive results in various machine learning domains, including computer vision, speech analysis, and recommender systems. However, a close examination of popular neural recommendation models reveals a paradox: while the overall item recommendation or action identification accuracy is high, accuracy levels are inadequate for a significant chunk of the target audience. Most users do not receive recommendations aligned to their specific tastes but are instead recommended popular generic items in the product inventory. Recall measured on a per-item basis also indicates a similar trend. Higher performance on popular item recommendations masks the poor recall in the long-tail. The masking effect is pronounced in the item-to-item collaborative filtering setting with deep-learned models (also referred to as Neural Collaborative Filtering or NCF) owing to the inherent biases induced by the skewed and sparse training data reflected in the training objectives for these models.

Improving and personalizing recommendations and achieving better performance for users with limited activity is critical to widely adopted neural recommender models. This is, however, a challenging task, owing to the immense complexities and computational costs associated with developing, training, and analyzing neural models across very diverse application scenarios. This thesis decomposes the broader challenge by identifying common subprob-

lems that repeat across a wide range of neural recommendation and inference models. We then develop generalizable solutions applicable across a wide range of model architectures and application scenarios. To better understand these challenges, we start by defining a few recurring terms in the next section.

1.2 TERMS AND DEFINITIONS

Common to all chapters in this thesis, the two fundamental entities of interest are *users* and *items*, although other associated entities might form part of the training data. The primary *recommendation task* constitutes matching items to users. Across our chapters, we consider several types of items and demonstrate the importance of addressing the sparsity and skew challenges across these diverse recommendation applications. We consider items that may be further decomposed, such as textual posts, characterized by the words, phrases, and word co-occurrences. We also study the complementary setting, where items are discrete independent entities, and show how to identify the similarities between them to understand and leverage their recurrent association structures.

1.2.1 Data-Modality

On modern online platforms, user and item data typically appear in multiple independently generated forms [171], each describing different facets of the respective entities or their interactions. To leverage these different facets of data towards recommendation and prediction tasks often requires the unification of diverse modeling considerations. We refer to each such facet of data as a *data-modality*.

The most common data-modality is that of *user-item interactions*, such as item purchases on e-commerce platforms or user-content interactions on a community question-answer (CQA) website. Further, the interactions may be untyped (e.g., all interactions are item purchases) or typed (e.g., users may interact with content by either liking, editing, or commenting on it).

Users and items may each be associated with descriptive feature modalities (interchangeably referred to as user or item *attributes* in our work) such as the demographic attributes of users or textual descriptions of items. Platforms incorporating social elements also include the user-user interaction modality, analogous to user-item interactions. A fourth distinct modality is that of *interaction context* - this contains features associated with each user-item or user-user interaction, such as the time/day of interaction. Unlike user or item

attributes, interaction features are not directly associated with either the user or the item and are specific to each interaction.

1.2.2 Tasks and Domains

Prediction and inference tasks may be associated with distinct data-modalities (e.g., predict an attribute of the user or item), as opposed to the *recommendation task*, which is specifically associated with the user-item interaction modality.

The term *multimodal recommendation* or *multimodal prediction* refers to item recommendation or user/item attribute prediction tasks that leverage more than one of the above data-modalities simultaneously. Conventional collaborative filtering is typically a unimodal recommendation problem leveraging only the user-item interaction modality, while the social recommendation task is bimodal. Multimodality introduces unique challenges since the different data-modalities may generate conflicting inferences about user preferences or item characteristics. These conflicts are best resolved contextually on a per-user or per-item basis.

A *recommendation domain* constitutes a specific set of users and the items that form the candidate pool for the recommendation model. A *multi-domain* recommendation problem incorporates two or more domains of recommendation. In contrast, *cross-domain* recommendation leverages users' item preferences in one domain to infer user preferences in the other. We consider both, *disjoint* and *overlapping* domains along the user or item axes in Chapter 6. Each prediction/recommendation task is associated with a specific recommendation domain.

1.2.3 Model Characteristics

We refer to the modeling considerations incorporated by each prediction or recommendation model as its *expressivity*. For instance, a model that includes *per-user feature weighting expressivity* can provide user-specific weights to each data-modality towards the prediction or recommendation task. Analogously, models with *multiplicative feature combination expressivity* can include products of distinct feature values towards inferences, as opposed to *additive expressivity* alone.

Finally, we study and classify models with regard to their *training* and *inference-time* characteristics. A more expressive model is capable of achieving stronger aggregate performance but may overfit to noise, among other training challenges [5]. One of our central goals is to reduce the parametric overheads with minimal loss in expressivity. Further, we aim to develop data-augmentation [138], self-supervision [61] and regularization strategies [210] to

train highly expressive recommender models, such as neural networks [111]. On the other hand, *inference-time* refers to all post-training aspects of the model, such as testing and deployment/online model updates (where applicable).

1.3 WHY IS THE SKEW AND SPARSITY PROBLEM INTERESTING?

Long-tailed distributions are commonly observed in user behavior data [12], and in particular, across data-modalities involving user interactions on online platforms. We observe skew in many facets, the social interactions on online platforms [142], the popularity of specific items and content [50], and even the contextual features associated with each interaction.

The challenges of skew and sparsity are not unique to behavior data. *Zipf's law* in textual data [180], imbalanced data in computer vision [93], and other machine learning domains [66] present similar challenges. However, we emphasize a few notable differences from recommendation tasks.

Consider Zipf's law in text-mining: While individual words exhibit skewed frequencies in the corpus, the long-tail words do not significantly impact downstream text-mining tasks. An aggregate inference on the set of all words proves sufficient for most downstream tasks, e.g., distributional word embeddings that learn aggregate co-occurrence patterns [140]. The quality of representations or inference associated with long-tail words has a limited impact on most application scenarios.

However, with user-based prediction and recommendation applications, sparse users repeat across the training data and the trained model's inference or recommendation objectives. At inference-time, sparse and data-rich users, each constitutes a single output instance.

Hence, note the *distributional disparity*: While sparse users contribute little to the training data, they are equally crucial at inference-time. As a consequence, *while the overall item recommendation or action identification accuracy is high, for a significant chunk of the target audience, accuracy levels are poor, as we observe in Chapter 4.* Most users do not receive recommendations aligned to their specific tastes but are instead recommended popular generic items in the product inventory. Recall measured on a per-item basis also indicates a similar trend. Higher performance on popular item recommendations masks the poor recall in the long-tail.

A second challenge is the innate multimodality of user data. Specifically, online platforms and applications elicit multimodal user participation incorporating user-item interactions, user-generated content, user-user interactions, and other modalities of user activity. Irrespective of the specifics, users, and items are each associated with multiple data-modalities exhibiting different distributional properties, e.g., varying degrees of feature skew and spar-

sity. For example, the importance of a specific social link towards recommendation depends on the two participants' availability of item preference data. In effect, we must examine multiple competing hypotheses associated with different data modalities to infer suitable recommendations and predict users' or items' attributes.

1.4 PROBLEMS ADDRESSED IN THIS THESIS

Owing to the above unique characteristics of user behavior data on online platforms, sparsity mitigation strategies in domains such as text-mining and computer vision are not directly applicable to user-inference and recommendation problems. We identify the following broad challenges associated with personalized inference and recommendation tasks:

1.4.1 Distributional Mismatch between Training Data and Inference

While sparse long-tail users and items are under-represented in the training data *vis-a-vis* active users and items, they are equally important at inference-time. Thus, the aggregate inferences from the training data can adversely impact long-tail users.

Recommendation vs. Other Machine Learning Domains: Unlike domains such as computer vision where distributional similarities may be leveraged via transfer-learning [9], the user training samples and inference-time datapoints in recommendation exhibit significant distributional differences. Thus, we must develop training methods that are skew-aware and platform-independent and actively estimate and compensate for these disparities in a data-driven manner.

1.4.2 Handling Disparities with Multimodal Data

Inference tasks are complicated by users and items that do not exhibit correlated data generation trends across the different data modalities on a multimodal online platform. For example, active users on some parts of the platform may be relatively inactive on others. Platforms that offer social interactions alongside content consumption may attract users who are inactive in one of these two data-modalities on the platform.

The resulting asymmetry in the user population causes significant disparities across the different data-modalities on the platform, thus necessitating a user-level evaluation of each data-modality for any multimodal inference and recommendation tasks. As a result, it is necessary to generate task-specific and user / item-specific aggregate representations as a function of the available behavioral data across all the data-modalities.

1.4.3 The Cold-Start or Few-Shot Challenge

Most recommendation algorithms learn separate embedding representations for users and items and rely on either static metrics (such as the cosine similarity) or learned metrics (bilinear functions) to quantify user-item propensity. However, these learned representations are not meaningful for the *cold-start / few-shot* setting, where new items or users provide either no interactions or a handful of interactions, respectively. While the few-shot learning problem also appears in prediction problems with *few-shot classes* [156] (i.e., classes with a handful of samples), the multimodality and scale of user-data (note that each user/item is a few-shot instance, unlike few-shot classes) renders these solutions inapplicable or ineffective.

1.4.4 Handling Multi-Task Personalization

Personalization efforts are computationally expensive and hence, seldom directed towards a single task or objective. The learned user profiles or representations are simultaneously leveraged for a wide range of prediction, inference, and recommendation tasks to better understand and serve users on online platforms. Each task, however, benefits from distinct task-specific representations [172] of the users and items. Thus, it is necessary to efficiently combine and trade off shared cross-task representations and independent task-specific representations to enable knowledge sharing.

In combination, solutions to the above problems address a comprehensive range of common application scenarios involving neural recommenders. We now discuss the key contributions of this thesis towards each of the above challenges in greater detail.

1.5 CONTRIBUTIONS OF THIS THESIS

1.5.1 Unified Mitigation of User Behavior Skew and Sparsity

Prior work addresses sparse user representations with a suite of single-modality and cross-modality clustering methods such as social regularization, transfer learning, and leveraging external or auxiliary feature data to smooth sparse user data. Regularizing or smoothing user activity by clustering them with similar peer users is expected to provide a coherent profile to describe sparse users. However, this presupposes identifying behaviorally homogenous groups of users with archetypal group behavior profiles to address the sparsity challenge while maintaining consistency within each user group. While domain and platform specifics can help define grouping mechanisms, we aim to address the more general scenario without

assuming their availability.

While large coarse groups are detrimental to the group profile's informativeness and user-inference quality, small groups lack the constituent user data to learn a group profile, even though they may exhibit greater coherence.

We address the *distributional mismatches* between the behavioral training data and the inference-time users, independent of dataset specifics, by identifying the close link between user behavior skew and sparsity. Identifying more informative and coherent groups in the presence of skew helps us do a better job bridging incomplete or sparse data for individual users; Simultaneously, the reverse is also true; better inference for sparse users would help us create such coherent groups.

Restaurant Analogy: A useful analogy to think of in the context of user clustering is one of the seating of users in a restaurant. Topics or profiles of interest can be thought of as dishes served on tables so that users who like the same dishes are likely to sit together. Specifically, we exploit the non-parametric Pitman-Yor process (or CRP) [154] as a prior, our key innovation in addressing sparsity and behavior skew lies in how we seat users onto tables. Users could be moved across differently-sized tables to improve coherence while reflecting behavior/preference skew. To continue the above analogy, we propose an iterative user seating mechanism that is simultaneously skew aware by incentivizing exploration to find the best tables for long-tail users and deals with sparse users by seating them in the most coherent groups based on their limited observational data.

Analogous to the *expectation-maximization* algorithm [144], the E-step seats users on tables serving the most relevant profiles, while the M-step updates the individual profiles based on the seated users. This introduces deep-coupling across the skew-aware grouping mechanism and the profile learning process, unlike prior work in behavior modeling or recommendation that sequentially form groups and learn profiles or only introduce a superficial link. In combination with the non-parametric priors, our profiles can adapt to varying degrees of skew and sparsity. The profiles may be flexibly defined to accommodate varying data modalities, depending on the platform of application. Our model is efficient and scales linearly in the number of users and interactions. We employ caching optimizations to speed up inference and scale to massive datasets with parallelized batch sampling.

The user profiles learned by the resulting skew-aware grouping process describe the item preferences of the constituent users in each group. However, the inference is complicated by the presence of sparse items, specifically discrete non-decomposable items (such as long-tail items on *e-commerce* platforms) that do not have any associated feature data. In such cases, user-grouping in isolation is insufficient owing to the skew on the item side, especially in the massive inventory scenario that we now discuss.

1.5.2 Handling Sparse Items and Infinite Inventory Recommendation

The size of the item inventories compounds the user-side behavior or preference skew problem. Aggregate recommendation results can be deceptive; recommendation models can achieve very high aggregate performance by recommending a small proportion of the inventory (as little as 5%) to most users.

Inventory Coverage: The bias towards a small proportion of the item inventory results in repetitive and impersonal recommendations to the vast majority of the user population, not satisfying their eccentric tastes [50]. We note that the co-occurrence information of items in the dataset can be leveraged to improve personalization. Specifically, we can identify the set of long-tail items associated with each non-long-tail item and avoid recommending correlated or replaceable items. We recognize that co-occurrence likelihoods are not equally significant; For example, some co-occurring items are easily substituted while others are not replaceable in the item inventory.

Prior work treats the two problems independently, i.e., the recommendation task and the identification of item-item associations. As a result, conventional neighbor [129] employ static pre-computed criteria to form links between items and regularize the learned representations. While it is possible to add a similar term to the objective functions of neural recommenders, we aim to learn the association structure rather than imposing it on the model with pre-computed metrics.

Self-Supervision: Towards the above goal of learning a loosely guided item-item association structure, we self-supervise the recommenders with a competing association model to infer the inter-item association structure, guided by item co-occurrences in the feedback data. The two models iteratively supervise and refine each other.

Furthermore, this framework's modular nature permits architectural independence; the two competing models are chosen to fit the specific application or platform requirements. As a result, we can significantly improve inventory coverage for state-of-the-art neural recommenders and simultaneously increase long-tail item recall. A slight dip may be observed in the recall metrics for non-long-tail items, which can be addressed by reserving a preset proportion of the items shown to the user for non-long-tail items alone.

In contrast to the grouping mechanisms described in Section 1.5.1 and Section 1.5.2, we now discuss the scenario where we can explicitly access interaction data-modalities in addition to the user-item interactions, such as user-user social interactions in the form of a follower-followee network, signed network, or collaborative content production. This results in multimodal settings, where the data from the different interaction modalities are jointly leveraged for the user and item inference tasks and recommendations.

1.5.3 Leveraging Multimodal User and Item Data

Distributional skew and sparsity issues recur across data modalities such as social interactions, review content, and user-item interactions. However, individual users who actively generate reviews or exhibit content preferences may offer limited social activity [97]. Ideally, recommendation models should rely more on the user's content preferences to make recommendations in such cases and reweight the preferences of the user's social neighbors in the complementary scenario. This setting is not unique to the social recommendation problem but to any multimodal platform where users generate interaction data via multiple independent avenues to participate on online platforms. We identify that the multimodal inference problem is inherently adversarial since the different user participation modalities compete to describe a user or recommend suitable items accurately. We develop a framework to unify these modalities towards inference while algorithmically tackling mode collapse [8], a known pitfall with adversarial learning models.

When a static alignment model is applied, where each social link is assigned equal importance towards the preference identification task, it results in uninformative links weighted the same as influential social links that inform the user's preferences. We unify users' interest and social distributions by attributing their purchase decisions across their data modalities, specifically purchase histories and social links. As a result, users with limited item records may rely more on their social connections and vice-versa in the complementary scenario. Furthermore, each social link is independently weighted with all available data. Thus, we incorporate diversity across social links and learn the varied impact of each link on user preferences, enabling a more expressive interest space. We also maintain modularity in how we parametrize the attribution function. We permit the attribution functions to leverage social and preference representations computed via independent gradient-optimizable models. As a result, we are agnostic to the specific details of the social network (such as signed social networks [34], multi-relational networks [181] or heterogenous networks [119]) and/or the item aspects (e.g., item covariates [111]). In this manner, our efforts to learn multimodal representations build on modeling efforts in unimodal domains, such as the above social network representation and item recommendation models.

This subsection explored the multimodal user-inference and recommendation setting to bridge sparsity and skew in each data modality. The proposed strategy complements and extends the grouping-based unimodal approaches described in Section 1.5.1 and Section 1.5.2. However, it assumes that each user/item generates sufficient inference data along with at least one of the available data modalities. This assumption fails for new users/items who lack data along *all* data modalities and may offer just a handful or a single interaction to

infer their preferences. We now describe how *interaction context*, i.e., contextual features associated with each user-item interaction instead of user/item data modalities, may be leveraged towards few-shot representations.

1.5.4 Contextualizing User-Item Interactions for Few-Shot Recommendation

An important instantiation of sparsity is the few-shot setting. We make inferences and recommendations to users with items, each of which only offers either a single or a handful of interactions towards inference. The few-shot inference problem is increasingly important on platforms with large pools of new (sparse) users and items. It occurs naturally in offline recommendations such as restaurants, services, merchant, or business recommendations. For example, geographic disparities in population density cause training challenges for recommendation models focusing on rural and suburban locations. We view this problem as a cross-domain transfer learning task since the user and item sets (merchants/businesses/restaurants) do not show any significant overlap across the different geographic locations, each of which constitutes an independent recommendation domain. Online content recommendation applications also feature cross-domain scenarios, e.g., the discussion domains of the Stack-Exchange community question-answer website [1].

However, we do not require the domains to be defined in this manner. Our approach is still applicable to platforms that are not naturally partitioned into recommendation domains. In such settings, we can carve separate domains for the popular items and active users, and the sparse users and long-tail items, respectively, and transfer knowledge from the active subsets to the sparse subsets for few-shot recommendation in the sparse subsets. Prior work leverages either the shared users and items as anchors to enable cross-domain learning [132, 218] and/or aligns the latent structure of the learned user and item representations [107, 150], we do not rely on the presence of shared users or restrictive structure alignment methods that may reduce the expressivity of the recommendation model. Instead, we contextualize each user-item interaction to understand the most critical combinations of contextual features that facilitate a user-item interaction.

Our key intuition is to infer such behavioral invariants from a dense-source domain where we have voluminous interaction histories of users with items and apply (or adapt) these learned invariants towards inference in the sparse-target domains. Clustering users who interact under covariant combinations of contextual predicates in different domains lets us better incorporate their behavioral similarities and analogously for the item sets. The user and item representations in sparse domains can be significantly improved when we combine

[1] https://stackexchange.com/sites

these transferrable covariances and use them to group few-shot users and items with the pre-existing users and items that interact under similar context combinations.

Finally, we explore the multi-task dimension to complement the challenges and solutions described in the above subsections. In tandem with the strategies to learn descriptive and effective representations in the presence of sparsity and skew, we can explicitly leverage the correlations that exist across the different user/item inference and recommendation tasks. For instance, predicting a user's cuisine preferences on the Yelp platform and recommending suitable restaurants are correlated tasks and benefit from a joint treatment via shared representational aspects. We now describe a domain-agnostic generalizable solution to leverage shared characteristics across multiple predictions, inference, and recommendation tasks to mitigate skewed and sparse behavioral data.

1.5.5 Multi-Task User and Item Representations

Online platforms often incorporate multiple distinct recommendations and prediction tasks associated with each user or item in their inventories. While there is reason to believe that users exhibit correlated behavior across the different tasks [150], the extent of correlation varies on a per-user basis [67]. Thus, learning a single user representation is insufficient to learn the user-specific eccentricities, although learning isolated representations does not leverage shared knowledge to benefit task performance mutually. We highlight the need to enable shared components in tandem with task-specific representations to independently assess the extent of shared knowledge for each user or item on the platform.

We ground our multi-task representations with a shared heterogeneous knowledge graph across all the inference and recommendation tasks, which provides us a highly expressive data representation for specialized domains and applications with multimodal data ranging from linguistics [220] and biomedicine [42] to finance [27] via interacting entities (nodes) and relationships (edges). Knowledge graphs enrich representation models by explicitly encoding the rich transitive association structure across diverse interacting entities. The structural properties of the graph are not specific to any of the inference tasks [38], and hence form a suitable basis for the shared components.

However, the graph's discrete link structure is not suited to knowledge sharing with gradient-updated inference and recommendation models. We thus transform the graph to a continuous embedding space which provides a shared representation across all tasks while retaining the association structure by encoding and stacking heuristic patterns such as *symmetry, antisymmetry, analogy, inversion* and *composition* [193].

We propose a unified framework for knowledge graph representation (the shared com-

ponent) and multi-task learning (task-specific transformation of the shared element) that permits the bidirectional transfer of knowledge between the graph and the different inference and recommendation tasks. The modular decoupling of the transformation functions and the underlying graph embeddings overcomes the limiting assumptions of past work that restrict the direction or type of knowledge transfer [21, 71]. Specifically, we demonstrate the utility of learning task-specific residual functions owing to their simplicity and optimization advantages [56]. The resulting components admit effective multi-task representations.

1.6 ORGANIZATION OF THIS THESIS

The rest of the thesis is organized as follows. In the next chapter, we closely study multiple related threads of prior work across several machine learning domains and identify their commonalities and disparities compared to recommender models and applications. We thematically position our contributions in the context of previous work.

In Chapter 3, we describe our unified approach to mitigating behavior skew and data sparsity with user interaction data. We develop an iterative optimization framework that couples user grouping with behavior profile learning in a skew-aware manner, fitting groups' sizes to the user data's aggregate distributional characteristics. Subsequently, in Chapter 4, we address the similar item-side challenges with massive (infinite) inventory recommendation. Learning descriptive item representations also benefits the grouping mechanism proposed in Chapter 3 towards profile learning.

While the models proposed in Chapter 3 and Chapter 4 cluster the user and item set towards mitigating skew and sparsity, they do not account for the multimodal scenario where the cluster structure may vary across the different data modalities. In Chapter 5, we develop a modular adversarial framework to integrate diverse modeling hypotheses across the data modalities and learn aggregate task-driven representations of users and items.

Chapter 6 tackles the cross-domain setting, where the sparse target-domains offer very limited user and item interaction histories. We develop a transferrable neural framework that relies on interaction context to leverage the cluster structure of users and items in the dense-domain, adapted to learn sparse-domain representations. Chapter 7 further extends our modeling solutions to the multi-task setting, where the learned representations are simultaneously leveraged towards more than one recommendation or prediction task. We propose an efficient residual learning strategy to leverage cross-task similarities.

Finally, in Chapter 8, we conclude this thesis by discussing our findings and presenting promising future work avenues.

CHAPTER 2: AN OVERVIEW OF PRIOR WORK

This chapter provides an overview of related prior work and perspectives to address the data skew and sparsity challenges in diverse machine learning domains. We primarily focus on neural models, architectures, and representations. However, the proposed techniques apply to most gradient-updated representation learning models. We study prior work towards the following broad objectives.

Skew and sparsity-aware user inference tasks and recommendation: We first analyze machine learning models that learn with distributionally skewed and sparse training data from the perspective of generalizability. Generalizability refers to applying directly or trivially adopting the proposed solutions to a different data modality or application scenario. For example, power-law skew/Zipf's law appears in both signed and unsigned social networks. However, these two types of social networks leverage distinct models and neural architectures for inferencing and representation learning tasks (e.g., graph convolution networks [89] and signed graph convolution network [34] variants). Hence, we identify common frameworks to address the shared sparsity and skew challenge independent of the specific model architecture. Our approaches are built upon generalizable abstractions to achieve such subsumptive modeling.

Incoporating multimodal data towards inference: We aim to provide learner guidance when we have more than one independently generated source of information or modality, as described in Section 1.2.1. Specifically, we identify a few broad frameworks: techniques that align latent entity representation across data sources and enable mutual regularization (e.g., *conditioned representations, metric learning, joint clustering regularizers*); techniques that help us learn diversified or disentangled representations for entities across two data sources, such that the respective representation spaces explain different underlying facets of each entity towards the inference objective (e.g., *adversarial disentanglement, boosting*); data, loss-function or training augmentation strategies.

We provide an overview of multiple distinct lines of machine learning in Section 2.1, and analyze how our work augments, bridges and adapts recurring themes and approaches towards recommendation and personalized inference tasks in Section 2.2.

2.1 SKEW AND SPARSITY-AWARE MODEL DESIGN

We emphasize a few broad limitations of the diverse models described in Table 2.1, especially when we work with sparse and skewed training samples, as with user data.

Table 2.1: We classify models along two axes: *Axis 1*: Unimodal models that focus a specific data or feature modality vs. multimodal models that address inference tasks involving more than one modality of features or training data. *Axis 2*: We further classify models and learning algorithms by their ability to account for skewed feature or interaction distributions in the training data. Note that we include both unsupervised and supervised representation models.

	Unimodal Models	**Multimodal models**
Does not account for data distribution	Denoising auto-encoder [224]	Collaborative denoising auto-encoder [109]
	Poisson point-process [64]	*k-step* factor-graph [158]
	N-gram language models [169]	Corpus-guided image caption [231]
	LDA topic-model [17]	Topic-link LDA [120]
	Deepwalk [153]	Metapath vector representation [38]
Learns data distribution parameters	Variational auto-encoder [115]	Collaborative variational auto-encoder [111]
	Recurrent neural point-process [226]	Dual-stream recurrent neural nets [157]
	Embedding-based language models [46]	Dependency-based word embeddings [104]
	Pitman-Yor topic-model [180]	Dependent Pitman-Yor model [190]
	Graph tail-node regression [122]	Graph tail-node regression [122] (also applies to heterogenous graphs)

Implicit apriori hypotheses: Denoising auto-encoders [224] find application across a wide range of representation learning tasks with diverse machine learning objectives [51, 224]. Despite this diversity, they apply pre-defined static noise functions to corrupt input feature representations and learn a robust encoding by extracting the denoised version. However, the effectiveness of the chosen noise function is reliant on the input feature distribution. Analogously, the *poisson point process* is inapplicable to many recommendation scenarios owing to its intensity function. The self-exciting process may be more appropriate to capture user interaction densities [39].

Distributional mismatch: Although topic-models are able to account for co-occurrences and multimodal data, unlike *n-gram* models, the commonly employed *dirichlet priors* are unsuited to skewed data owing to their mean-heavy distribution. The skew challenge manifests in graph representation applications as well [153] since social networks exhibit power-law skews in their node degrees resulting in uninformative representations for sparse nodes.

Insufficient expressivity: *N-gram* models (and the relevant back-off models [85]) do not consider the implicit similarities and dissimilarities of the entities in the corpus towards

inference. Analogously, the *poisson point process* does not learn the sequential autocorrelations associated with online user behavior [158] since it lacks interaction-specific expressivity in its intensity function. This challenge is also common in multimodal recommendation models that cannot account for the diversity of users across different data modalities [114]. We consider a few alternate models that overcome these limitations:

Data-driven clustering: Unlike denoising auto-encoders, variational auto-encoders [115] decouple the direct link between the input feature representation and the latent space encoding. Instead, the input features are employed to select an appropriate distribution, from which the encoding is then drawn. Thus, the latent space clusters adapt to the input feature distributions. Similarly, non-parametric Pitman-Yor models introduce data-driven clustering mechanism in both unimodal and multimodal scenarios [180, 190].

Learnable meta-parameters: The Pitman-Yor topic model [180] incorporates learnable grouping parameters, while sparse (by degree) node representations in graphs are handled via few-shot regression on their neighbor node embeddings [122]. In both cases, the meta-parameters are data-driven. They do not impose static pre-defined structural constraints towards latent cluster formation.

Data and task-dependent expressivity: Recurrent point processes [226] learn intensity functions conditioned on each input sequence to permit modeling a wide range of sequence-based applications. Analogously, sentence embedding models [91] account for varied word co-occurrence frequencies, as well as the sequential ordering of words towards inference tasks.

2.1.1 Multimodal Extensions

We also reference multimodal modeling approaches across several distinct domains in Table 2.1. We identify a few common themes and challenges across multimodal models:

Shared and independent representations: Although joint training leads to knowledge transfer across data sources, maintaining shared entity representations is restrictive. It may cause overfitting to either source in the presence of activity skew. Shared representations refer to parameters reused across the architectures or models associated with the different data modalities. While multimodal generative approaches often share hyperparameters or distribution parameters across modalities [159, 160], neural methods instead incorporate shared neural layers and transform functions [191].

Alignment vs. Fusion vs. Disentanglement: Creating cross-modality similarity functions is a challenging task, especially in the context of recommendation, where the distributional aspects of each modality may vary significantly. As a result, hard alignment

strategies and static metrics [76, 129] are often too restrictive to be included in generalizable strategies or solutions. A natural pivot for representation alignment is the user, the shared entity across multiple modalities of data. However, users may not exhibit correlated cross-modal behavior [77]. Thus, any representational alignment strategies should incorporate learnable components that permit per-user reweighting of the data modalities towards aggregate representations [97].

Approaches that fuse multimodal representations to generate aggregate representations assume implicit correlations [182] and fail when there are significant distributional/semantic differences [146] across the chosen data modalities. In such settings, it is sometimes beneficial to explicitly employ a disentanglement objective to avoid uninformative representations and/or overfitting to trivial patterns in the data [163]. We can view fusion and disentanglement as two sides of the coin. One explains each sample with all modalities of data simultaneously. In contrast, the second explains each sample with exactly one modality of data. As a result, it may be beneficial to employ disentanglement strategies in the initial representations and fusion methods in the final prediction or propensity estimation.

Avoiding mode collapse: While the phrase *mode collapse* is typically used in the context of generative adversarial networks [187], the broader challenge applies to iterative optimization strategies that attempt to co-learn representation models across two modalities of data, specifically when users are the entities linking the two modalities. We identify that an independent representation of user data, separate from the two adversarial modalities, can serve as a tiebreaker/attribution strategy to avoid degenerate solutions that result in disregarding the data in either modality [97].

2.1.2 Sparsity Mitigation Strategies

Next, we consider the data sparsity aspect. In the context of neural models, sparsity mitigation strategies can be broadly partitioned into three buckets: *data / training-approach / loss-function augmentation* to improve or stabilize learning, prioritize informative data points either via static or progressive criteria, *representation clustering methods* and embedding re-arrangement via *alignment / co-learning / regularization strategies*. Each bucket can be further sub-categorized based on the criteria for augmentation, cluster formation, or the nature of the imposed representation alignment / regularization functions, respectively.

Data augmentation and sample reweighting: Sample reweighting or loss augmentation strategies can be leveraged to bridge data sparsity by identifying and emphasizing the most important or informative training points [186]. Reweighting can be achieved with both static pre-defined criteria for informativeness [233] or dynamically updated weights condi-

Table 2.2: We classify sparsity-aware learning strategies along two axes: *Axis 1*: Unimodal or multimodal approaches to address sparse inference tasks. *Axis 2*: We further classify the adopted sparsity mitigation strategies by their adaptation to the distributional aspects of the training data and input features.

	Unimodal Models	**Multimodal models**
Does not account for data distribution	Co-occurrence regularizer [234]	Cross-modal representation alignment [129, 241]
	Data-augmentation [138]	Multimodal data-augmentation [70]
	Noise contrastive estimation with static samplers [164]	Pseudo-relevance feedback negative samples [238]
	Oversampling and undersampling [233]	Synthetic sampling in each data modality [55]
Learns data distribution parameters	Self-supervised association learning [95]	Multimodal neighbor models [208]
	Margin-based active learning [10]	Cross-modal mutual information maximization [207]
	Negative example mining [197]	Modality disentanglement [163]
	Sample informativeness [20]	Multimodal representation fusion [182]
	External Regularizer [77]	Interaction context regularizers [137]

tioned on the model-state [127]. Analogously, data-augmentation methods seek to bridge sparsity via synthetic samples that selectively replicate the most important training samples, either on a task-specific basis [23], based on measures of hardness [37] or hybrid strategies.

Clustering via static alignment / regularizers vs. learnable metrics: Includes methods that learn representations based on data-driven alignment metrics [63] and shape or distribution hypotheses on the embedding space, e.g., hyperbolic embedding spaces combine structural information such as taxonomies with embedding spaces [24]. On the other hand, the proximity metrics may incorporate learnable parameters, e.g., bilinear alignment functions [228]. Non-parametric clustering methods are an alternate class of methods that expand their parameter sets with the available data [190]. With graph representations of data, path-based heuristics [193] may be applied to enrich the set of functional associations across the entities towards representation learning and to infer their relations in heterogeneous, multi-relational graphs [75].

Leveraging external data vs. interaction context: External data sources are typically obtained from a different platform [77] and employed only to regularize entity representations, unlike participation modalities on the same platform. On the other hand,

interaction context is specific to each interaction modality (such as user-item interactions in recommendation tasks and user-user interactions on social networks). It helps us attribute each interaction to specific subsets of features [137].

2.1.3 Cross-Domain / Transfer-Learning, Multi-Task Learning, and Meta-Learning

Cross-domain learning broadly encompasses models and algorithms that adapt from one domain to another domain sharing similar data characteristics (not necessarily distribution characteristics, as we show in Chapter 6). Most prior work along this line is only focused on learning from a single source domain to a target domain. Structure transfer methods regularize the embedding subspace structure via components [107, 150], joint factorization [79, 118], shared and domain-specific cluster structure [48, 152] or unified prediction tasks [97, 179]. Co-clustering methods leverage shared entities as anchors for sparse domain inference [132, 218]. However, recommendation domains may often encompass disjoint sets of items and users (Section 1.2.2).

In particular, an unanswered challenge is how to apply the knowledge learned from a single dense source domain to many non-overlapping target domains, where each target domain may encompass slightly different distributions. We focus on two avenues of progress in Chapter 6: *Input adaptation* and *conditional adaptation* and study the benefits of direct parameter sharing, among other scalability criteria such as few-shot learning in each target domain. The term *transfer learning* [150] is sometimes used to refer to cross-domain or multi-task learning.

Multi-task learning (MTL), unlike cross-domain learning, typically refers to more than one task concerning the same underlying entity sets and finds applications in domains such as natural language processing, speech recognition, computer vision, and drug discovery [172]. MTL encompasses multiple optimization forms: episodic training or learning-to-learn [45] as in meta-learning algorithms, joint learning of more than one task, and parameter sharing across auxiliary tasks. Generally, MTL algorithms involve optimizing more than one loss function iteratively, in contrast to single-task learning.

Aforementioned also includes hard-parameter sharing or alignment across multiple tasks [13], or soft parameter sharing via learnable alignments, e.g., iterative multimodel optimizations or bilinear representation alignments [111, 228]. Note that soft sharing incurs parameter overheads. Thus, the key questions that we identify in the context of multi-task learning are two-fold: How do we *mutually leverage independent task-models* in an architecture / loss-function agnostic manner, and how do we minimize the overall parameter overhead while still enabling sufficient task-model expressivity? We attempt to trade off these two essential

criteria by leveraging knowledge graph representations in Chapter 7.

In contrast to cross-domain and multi-task learning, *meta-learning* modifies the learning algorithm via multiple learning episodes, or gradient criteria [45, 101]. Gradient-based models analyze the learning models' plasticity to new data samples sampled from the task-distribution (or domains) and optimize improving model initializations. However, gradient models are typically constrained to architecturally simple models [192] with multi-task training samples, and hence unsuited to the complex multimodal recommendation scenario. In Chapter 6, we develop a combined approach unifying aspects of meta-learning and transfer-learning to address both the scalability challenge and the one source, many target-domain challenges in recommendation.

2.2 POSITIONING THE CONTRIBUTIONS OF THIS THESIS

In Table 2.3, we revisit this thesis's contributions in the context of the above sparsity and skew-aware modeling approaches described in Section 2.1. We show how our work bridges multiple related work themes and addresses their limitations towards sparsity and skew-aware recommendation and personalized inference tasks while maintaining architecture-agnostic generalization and broad applicability.

Table 2.3: We describe the application scenarios and approaches towards mitigating data sparsity and skew. *Axis 1*: Application scenarios of our models, *Axis 2*: Broad themes of each chapter, and the modeling aspects bridged or incorporated by our work.

	Unimodal Learning	**Multimodal Learning**	**Cross-domain Learning**	**Multi-task Learning**
Adaptive representation clusters	ch3, ch4	ch4, ch5	ch6	
Adaptive noise contrastive estimation	ch4	ch3		
Learnable representation alignments	ch4	ch5, ch7		ch7
Parameter sharing		ch6, ch7	ch6	ch7
Meta-learning		ch6	ch6	
Architecure and platform agnostic approach	ch4	ch3, ch5, ch7	ch6	ch7

We adopt multiple strategies to adaptively cluster the user and item representations spaces towards inference and recommendation tasks. In Chapter 3, we show how a skew-aware non-parametric clustering process can be coupled with generative models of user behavior, resulting in skew-aware clusters of users described by similar profiles. Unlike prior approaches that either cluster with mean-heavy priors [159, 160] or do not mitigate skew and sparsity jointly [14], our approach handles sparse users even in the presence of aggregate behavior skew. In Chapter 4 we guide the item representation clusters using aggregate co-occurrence frequencies and co-learn users' preferences. In Chapter 5, we show how to jointly cluster users across two representation spaces while avoiding mode-collapse (Section 2.1.1). We also describe contextual invariants to enable cross-domain *invariant-driven* clustering in Chapter 6.

In Chapter 4 and Chapter 5, the clustering strategy's adaptive aspect derives from adaptive noise-contrastive estimation. We select or generate the best negative samples or cross-modal samples to accelerate the cluster learning process. We also leverage an architecture-agnostic strategy. The samples' quality is data-driven and does not rely on any specific modeling hypotheses across the user participation modalities. The architecture-agnostic adaptive clustering strategy, as well as the negative samples, leverage learnable cross-modal alignment functions. This accounts for the distributional heterogeneities introduced by uneven user participation across the data modalities.

We also consistently maintain parsimony in our frameworks. In the cross-domain and multi-task scenarios, soft parameter-sharing strategies result in parameter duplication. However, hard parameter-sharing across domains or tasks severely restricts the expressivity of the joint model. To overcome these challenges, we make two key contributions. In Chapter 6, we deal with the one source to many target parameter-sharing challenges by altering the *input distribution* to the shared modules to account for target domain heterogeneity, rather than learning an alternate set of parameters from scratch. Analogously, we describe an inexpensive residual learning strategy in Chapter 7 to account for various task distributions, analogous to the domain-specific distribution adaptations in Chapter 6.

Model architecture and modality agnostic strategies: Our abstractions do not limit the kinds of data modalities or model architectures that can be applied towards learning user or item representations. In Chapter 3, the skew-aware grouping mechanism can incorporate any generative profile model to describe user data. The cluster structure then adapts to the specific profile model. In Chapter 5, the proposed adaptive noise contrastive estimation strategy does not introduce any specific architectural constraints. We can thus leverage the user representations learned by any differentiable gradient model across the competing data modalities. Analogously, the invariant extraction strategy defined in Chapter 6 only

requires the context module to incorporate a specific architecture to enable multiplicative interactions. The user and item representation models and the ranking and clustering models may be modified without impacting cross-domain module transfer. Finally, in Chapter 7, we enable multi-task residuals to adapt to the task distributions agnostic to the specific models (and their inductive biases) that generate the distributions.

CHAPTER 3: LEARNING USER PROFILES BY JOINTLY MITIGATING SPARSITY AND DISTRIBUTIONAL SKEW

This chapter proposes an approach to learn robust behavior representations in online platforms by addressing the challenges of user behavior skew and sparse participation. Latent behavior models are essential in various applications: recommender systems, prediction, user profiling, community characterization. Our framework is the first to address skew and sparsity across graphical behavior models jointly. We propose a generalizable bayesian approach to partition users in the presence of skew while simultaneously learning latent behavior profiles over these partitions to address user-level sparsity. Our behavior profiles incorporate the temporal activity and links between participants, although the proposed framework is flexible in introducing other participant behavior definitions. Our approach explicitly discounts frequent behaviors and learns variable size partitions capturing diverse behavior trends. The partitioning approach is data-driven with no rigid assumptions, adapting to varying degrees of skew and sparsity.

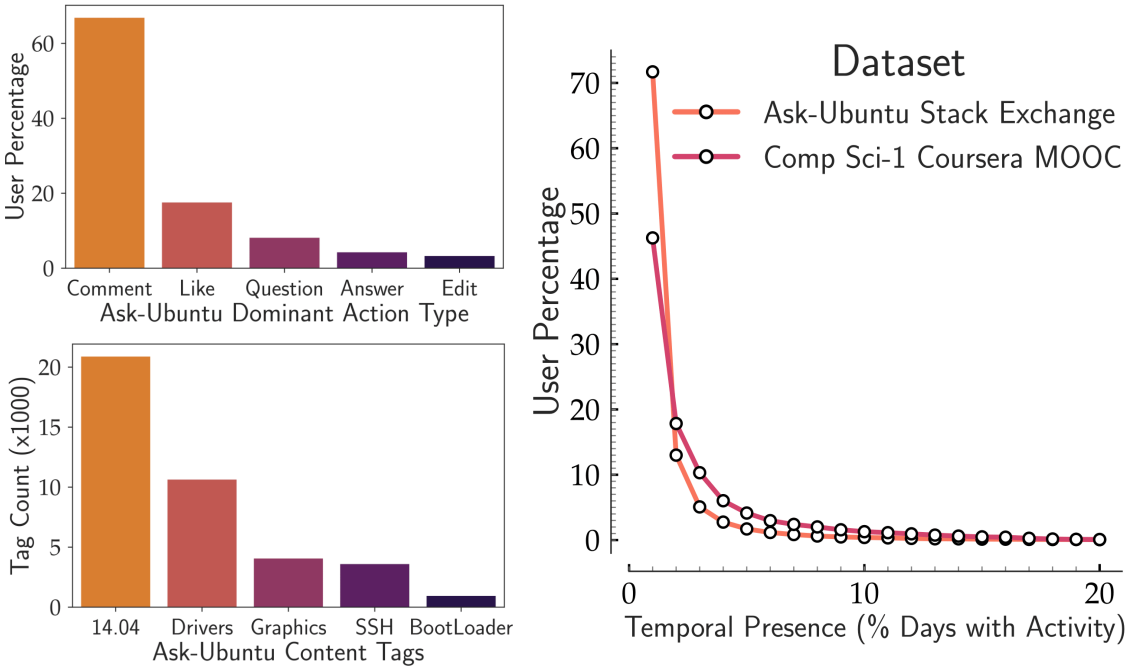
Qualitative analysis indicates our ability to discover niche and informative user groups on large online platforms. Results on User Characterization (+6-22% AUC); Content Recommendation (+6-43% AUC) and Future Activity Prediction (+12-25% RMSE) indicate significant gains over state-of-the-art baselines. Furthermore, we validate user cluster quality with magnified gains in the characterization of users with sparse activity.

3.1 INTRODUCTION

This chapter addresses the challenge of learning robust statistical representations of participant behavior on online social networks. Graphical behavior models have found success in several social media applications: content recommendation [159, 230], behavior prediction [160, 237], user characterization [131] and community profiling [19]. Despite the large sizes of these social networks (e.g., several million users), developing robust behavior profiles is challenging. We know from prior work [12] that activity on online networks is heavy-tailed (a small set of users account for most interactions) with several temporally sparse users. Furthermore, user activity styles and topical interests are highly skewed (imbalanced) within the population, complicating the inference of prototypical behavior types. Figure 3.1 shows a typical example of behavior skew and temporal sparsity in `AskUbuntu`[1], a popular online Q&A forum.

[1] `https://askubuntu.com/`

Figure 3.1: Dominant Action Types and Content are highly skewed in Ask-Ubuntu, User presence exhibits steep power-law ($\eta \approx 3$) indicating several inconsistent or inactive users.



Past works address one of the challenges (either sparsity or skew) separately in graphical behavior models but do not adopt a unified approach to learn representations. Clustering is one common way to address sparsity [161, 227]. However, using clustering techniques in the presence of behavior skew can lead to uninformative results. For example, when topic models do not account for skew (e.g., Zipf's law), the resulting topics are less descriptive [180].

The use of suitable priors over the cluster sizes is a way to deal with skew. Beutel et al. [14] propose the use of the Pitman-Yor process [154] (visualized via Chinese Restaurant Process; CRP) to model skew in user data. However, a direct application of the CRP-prior to behavior models cannot address sparsity. This is because behavior profiles are still learned at the user level. Inactive users degrade the ability to learn robust latent representations; a lack of robust representations affects cluster quality.

Our main technical insight: We need to address behavior skew and temporal sparsity of inactive users simultaneously. While we exploit the Pitman-Yor process (or CRP) as a prior, our key innovation in addressing sparsity and behavior skew lies in how we "seat" users onto tables. Our intuition is to associate inactive users with those active users to whom they were most similar when these sparse users were active. Thus, to address sparsity, we identify three concrete lines of attack: Profiles need to be learned from data at the granularity of a table (or equivalently, a group of users), *not* at the user-level; Behavioral similarity should guide

user seating on these tables; We should discount typical behavioral profiles to encourage identification of niche behaviors in the presence of skew. We refer to our model as CMAP (CRP-based Multi-Facet Activity Profiling) in the rest of this chapter. To summarize our contributions:

Jointly address skew and sparsity: To the best of our knowledge, this is the first work to *jointly* address behavior skew and sparsity with graphical behavior models. Our partitioning scheme can adapt to varying levels of behavior skews, effectively uncover fine-grained or niche behavior profiles, and address user-level sparsity.

Generalizability: While in this work, we employ user activity and knowledge-exchanges, our framework generalizes well. The constituents of a behavioral profile can be easily adapted to new applications and platforms while retaining skew and sparsity awareness in the learning process.

Efficiency: Our model is efficient: the computational complexity is linear in the number of users and interactions. We employ caching optimizations to speed up inference and scale to massive datasets with parallelized batch sampling.

We show strong quantitative and qualitative results on diverse datasets (public StackExchange datasets and Coursera MOOCs[2]). We chose our datasets across technical/non-technical subject domains and varying population sizes, with all datasets seen to exhibit significant behavioral skew and sparsity (table 3.5). We evaluate CMAP against state-of-the-art baselines on three familiar task types: user characterization (reputation; certificate prediction on MOOCs), content recommendation, and future activity prediction. Through extensive qualitative analysis, we find CMAP gains to be most significant for sparse users. The behavioral profiles learned are coherent and varied in size, capturing underlying behavioral skew. Our results have an impact on the practical realities of large-scale social network dataset analyses since successfully addressing behavioral skew and sparsity is critical to familiar applications such as behavioral profiling and content recommendation.

We organize the rest of the chapter as follows. In Section 3.2 we discuss related work. We formally define the problem and proposed approach in Section 3.3 and Section 3.4. We then discuss model inference, datasets and results in Sections Section 3.5, Section 3.6 and Section 3.7, concluding in Section 3.8.

3.2 RELATED WORK

At a high level, our motivations are shared with skew-aware topic models to improve document representation [180] by accounting for Zipf's law, and short-text clustering methods

[2] `https://stackexchange.com`, `https://coursera.org`

[161, 236] to address content sparsity in text snippets. Graphical behavior models employ simple Dirichlet priors in user profile assignments [131, 159, 160]. However, this setting is limited in its ability to model behavior skew and cannot cleanly separate niche and typical behavior profiles. Our qualitative results (section 3.7.4) reflect this observation.

In collaborative filtering, efforts have been made to transfer the user-item latent structure across platforms [41, 150] via consensus models to tackle sparsity. In the implicit feedback setting, this approach assumes alignment of user behavior across platforms. However, user interests and consumption trends vary not just by platform, but action-type and time as well [77, 240]. User anonymization (such as in MOOCs) can also pose difficulties in acquiring cross-platform data. We choose not to rely on external data.

Beutel et al. [14] propose a bayesian approach to group users with limited rating information and capture skewed product ratings. While the direct application of Pitman-Yor priors [154] to group users can capture skew in cluster sizes, it does not address the inactive user problem. In contrast, we factor in the latent behavior profiles in the seating to address sparsity via joint profiling of users [227]. The skew-aware partitioning and profile learning tasks are deeply coupled, unlike the superficial connection in past work.

Recently, Jiang et al. [77] proposed sparsity-aware tensor factorization for user behavior analysis. User representations are regularized with external data such as author-author citations in academic networks, however, not accounting for behavior skew. Behavior Factorization [240] simultaneously factorizes action-specific content affinities of users. Quadratic scaling imposes computational limits on these methods. Deep recurrent networks have also been explored to model temporal student behavior on MOOCs [158].

We choose FEMA [77] (Sparsity-aware Tensor Factorization), BLDA [159] (LDA-based generative user model) and LadFG [158] (Deep Recurrent network for temporal activity and attributes) as representative baselines for comparison with our approach. Table 3.2 provides a summary of the aspects addressed by each model.

3.3 PROBLEM DEFINITION

We apply our approach to learn representations of user behavior in multiple Coursera MOOCs and Stack-Exchange Q&A websites. The available facets of user activity include textual content, actions, time, and inter-participant knowledge-exchanges.

Let $\mathcal{U}$ denote user set in a Stack-Exchange or MOOC dataset. Users employ a set of discrete actions $\mathcal{A}$ to interact with content generated from vocabulary $\mathcal{V}$. A user interaction d (atomic unit of participant activity) Is a tuple $d = (a, W, t)$, where the user performs action $a \in \mathcal{A}$ on content $W = \{w_1, w_2 \ldots \mid w_i \in \mathcal{V}\}$ at time-stamp $t \in [0, 1]$ (normalized over the

Table 3.1: Comparing the data-challenges addressed by baseline models with our proposed approach (**CMAP**).

Aspect	**BLDA**	**LadFG**	**FEMA**	**CMAP**
Skew-aware	No	No	No	Yes, via CRP
User-level Sparsity	No	No	External Regularizer	Profile-based Clustering
Multi-facet	Limited to Text/Action	Yes	Yes	Yes
Integrate with latent models	Limited to Text/Action	No	No	Yes
Runtime	Linear	Linear	Quadratic	Linear

time-span of the activity logs). We denote the set of all interactions of $u \in \mathcal{U}$ as $\mathcal{D}_u$. Thus the collection of interactions in the dataset is $\mathcal{D} = \bigcup_{u \in \mathcal{U}} \mathcal{D}_u$. The action set for each dataset is described in table 3.4. Lecture interaction content for MOOC datasets is extracted from the respective transcripts.

Inter-participant knowledge-exchanges are represented by a directed multigraph $G = (\mathcal{U}, E)$. A directed labeled edge $(u, v, \ell) \in E$ represents an interaction of user u, $d_u \in \mathcal{D}_u$(e.g. "answer") that is in response to an interaction of user v, $d_v \in \mathcal{D}_v$ (e.g. "ask question") with label $\ell \in \mathcal{L}$ indicating the nature of the exchange (e.g. "answer" $\rightarrow$ "question"). We denote the set of all exchanges in which participant u is involved by L_u, so that $E = \bigcup_{u \in \mathcal{U}} L_u$.

Our goal is to obtain a set of temporal activity profiles R describing facets of user behavior and infer user representations $\mathcal{P}_u, u \in \mathcal{U}$ as a mixture over the inferred behavior profiles $r \in R$.

3.4 OUR APPROACH

We begin in section 3.4.1 with intuitions to jointly address the behavior skew and sparsity challenges. In section 3.4.2, we describe a skew-aware user seating model guided by behavior profiles, concluding in section 3.4.3 with a description of our profile model.

3.4.1 Attacking the Skew-Sparsity Challenge

We begin by formally discussing the Pitman-Yor process [154] and then highlight challenges in the presence of sparsity.

Beutel et al. [14] employed the Pitman-Yor process via Chinese restaurant seating [4], as a simple prior over clusters to identify skewed user data trends. The conventional Chinese Restaurant arrangement induces a non-parametric prior over integer partitions (or indistinguishable entities), with concentration γ, discount δ, and base distribution G_0, to seat users across tables (partitions). Each user is either seated on an existing table $x \in \{1, \ldots, \chi\}$, or assigned a new table $\chi + 1$ as follows:

$$p(x \mid u) \propto \begin{cases} \frac{n_x - \delta}{N + \gamma}, & x \in \{1, \ldots, \chi\}, \text{ existing table,} \\ \frac{\gamma + \chi\delta}{N + \gamma}, & x = \chi + 1, \text{new table,} \end{cases} \tag{3.1}$$

where n_x is the user-count on existing tables $x \in \{1, \ldots, \chi\}$, $\chi + 1$ denotes a new table and $N = \sum_{x \in \{1,\ldots,\chi\}} n_x$ is the total user-count. A direct application of Equation (3.1) as a simple prior can address skew in profile proportions, but not sparsity. This is because, sparse users introduce noise into estimation of the corresponding behavioral profiles. To address sparsity, we need to find a way to "fill in the gaps" in our knowledge about inactive users. Section 3.4.1

Thus, to address sparsity, we identify three concrete lines of attack: Profiles need to be learned from data at the granularity of a table (or equivalently, a group of users), *not* at the level of an individual; Behavioral similarity should guide seating on these tables; We should discount typical behavioral profiles to encourage identification of niche behaviors and improve profile resolution.

3.4.2 Our Profile-Driven Seating

Now, we introduce our profile-driven seating approach that builds upon CRP to simultaneously generate partitions of similar users and learn behavior profiles describing users in these partitions. Consider a set of latent profiles $r \in R$ describing observed facets of user data with conditional likelihood $p(u \mid r)$ for $u \in \mathcal{U}$. We "serve" a profile $r_x \in R$ to users seated on each table $x \in \{1, \ldots, \chi\}$. A user u is seated on an existing table $x \in \{1, \ldots, \chi\}$ serving profile r_x or a new table $\chi + 1$ as follows,

$$p(x \mid u) \propto \begin{cases} \frac{n_x - \delta}{N + \gamma} \times p(u \mid r_x), & x \in \{1, \ldots, \chi\}, \\ \frac{\gamma + \chi\delta}{N + \gamma} \times \frac{1}{|R|} \sum_{r \in R} p(u \mid r), & x = \chi + 1. \end{cases} \tag{3.2}$$

Note that the likelihood $p(x \mid u)$ of choosing an *existing* table $x \in \{1, \ldots, \chi\}$ for user u

Table 3.2: Notations for our user seating arrangement.

Symbol	Description
N, R	Number of seated users, Set of profiles
$\{1, \ldots, \chi\}$, $\chi + 1$	Set of existing tables, New table
n_x, r_x	User count on table x, profile served on x
χ_r, N_r	Number of tables serving profile r, Total users seated on tables serving profile r

depends on the conditional $p(u \mid r_x)$ of the profile r_x served on the table and the number of users seated on table x. Further, the seating likelihoods for existing tables depend on the latent profiles served, while the latent profiles r_x are learned from the table x they are served on. This process introduces a mutual coupling between seating and profile learning.

The effect of discount parameter δ: A larger setting of the discount parameter δ encourages the formation of new tables, leading to a preference for exploration over exploitation in the profile learning process. In effect, the threshold for choosing to assign a new table to a user is lowered when the user is not described with a sufficiently high likelihood by the behavioral profiles served on the existing tables.

The likelihood of assigning the user to a new table $x = \chi + 1$ depends on the sum of conditionals $p(u \mid r)$ with a uniform prior $\frac{1}{|R|}$, and the number of existing tables χ. Notice the effect of the discount factor δ: increasing δ favors exploration by forming new tables. Niche users are likely to be seated separately with a different profile served to them.

Key modifications to CRP: The main difference with the basic CRP (also referred to as the *Stick-Breaking* process or the *Pitman-Yor* process) Equation (3.1), which partitions users based on the table size distribution, is that in our approach, we seat users based on the table size distribution, the profiles served on those tables, and the conditional probability of the user given the served behavioral profile.

Equation (3.2) reduces to Equation (3.1) when all profiles $r \in R$ are equally likely for every user. We can show that our seating process is exchangeable, similar to [4]. The likelihood of a seating arrangement does not depend on the order in which we seat users on the tables.

When user u is seated on a new table $\chi + 1$, we draw profile variable $r_{\chi+1} \in R$ on the new table as follows:

$$p(r_{\chi+1} \mid u) \sim p(u \mid r)p(r), \tag{3.3}$$

where $p(r)$ is the Pitman-Yor base distribution G_0, or prior over the set of profiles. We set G_0 to be uniform to avoid bias.

The likelihood $p(r \mid u)$ of assigning profile r when seating user u, is proportional to the

sum of likelihoods of seating the user on an existing table $x \in \{1, \ldots, \chi\}$ serving profile r (i.e. $r_x = r$), or seating on a new table $\chi + 1$ with the profile $r_{\chi+1} = r$. That is:

$$p(r \mid u) \propto \left(\sum_{\substack{x \in \{1,\ldots,\chi\}, \\ r_x = r}} \frac{n_x - \delta}{N + \gamma} p(u \mid r) \right) + \frac{1}{|R|} \cdot \frac{\gamma + \chi\delta}{N + \gamma} p(u \mid r), \tag{3.4}$$

$$\propto \left(\frac{N_r - \chi_r \delta}{N + \gamma} + \frac{\gamma + \chi\delta}{|R|(N + \gamma)} \right) p(u \mid r), \tag{3.5}$$

where χ_r is the number of existing partitions serving profile r and N_r is the total number of users seated on tables serving profile r.

Three insights stem from Equation (3.5). First, the skew in profile sizes depends on the counts of users exhibiting similar behavior patterns ($\propto p(u \mid r)$) enabling adaptive fits unlike Beutel et al. [14]. Second, we discount common profiles served on multiple tables by the product $\chi_r\delta$. Since χ_r is larger for common profiles drawn on many tables, we discount common profiles more than niche profiles. This "common profile discounting" enables us to learn behavioral profile variations. Finally, not constraining the number of tables introduces stochasticity in profile learning and encourages exploration.

We assign users with limited activity to tables that well explain their data, biased by the priors in Equation (3.5). Our partitioning scheme assigns the *same* profile to users sharing a table, reducing the effect of inactive users since profiles describe behavioral groups.

In the next subsection, we introduce our temporal activity profiles $r \in R$ for representing user activity in our datasets.

3.4.3 Latent Profile Description

In this section, we formally define our behavioral profiles to describe user activity. We reiterate that our framework is flexible to other profile definitions depending on the requirements. In our datasets, behavioral profiles ($r \in R$) encode what actions users take (e.g., comment on a question), associated content (e.g., the topic of the question), and when they take them. Furthermore, users participate in conversations (e.g., answer in response to a question), we term these directed links as "knowledge exchange."

Our profiles thus have two constituents: Joint associations of actions and words; referred to as "action-topics", and temporal distributions indicating when the action-topics are executed. Each action-topic $k \in K$ models user actions and the associated words, with $\phi_k^{\mathcal{V}}$ (multinomial over words with vocabulary $\mathcal{V}$) and $\phi_k^{\mathcal{A}}$ (multinomial over actions $\mathcal{A}$). Motivated by Wang and McCallum [217], we employ a continuous time model, Beta$(\alpha_{r,k}, \beta_{r,k})$

Algorithm 3.1: *Behavior profile and action-topic generation process.*

function GENERATE PROFILES(*Prior parameters*)
  for $k \in K$ **do** ▷ Draw the action-topics
    $\phi_k^{\mathcal{V}} \sim Dir(\alpha_{\mathcal{V}})$ ▷ Word distributions
    $\phi_k^{\mathcal{A}} \sim Dir(\alpha_{\mathcal{A}})$ ▷ Action distributions
  for $r \in R$ **do** ▷ Draw the activity profiles
    $\phi_r^K \sim Dir(\alpha_K)$ ▷ Split over action-topics
    for $r' \in R$ **do**
      $\phi_{r,r'}^{\mathcal{L}} \sim Dir(\alpha_{\mathcal{L}})$ ▷ Knowledge exchange from $r \to r'$
  return K, R

distributions, over a normalized time span to capture the temporal trend of each action-topic k within *each* profile r.

Thus for any interaction $d = (a, W, t)$, the probability $p(d \mid r, k)$ of a user interaction d given a profile r and topic k is:

$$p(d \mid r, k) \propto \underbrace{\phi_k^{\mathcal{A}}(a) \prod_{w \in W} \phi_k^{\mathcal{V}}(w)}_{\text{'what': profile independent}} \times \underbrace{\frac{t^{\alpha_{r,k}-1}(1-t)^{\beta_{r,k}-1}}{\mathrm{B}(\alpha_{r,k}, \beta_{r,k})}}_{\text{'when': profile dependent}}, \tag{3.6}$$

where B refers to the beta function.

Notice that while the action-topics are shared between profiles, each profile r has a *different* temporal distribution associated with each action topic. I.e., there are K action topics, but $R \times K$ temporal distributions. This modeling choice allows users with different overall behavioral profiles to participate in the same action topic at different times.

Since each behavioral profile r is a mixture over the K action topics and the associated temporal distributions, the likelihood $p(d \mid r)$ of user interaction d (as defined in section 3.3) for profile r is:

$$p(d \mid r) \propto \sum_k p(d \mid r, k) \times \phi_r^K(k), \tag{3.7}$$

where $\phi_r^K(k)$ is a K dimensional multinomial mixture over action-topics for each profile.

The next modeling step is to capture the exchange of knowledge between users. Instead of modeling it at the level of every pair of users, we model relationships between the pairs of profiles (r, r'), since every user is assigned to a single profile. This modeling choice is guided by sparsity. If we model every pair of users, we will develop a poor understanding of pairwise user interactions, owing to the heavy-tailed activity distribution (i.e., most users contribute little; c.f. Figure 3.1).

Algorithm 3.2: *Drawing user data $\mathcal{D}_u$, L_u from behavior profiles $r \in R$ served on the user's assigned table.*

function GENERATE INTERACTIONS($\mathcal{U}$, $\mathcal{D}_u$, L_u)
  K, $R \leftarrow$ GENERATE PROFILES(Prior parameters)
  for $u \in \mathcal{U}$ **do** ▷ Iterate over the user set
    $r_u \leftarrow r_x$ ▷ The user is seated on table x as in Equation (3.2)
    for $d = (a, W, t) \in \mathcal{D}_u$ **do** ▷ Iterate over each user's interactions
      Choose action-topic $k_d \sim Multi(\phi^K_{r_u})$
      for word $w \in W$ **do**
        Draw $w \sim Multi(\phi^{\mathcal{V}}_{k_d})$
      Draw action $a \sim Multi(\phi^{\mathcal{A}}_{k_d})$
      Draw time-stamp $t \sim Beta(\alpha_{r_u,k_d}, \beta_{r_u,k_d})$
    for each inward exchange (s,u,ℓ) $\in L_u$ **do** ▷ User's inward exchanges
      Draw $\ell \sim Multi(\phi^{\mathcal{L}}_{r_s,r_u})$
    for each outward exchange (u,y,ℓ) $\in L_u$ **do** ▷ User's outward exchanges
      Draw $\ell \sim Multi(\phi^{\mathcal{L}}_{r_u,r_y})$
  return $r_u \forall u \in \mathcal{U}$, $k_d \forall d \in \mathcal{D}_u$

We associate a label $\ell \in \mathcal{L}$ indicating the exchange type (e.g. Question $\rightarrow$ Answer, Comment $\rightarrow$ Answer etc.) between an ordered pair of users (u, v). To capture the knowledge exchange between profile pairs, we set-up $|R|^2$ multinomial distributions over exchange types $\phi^{\mathcal{L}}_{r,r'}$ between all ordered profile pairs (r, r').

Let L_u denote all exchanges for user u with other users v. Notice that sometimes u may initiate the exchange (e.g. ask a question) or respond (e.g. answer). Then, the likelihood $p(L_u \mid r)$ depends on the profiles being served to users involved in exchanges with u. Thus:

$$p(L_u \mid r) \propto \underbrace{\prod_{(s,u,\ell) \in L_u} \phi^{\mathcal{L}}_{r_s,r}(\ell)}_{\text{inbound exchange}} \times \underbrace{\prod_{(u,y,\ell) \in L_u} \phi^{\mathcal{L}}_{r,r_y}(\ell)}_{\text{outbound exchange}}, \tag{3.8}$$

where $\phi^{\mathcal{L}}_{r_s,r}(\ell)$ is the likelihood of an in-bound exchange from source user s served profile r_s, and $\phi^{\mathcal{L}}_{r,r_y}(\ell)$, for an out-bound exchange to user y served r_y.

The overall conditional likelihood $p(u \mid r)$ is the product of likelihood of exchanges $p(L_u \mid r)$ and likelihood of content interactions $p(d \mid r)$ of each user:

$$P(u \mid r) \propto p(L_u \mid r) \times \prod_{d \in D_u} p(d \mid r). \tag{3.9}$$

Algorithm 3.2 summarizes the generative process corresponding to Equation (3.9). We

Table 3.3: Gibbs-Sampler count variables.

Symbol	Description
$n_k^{(w)}, n_k^{(a)}, n_k^{(.)}$	Number of times word w, action a were assigned to topic k, and respective marginals
$n_r^{(k)}, n_r^{(.)}$	Number of times interactions of users served profile r was assigned topic k and marginal
$n_{r,r'}^{(\ell)}, n_{r,r'}^{(.)}$	Number of ℓ-labeled exchanges, all exchanges between users in tables serving r with r'

combine $p(u \mid r)$ from Equation (3.9) with $p(x \mid u)$ (Equation (3.2)) to seat users u on tables x, serving profile r_x.

In this section, we identified the challenges of using only skew-aware partitions [14] when we also have sparse users. Our intuition was to seat users based on their behavioral similarity and not learn profiles at the level of an individual but of a group. We discount common behaviors, encouraging the identification of niche behavior. We introduced action-topics, and each profile is a mixture of these topics. Importantly, each profile learns a different temporal distribution for each topic. Finally, we showed how interactions between profiles guide user seating — that is, users who behave similarly in their interaction with other groups are more likely to be seated together.

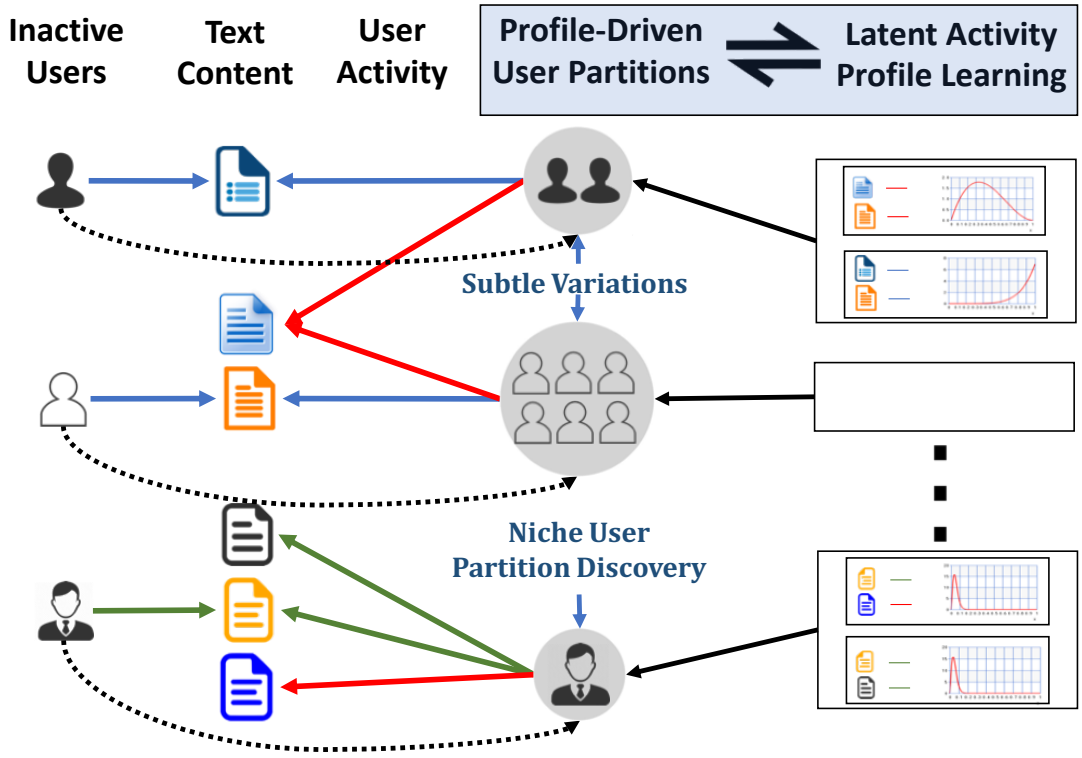
3.5 MODEL INFERENCE

In this section, we describe an efficient Gibbs-sampling approach for model inference, analyze its computational complexity and propose a parallel batch-sampling approach for speed-up. In each iteration of Gibbs-sampling, we unseat users one at a time and re-sample their seating as in Equation (3.2). Profile and Action-topic distributions are simultaneously updated, while hyper-parameters are modified between Gibbs iterations. We factor the seating sampler (Equation (3.13)) for efficiency since the number of tables is not fixed. We speed-up convergence times with coherent initial seating based on similar action distributions and content tags.

The likelihood of generating a user interaction $d = (a, W, t) \in \mathcal{D}_u$ conditional on action-topic $k \in K$ is:

$$p(a, W \mid k) \propto \frac{n_k^{(a)} + \alpha_{\mathcal{A}}}{n_k^{(.)} + |\mathcal{A}|\alpha_{\mathcal{A}}} \times \prod_{w \in W} \frac{n_k^{(w)} + \alpha_{\mathcal{V}}}{n_k^{(.)} + |\mathcal{V}|\alpha_{\mathcal{V}}}. \tag{3.10}$$

Figure 3.2: Our Gibbs-sampler simultaneously samples the seating arrangement of users (eq. 13) and learns profiles to describe the seated users (eq. 9, 10, 11). Users are grouped by behavioral similarity to overcome sparsity.



Thus, the likelihood $p(d \mid r)$ of interaction $d = (a, W, t)$ for a user served activity profile $r \in R$, Equation (3.7) is:

$$p(d \mid r) \propto \sum_{k \in K} \frac{n_r^k + \alpha_K}{n_r^{(.)} + |K|\alpha_K} \times p(a, W, t \mid k, r). \tag{3.11}$$

The likelihood that knowledge exchange occurs between profile pairs (r, r') on type ℓ is:

$$\phi_{r,r'}^{\mathcal{L}}(\ell) = \frac{n_{r,r'}^{\ell} + \alpha_{\mathcal{L}}}{n_{r,r'}^{(.)} + |\mathcal{L}|\alpha_{\mathcal{L}}}. \tag{3.12}$$

Thus, the conditional likelihood in Equation (3.9) can be obtained via Equation (3.11) over $\mathcal{D}_u$ and Equation (3.12) over L_u respectively. We can seat a user u either on an existing table $x \in \{1, \ldots, \chi\}$ serving profile r_x or on a new table $\chi + 1$; Equation (3.2), conditioned on the seating of all other users, denoted by x_{-u}. To avoid likelihood computation over all tables, we perform the draw in two factored steps. We first sample the profile served to u

by marginalizing over tables via Equation (3.5),

$$P(r \mid u, x_{-u}) \sim \left(\frac{N_r - \chi_r \delta}{N + \gamma} + \frac{\gamma + \chi\delta}{|R|(N+\gamma)} \right) p(u \mid r), \tag{3.13}$$

and then sample from the set of tables serving the sampled profile (including the possibility of a new table with this profile draw),

$$P(x \mid r, u, x_{-u}) \sim \begin{cases} \frac{n_x - \delta}{N+\gamma} \times \mathbb{1}(r_x = r), & x \in \{1, \ldots, \chi\}, \\ \frac{\gamma + \chi\delta}{N+\gamma} \times \frac{1}{|R|}, & x = \chi + 1 \end{cases} \tag{3.14}$$

Note that $N = |\mathcal{U}| - 1$, i.e. all users except u. If we draw a new table $\chi + 1$, we assign the sampled profile variable r. We update all counts (Table 3.3) corresponding to prior profile and action-topic assignments for u.

We use well known techniques to update parameters. At the end of each sampling iteration, we update Multinomial-Dirichlet priors $\alpha_{\mathcal{V}}$, $\alpha_{\mathcal{A}}$, α_K and $\alpha_{\mathcal{L}}$ by Fixed point iteration [141]. We update Beta parameters $(\alpha_{r,k}, \beta_{r,k})$ by the method of moments [217]. We round all time-stamps to double-digit precision and we cache probability values $p(t \mid r, k) \; \forall \; t \in [0, 1], r \in R, k \in K$ at the end of each sampling iteration, thus avoiding $R \times K$ scaling for $p(u \mid r)$ in Equation (3.13). While we fix the Pitman-Yor parameters in our experiments for simplicity, if needed, we can estimate them via auxiliary variable sampling [180, 199].

Computational Complexity

Our inference is linear in the number of users $|\mathcal{U}|$ and interactions $|\mathcal{D}|$, scaled by $R + K$ (see empirical results in Figure 3.6). To see this, notice that in each Gibbs iteration, computing Equations (3.10) and (3.11) involves $|\mathcal{D}| \times (K + R)$ computations. Equation (3.13) requires an additional $|\mathcal{U}| \times R$ computations. We prevent $R \times K$ scaling for $p(u \mid r)$ in Equation (3.13) by caching. We cache the first product term of Equation (3.13) for each $r \in R$, and update it only when there is a change in the seating arrangements on tables serving profile r.

Parallelization with Batch Sampling

We scale to massive datasets by parallelizing inference via batch sampling. The Gibbs sampler described above samples each user's seating $P(x_u \mid u, x_{-u})$ in Equation (3.14), conditioned on all other users. This necessitates iteration over $\mathcal{U}$. Instead, seating arrange-

Table 3.4: User Action Description (Coursera/Stack-Exchange).

Platform	Action	Description
Coursera MOOC	Play	First lecture segment view
	Rewatch	Repeat lecture segment view
	Clear Concept	Back and forth movement, pauses
	Skip	Unwatched lecture segment
	Create Thread	Create forum thread for inquiries
	Post	Reply to existing threads
	Comment	Comment on existing posts
Stack-Exchange	Question	Posting a question
	Answer	Authoring answer to a question
	Comment	Comment on a question/answer
	Edit	Modify posted content
	Follow	Following posted content

ments could be simultaneously sampled in batches $U \subset \mathcal{U}$ conditioned on all users outside the batch, i.e., $P(x_U \mid U, x_{\mathcal{U}-U})$ where x_U denotes the table assignments to users in batch U. For efficiency, batches must be chosen with comparable computation. We approximate computation for $u \in \mathcal{U} \propto |\mathcal{D}_u| + |\mathcal{L}_u|$ to decide apriori batch splits for sampling iterations. Note that when batch sampling, we can only exploit knowledge exchange links between users in the batch and users *not* in the batch. In practice, since $|U| \ll |\mathcal{U}|$, the impact of not using knowledge exchanges between users in the same batch turns out to be negligible.

3.6 DATASET DESCRIPTION

We now provide a brief description of the Coursera MOOC and Stack-Exchange datasets that we use in our experiments and characterize them in terms of the extent of skew and sparsity exhibited across each dataset.

Stack-Exchanges are community Q&A websites where participants discuss a wide range of topics primarily via user-authored questions, answers, and comments. Users interact with each other and perform a range of actions (e.g., post question, answer, comment, etc.). We experiment on 10 Stack-Exchanges, chosen for thematic diversity and size variation.

On the other hand, Coursera MOOCs feature video lectures for students to watch and a forum where students and instructors can interact. We analyze the actions (e.g., play, skip, rewind, etc.) on the videos, lecture content via subtitles, and the forum interaction for four MOOCs chosen for thematic diversity. The user action types and datasets are summarized

Table 3.5: Preliminary analysis indicates significant behavior skew and inactive user proportion, although slightly reduced in specialized domains, e.g., Christianity.

Platform	Dataset	Users	Interactions	η_t	S_N
Coursera MOOC	Comp Sci-1	26,542	834,439	-2.51	0.67
	Math	10,796	162,810	-2.90	0.69
	Nature	6,940	197,367	-2.43	0.70
	Comp Sci-2	10,796	165,830	-2.14	0.73
Stack-Exchange	Ask-Ubuntu	220,365	2,075,611	-2.81	0.65
	Android	28,749	182,284	-2.32	0.56
	Travel	20,961	277,823	-2.01	0.66
	Movies	14,965	150,195	-2.17	0.67
	Chemistry	13,052	175,519	-2.05	0.63
	Biology	10,031	138,850	-2.03	0.71
	Workplace	19,820	275,162	-2.05	0.59
	Christianity	6,417	130,822	-1.71	0.64
	Comp Sci	16,954	183,260	-2.26	0.62
	Money	16,688	179,581	-1.72	0.63

in Table 3.4 and Table 3.5 respectively. We chose these two diverse application platforms to exhibit the generalizability of our proposed approach.

To get a feel for these datasets, let us examine sparsity and behavior skew. To understand sparsity, we compute the power-law ($f_w = c.w^{\eta_t}$) exponent η_t that best describes the fraction of users f_w who were active for w-weeks. A more negative index indicates that fewer users are consistently active. As a reference point, when $\eta_t = 0$, a constant fraction of users are always active. Thus when we notice that $\eta_t = -2.81$ for `Ask Ubuntu` Stack Exchange in Table 3.5, it means that the number of users who are active for two weeks is just 14% of those active for one week. Table 3.5 indicates that larger Stack Exchanges tend to have greater sparsity.

We measure skew by first identifying each user's dominant action type or style (e.g., commenter, editor) and then compute the normalized entropy S_N of the resulting user distribution.

In a large Stack-Exchange such as `Ask-Ubuntu`, while less than 5% (c.f Figure 3.1) of the users have 'Answer' as their dominant type, over 60% of the users have 'Comment' as their dominant action. This does not consider content topics, which results in greater skew. When $S_N = 1$, all dominant action types are equally likely; in contrast, $S_N = 0$ indicates a single dominant action type. In MOOCs, 'Play' is the dominant action type with low forum participation (participation rates ∼10-15% in our MOOC forums).

3.7 EXPERIMENTAL RESULTS

In this section, we present extensive quantitative and qualitative analyses of our model. We begin by introducing baseline methods (Section 3.7.1), followed by prediction tasks undertaken (Section 3.7.2), and present impressive quantitative results for CMAP in Section 3.7.3. Then in Section 3.7.4, we qualitatively analyze the reasons for CMAP's gains over baselines. In Section 3.7.6 we examine a counterfactual: *what if the data had less skew?* Finally, we analyze scalability (Section 3.7.7), parameter sensitivity (Section 3.7.8) and discuss limitations in Section 3.7.9.

3.7.1 Baseline Methods

We compare our model (CMAP) with user representations from three state-of-the-art models and two standard baselines. We list the baselines below.

LadFG [158]: LadFG is a deep recurrent approach to learn behavior representations from temporal activity and demographic information of users. We provide LadFG action-content data from interactions and all available user demographic information.

BLDA [159]: BLDA is an LDA-based extension to capture latent associations of user actions and content. It represents users as a mixture of these content-action topics.

FEMA [77]: FEMA is a multifaceted sparsity-aware tensor factorization approach employing external regularizers for smoothing. Facets in our datasets are users, words, and actions. We set user and word regularizers to their exchanges and co-occurrence count, respectively. We could not run FEMA on Ask-Ubuntu and Comp Sci-1 datasets due to very high memory and compute requirements (Regularizer matrices in FEMA scale quadratically $O(|\mathcal{U}|^2)$).

DMM (Only text) [236]: We apply DMM to the textual content of all interactions to learn topics. We represent users by the proportions of topics in their interaction content.

Logistic Regression Classifier (LRC) [103]: Logistic regression based classification model. Input features are DMM topics that the user interacts with and actions in each topic (Answer, Edit etc.).

We construct user representations for models as follows: For CMAP (Ours), we use the $|R|$-dimensional normalized conditionals $P(r \mid u)$ for each user as given by Equation (3.5); For BLDA, we use normalized conditionals over the set of behaviors for each user as computed by the authors [159]; for FEMA, we use respective rows of user projection matrix, $\mathbf{A}_t$ [77]; for LadFG, we use latent user embeddings learned upon training; for DMM, we use topic proportions for user generated text. We use LRC only for prediction tasks, as

it does not build user representations.

For fair comparison, the user representations from baselines were the same dimensionality as the profile count $|R|$ for our model. We use $|R| = 20$ and 40 Action-Topics ($|K|$) for all datasets. We initialize Dirichlet priors as: $\phi_k^{\mathcal{V}}, \phi_k^{\mathcal{A}}, \phi_r^K$ and $\phi_{r,r'}^{\mathcal{L}}$ with the common strategy [35, 94, 235] ($\alpha_X = 50/|X|, X = \{\mathcal{A}, \mathcal{L}, K\}$, and $\alpha_{\mathcal{V}} = 0.01$) and *Beta* parameters α_{rk}, β_{rk} to 1. CRP parameter initialization $\delta = 0.5, \gamma = 1$ performed well consistently. Our experiments were performed on a single x64 XSEDE compute node[3] [200] (Intel Xeon E5-2680v3, 64 GB Memory). Our implementations are available online[4].

3.7.2 Prediction Tasks

We identify three distinct task types for evaluating the quality of user representations across methods. We focus on two User Characterization tasks, a Future Activity Prediction task, and Question Recommendation in Stack-Exchanges. Below, we list the tasks.

User Characterization (MOOC) - Certificate Earner: Coursera awards certifications to students maintaining high cumulative grades over assignments. We predict students obtaining certificates with the user representations obtained from each model.

User Characterization (Stack-Exchange) - Reputed User: For Stack-Exchanges, we predict if participants have a high reputation with user representations from each model. We define users in a Stack-Exchange to have a high reputation if they lie in the top quartile (25%) of all reputation scores.

Question Recommendation (Stack-Exchange): For popular questions in Stack-Exchanges, we identify suitable users to answer them. In each dataset, we choose a set of 100 held-out popular questions & learn user representations by applying models to their remaining activity. We then perform 5-fold Cross-Validation for each held-out question with the known users who answered the question and an equal number of negative users chosen at random.

Future Activity Prediction (All Datasets): We obtain topic assignments for user interactions with DMM [236] ($T = 20$). For each user, we predict their future activity mixture over topics & actions given user representations with their past activity from each model (6-month data held-out). LRC is not used in Future Activity Prediction as it does not build a user representation.

We use standard classifiers and evaluation metrics. Characterization and Recommendation use linear-kernel SVM evaluated with Precision, Recall, F1-Score, and Area-Under-Curve

[3] `https://www.xsede.org/`

[4] https://github.com/ash-shar/CMAP

Table 3.6: Reputed User Prediction ($\mu \pm \sigma$ across Stack-Exchanges). We obtain improvements of 6.65-21.43% AUC.

Method	Precision	Recall	F1-score	AUC
LRC	0.73 ± 0.04	0.69 ± 0.04	0.72 ± 0.03	0.73 ± 0.03
DMM	0.69 ± 0.05	0.65 ± 0.04	0.66 ± 0.04	0.70 ± 0.04
LadFG	0.86 ± 0.03	0.75 ± 0.03	0.79 ± 0.02	0.80 ± 0.03
FEMA	0.79 ± 0.04	0.73 ± 0.03	0.77± 0.03	0.79 ± 0.04
BLDA	0.75 ± 0.04	0.71 ± 0.04	0.74 ± 0.03	0.74 ± 0.04
CMAP	0.85 ± 0.02	**0.83 ± 0.03**	**0.84 ± 0.02**	**0.86 ± 0.02**

(AUC). Future Activity Prediction uses Linear Regression. Both were implemented with default parameters in sklearn [5]. For the activity prediction task, we measure the Root Mean Squared Error (RMSE) in predicted activity proportions for *(topic, action)* pairs against actual proportions of users in the held-out future activity. We compute results with 5-fold cross-validation for each dataset. Statistically significant gains (Paired t-test, $p < 0.05$) are bold-faced.

3.7.3 Results

We examine the experimental results for each of the three tasks—User characterization, question recommendation, and future activity prediction in this section.

Our method improves on the baselines in the *reputation prediction task* by 6.26-15.97% AUC averaged across the Stack-Exchanges; Table 3.6 shows the results with statistically significant improvements in bold. LadFG performs slightly better on the overall precision in reputation prediction (not statistically significant), likely due to over-fitting the embeddings to user-level data resulting in a low recall. Our ability to discover more distinct user clusters even with the same latent dimensions as baselines (refer fig. 3.4) is the main reason for our gains in predicting reputation. Similarly, we improve on *certification prediction* (see Table 3.7) by 6.65-21.43% AUC averaged over MOOCs.

For the *question recommendation task* Table 3.8, we see gains between 6-47% AUC over the baselines. To do well in this task, we require the model to make finer distinctions between the topical preferences of users; user reputation and action style are also important in this task.

For the *future activity prediction task*, our method shows gains over baselines in RMSE by 12%-25% on MOOCs and between 9.5%-22% on Stack-Exchanges; (see Table 3.9). Gains

[5] http://scikit-learn.org/

Table 3.7: Certificate Earner Prediction ($\mu \pm \sigma$ across MOOCs); CMAP improves upon baselines by 6.65-21.43% AUC.

Method	Precision	Recall	F1-score	AUC
LRC	0.76 ± 0.04	0.71 ± 0.05	0.74 ± 0.04	0.72 ± 0.03
DMM	0.77 ± 0.03	0.74 ± 0.04	0.75 ± 0.03	0.74 ± 0.03
LadFG	0.81 ± 0.02	0.78 ± 0.02	0.79 ± 0.02	0.79 ± 0.02
FEMA	0.78 ± 0.03	0.75 ± 0.04	0.76 ± 0.03	0.78 ± 0.03
BLDA	0.80 ± 0.04	0.75 ± 0.03	0.77 ± 0.03	0.77 ± 0.04
CMAP	**0.86 ± 0.02**	**0.81 ± 0.03**	**0.83 ± 0.02**	**0.84 ± 0.02**

Table 3.8: Question Recommendation ($\mu \pm \sigma$ across Stack-Exchanges)with 6.30-47.45% AUC gains for CMAP. DMM performs quite well owing to importance of content in this task.

Method	Precision	Recall	F1-score	AUC
LRC	0.65 ± 0.06	0.57 ± 0.08	0.60 ± 0.06	0.57 ± 0.05
DMM	0.72 ± 0.04	0.81 ± 0.05	0.75 ± 0.04	0.74 ± 0.04
LadFG	0.88 ± 0.03	0.60 ± 0.02	0.71 ± 0.02	0.76 ± 0.04
FEMA	0.79 ± 0.05	0.73 ± 0.06	0.77± 0.05	0.79 ± 0.03
BLDA	0.70 ± 0.04	**0.84 ± 0.04**	0.77 ± 0.03	0.75 ± 0.04
CMAP	**0.89 ± 0.03**	0.81 ± 0.02	**0.85 ± 0.03**	**0.84 ± 0.02**

are explained by our model's ability to make a finer distinction on action styles and better distinctions between profiles assigned to users.

In this section, we showed impressive performance gains on three types of tasks for our model over state-of-the-art baselines. In the next section, we qualitatively analyze the reasons for its success.

3.7.4 Why does CMAP Work Well?

To interpret the gains obtained by CMAP, we examine the extracted clusters in Section 3.7.4 and then look at users responsible for the performance gains of our model in Section 3.7.5.

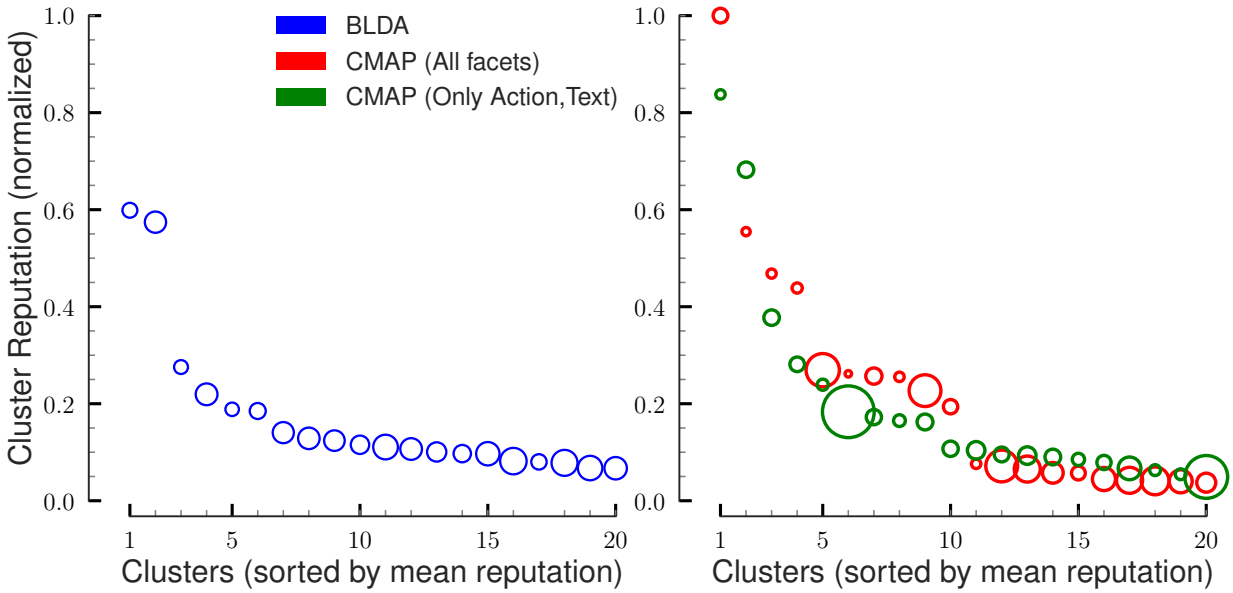
The Impact of Profile Driven Seating

We now compare clusters obtained through CMAP seating against conventional generative assignments in BLDA [159] on Stack-Exchanges. Both models group users best described

Table 3.9: Future Activity Prediction (RMSE ($\times 10^{-2}$) $\mu \pm \sigma$), Lower RMSE is better. CMAP ouperforms baselines in MOOCs (12%-25%) and Stack-Exchanges (9.5%-22%).

Method	DMM	LadFG	FEMA	BLDA	CMAP
MOOC	4.9 ± 0.4	4.2 ±0.3	4.1 ± 0.2	4.4 ± 0.4	**3.6 ± 0.2**
Stack-Ex	8.6 ± 0.6	7.9 ± 0.4	7.5 ± 0.3	7.4 ± 0.5	**6.7 ± 0.4**

Figure 3.3: Bubbles denote user clusters discovered by each model in the Ask-Ubuntu dataset (Bubble size $\propto$ Users in Cluster). CMAP discovers fine distinctions of reputed users (Profiles 1,2,3,4) by content preference and activity (Table 3.10). BLDA clusters are mean-sized and close to the population average in reputation. In contrast, our assignments better reflect the behavior skew of users in the dataset.



by the same profile to form clusters. We use the average user reputations of the clusters (appropriately normalized) as an external validation metric for cluster quality. We also run our model excluding time and knowledge-exchanges to see the effect on the clusters. Figure 3.3 shows the result from the `Ask-Ubuntu` Stack Exchange, and Table 3.10 shows the main activities and topics of the top three CMAP clusters.

We make the following key observations from the clusters:

The mean-shift problem: The Dirichlet-Multinomial setting in BLDA tends to merge profiles and hence shift cluster sizes and average participant reputation closer to the mean. Figure 3.3 shows that 15 of 20 BLDA clusters have nearly the same size and average reputation. Both variants of CMAP show diversity in cluster size and high reputation variability across

Table 3.10: Action and Content description of users in the top-3 clusters discovered by CMAP in Ask-Ubuntu, +/- values of action proportions against the average Ask-Ubuntu user.

Cluster	Action Style	Common Topics
1	+31% Answer, +24% Edits, -09% Questions	Graphics Drivers, Booting Issues, Disk Partitions
2	+67%Answer, -03% Edits, -21% Questions	Gnome, Desktop, Package Install
3	+11% Answer, -04% Edits, +47% Questions	Script, Application, Sudo Access

tables. Our cluster assignments appear to mirror the behavior skew for `Ask-Ubuntu` (c.f. Figure 3.1).

Profile quality: CMAP learns finer variation in the topic affinities and actions of expert users. We can observe these variations from Figure 3.3 and from Table 3.10. The top three profiles are of higher reputation, smaller in size, and from Table 3.10, each of these clusters shows distinct activities different from the mean activity. CMAP clusters appear to better reflect skewed user activity (c.f. Table 3.5) and content preference (c.f. Figure 3.1) with flexible profile-driven seating.

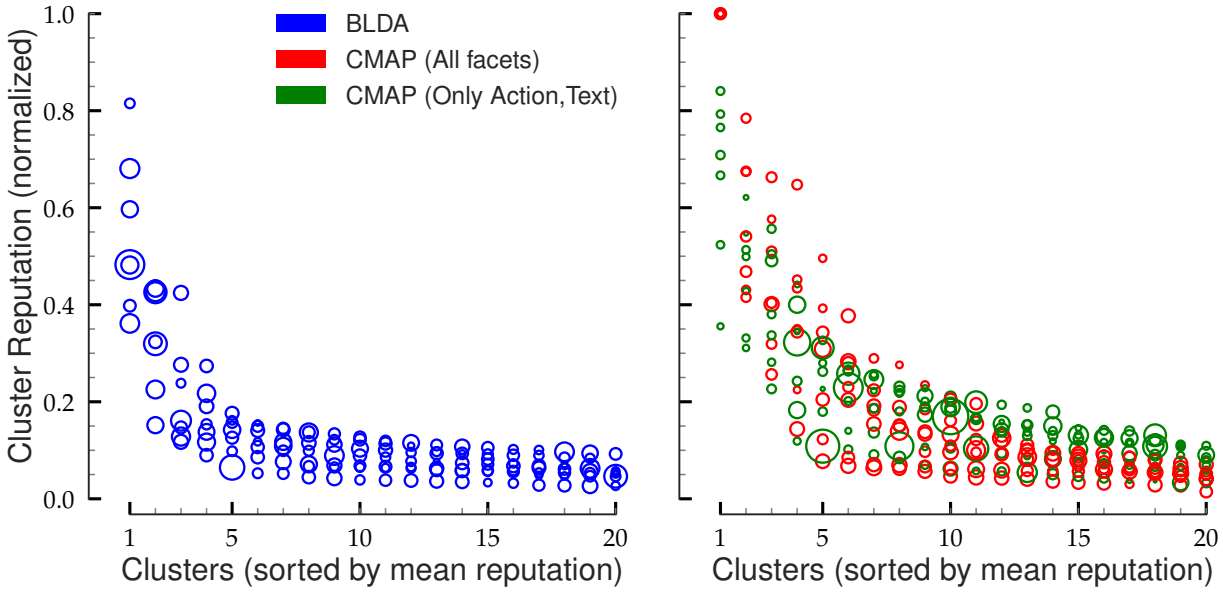
We observe a similar trend in the aggregated clusters obtained from all the other Stack-Exchange datasets (c.f. Figure 3.4). The Dirichlet-Multinomial setting in BLDA results in similarly sized clusters which cannot model highly skewed content and action affinities of users. Note the fewer high-reputation clusters of BLDA in comparison to the finer distinctions of reputed users in our model. Our performance in prediction and recommendation reflect these observations; we see significant gains in our ability to characterize reputed users and recommend suitable content (Section 3.7.2).

3.7.5 Making Gains on Inactive Users

We now investigate the source of our gains. We split users in each Stack-Exchange and MOOC into four quartiles based on interaction count (Quartile-1 is the least active). Then, we evaluate each method on Reputation and Certificate Prediction AUC in each quartile of Stack Exchange and MOOC datasets, respectively.

Our model shows large gains (Figure 3.5) in Quartiles 1,2 that contain sparse users. We attribute these gains due to our joint profile learning to describe similar users seated on

Figure 3.4: Bubbles denote clusters in other Stack-Exchanges (Bubble size $\propto$ Users in Cluster). CMAP discovers highest reputation clusters in all datasets (Thick red dot, Top-left). BLDA clusters tend to mean reputation, size (Mean-Shift) not capturing disparities. In our case, profiles 1,2,3,4 appear to capture niche, highly reputed user behaviors.



tables. The decision to address skew and sparsity jointly has two advantages: better profile fits for sparse users; more distinct and informative profiles in skewed scenarios. In contrast, models building representations at the user level perform weakly in Quartiles-1,2 since these methods rely on interaction volume. We make smaller gains in Quartiles 3,4.

To summarize: jointly addressing sparsity and skew by profile-driven seating is responsible for our gains. Importantly, the clusters are coherent; the model learns fine distinctions in behavioral profiles and exhibits behavior skew found in the underlying data.

3.7.6 What if there was Less Skew?

In this section, we study a counterfactual: what if the real-world datasets were less skewed?

To study this question, we sub-sample users who predominantly perform the two most common actions in our largest datasets, `Ask-Ubuntu` (Comments and Questions) and `Comp Sci-1 MOOC` (Play and Skip). These users are sub-sampled by half while retaining all other users, reducing overall action skew in the data. Baseline models are expected to perform better with reduced skew. All models degrade in Ask-Ubuntu owing to significant loss of content.

Figure 3.5: Effects of activity sparsity on prediction tasks (AUC) for Stack Exchanges (datasets 1-10) and MOOCs (datasets 11-14). CMAP has the greatest performance gains in Quartile-1 (Sparse), the performance gap reduces for active users (Quartile-4).

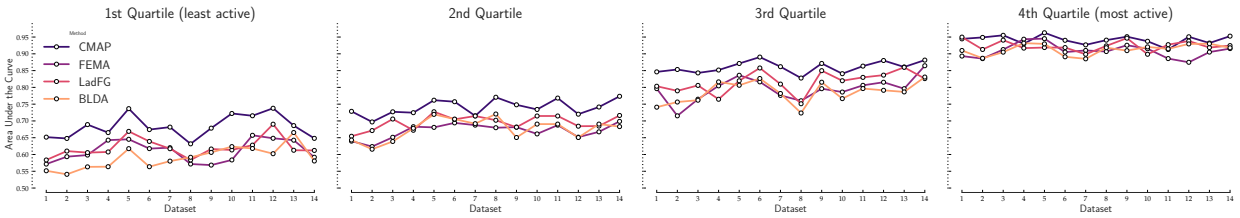


Table 3.11: CMAP outperforms baselines (AUC) in the de-skewed datasets, but with smaller gains.

Method	**Ask-Ubuntu**		**CompSci1 MOOC**	
	Original	Deskewed	Original	Deskewed
LRC	0.671	0.656	0.713	0.734
DMM	0.647	0.611	0.684	0.672
LadFG	0.734	0.718	0.806	0.830
BLDA	0.706	0.683	0.739	0.788
CMAP	**0.823**	**0.746**	**0.851**	0.849

Table 11 shows that CMAP still maintains a lead owing to inactive users. We also investigate performance gains in a highly skewed and sparse Stack-Exchange (Ask-Ubuntu) vs least skewed (Christianity) in Table 12. On average, we outperform baselines by 13.3% AUC for Ask-Ubuntu vs 10.1% for Christianity Stack-Exchange in User Characterization.

3.7.7 Scalability Analysis

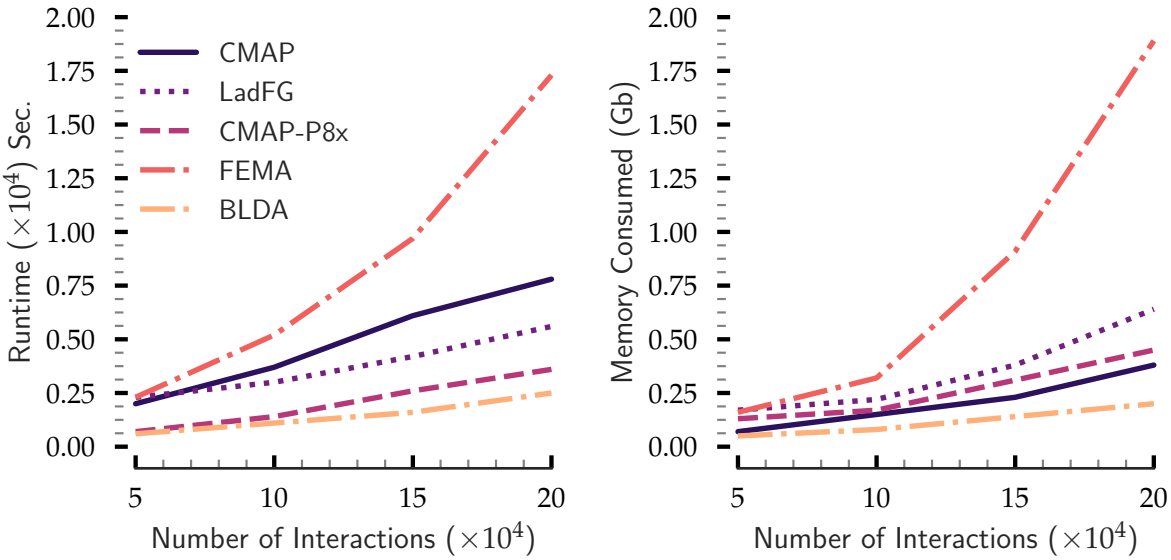
We compared the runtimes and memory consumption of our serial and batch-sampling (with 8 cores) inference algorithms with other models for different volumes of interaction data obtained from random samples of the Ask-Ubuntu Stack-Exchange.

Model Analysis: BLDA is the fastest among the set of compared models owing to its simplistic profiling model. Our 8x batch sampler (with the significantly more complex generative model) is comparable to BLDA in runtime. The FEMA tensor approach was the least scalable (in memory consumption and runtime) owing to the $O(|\mathcal{U}|^2)$ growth of the User-User regularizer matrix. Figure 3.6 exhibits the comparisons across the methods against the total count of user interactions in the dataset. We measure the absolute runtime values and plot the curves to observe scaling effects.

Table 3.12: We see greater gains for User Characterization in a high-skew dataset (Ask-Ubuntu) vs low-skew (Christianity).

Method	DMM	LRC	LadFG	FEMA	CMAP	BLDA
Ask-Ubuntu	0.647	0.671	0.734	-	**0.823**	0.706
Christianity	0.684	0.720	0.842	0.818	**0.856**	0.791

Figure 3.6: Effects of dataset size on algorithm runtime and memory consumption. BLDA is the fastest among the set of compared models.



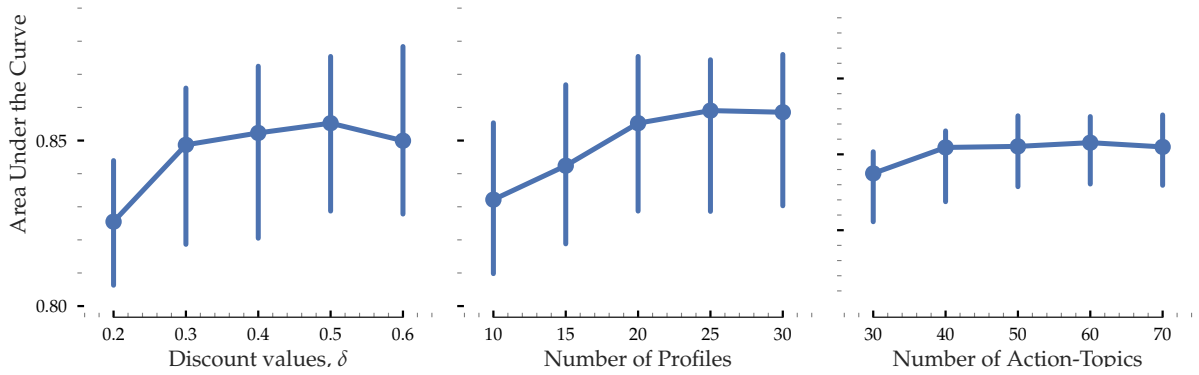
3.7.8 Parameter Sensitivity Analysis

Our model is primarily impacted by three core parameter values: the maximum number of behavior profiles R, the number of action-topics K, and the Pitman-Yor discount parameter δ, which controls the extent of exploration when new profiles are assigned to users.

We find aggregate results to exhibit stability in a broad range of parameter values. This indicates that our model requires minimal parameter tuning in practice (Figure 3.7). It is worth noting that while R primarily impacts the granularity of the discovered activity profiles, K impacts the resolution of content-action associations. Dirichlet and other hyper-parameters have negligible impact on the profiles and seating arrangement learned by our model.

Our inference algorithm converges within 1% AUC in less than 400 sampling iterations across all datasets. As previously described, the total computational expense is proportional

Figure 3.7: Mean performance(AUC) & 95% confidence interval with varying model parameters one at a time: δ, R, K. Stability is observed in broad ranges of parameter values.



to both, the number of sampling iterations, and the total number of user interactions.

3.7.9 Limitations

We identify two limitations. First, we make no assumptions about the structure of knowledge (e.g., knowledge of "probability" is helpful to understand "statistical models"); incorporating knowledge structure, perhaps in the form of an appropriate prior will help with better understanding participants with low activity. Second, we assume a bounded time range. The development of latent profiles for streaming activity can lead to deployment with real-time data.

It would be an interesting exercise to observe the effect of dynamic updates to Pitman-Yor hyper-parameters over sampling iterations [199]. Although such an approach has been explored for LDA [209], it is unclear how over-fitting in our approach can be avoided in the case of hyper-parameter drift. We plan this study for future work.

3.8 CONCLUSION AND NEXT STEPS

3.8.1 Chapter Summary

This chapter proposed a coupled clustering and profile fitting approach to jointly mitigate user behavior skew and sparsity and learn descriptive statistical representations of user behavior. Unlike prior methods that provide limited solutions to aggregate data skew or sparse data, our framework jointly addresses skew and sparsity across graphical behavioral models of individual user behavior, independent of the model's specifics or the modeled data

modalities. Our primary technical contribution is to partition users and learn behavioral profiles corresponding to each partition with a non-parametric Pitman-Yor process governing partitions' formation. Our approach deeply couples the user-group assignments and group-profile learning process. It incentivizes exploration to prevent saturation or convergence to degenerate solutions (e.g., all users assigned to a single aggregate behavior profile).

We can flexibly choose the data modalities and interaction types modeled by the user profile model depending on the platform and downstream task requirements. Extensive experiments over large online forums validate our behavior profiles' informativeness across diverse recommendation and profiling tasks. Qualitative analysis indicates our ability to discover niche and informative user groups that strongly reflect the actual empirical reputation/experience distribution on the Stack-Exchange platform. On the whole, we show strong inference and recommendation gains for sparse participants. Furthermore, our algorithms scale linearly and do not require supervision or auxiliary data.

3.8.2 Improvements to the Proposed Framework

We identify a few rewarding future directions to enhance the applications of our model. The streaming recommendation problem is typically handled in a session-segmented manner, and user behavior can significantly change across sessions depending on their specific intents. A straightforward way to extend the current framework to such a scenario is to integrate a session/intent-based probabilistic graphical model with our grouping mechanism. The graphical model would then enable a choice of intents for each user visit. The grouping process would leverage the meta-distributions over a fixed set of user intents to group users with similar distributions [49]. An alternate approach is to develop an incremental model for streaming data, where users are permitted to evolve group memberships temporally.

Incorporating knowledge priors on expected behavior patterns (e.g., how students lacking a strong mathematical background might review video content in an advanced MOOC) in the context of the MOOC platform [7] can enable constrained group formation and speed-up model convergence.

3.8.3 Addressing the Limitations of a User-Focused Approach

While the proposed profiling model is effective with skewed and sparse user data, it relies on the item inventory or item attributes to learn the user profiles. Since user behavior profiles are essentially mixtures or factored distributions over item-set views, their informativeness depends on the underlying item-set. The behavior profiles are reliant on item descriptors

such as textual content and user-item interaction types in other scenarios. However, on numerous recommendation platforms, the items may not offer extensive documentation or interaction histories, especially in the infinite inventory setting [50].

The inventory or item listings may grow rapidly, resulting in sparse long-tail items offering very little interaction data. The user profiles learned by the current skew-aware grouping process describe the constituent users' item preferences. However, the presence of sparse items complicates the inference task, specifically with discrete non-decomposable items (such as long-tail items on *e-commerce* platforms) that lack associated feature data to connect them to the rest of the item inventory.

In such cases, user-grouping in isolation is insufficient due to skew and sparsity on the item side. Specifically, how do we represent long-tail items or learn feature representations sufficiently descriptive to perform effective user recommendations? The availability of descriptive item representations would permit us to effectively apply user grouping techniques like the one developed in this chapter. In the next chapter, we answer this question in a platform-agnostic data-driven manner by leveraging the distinguishing aspects of items that users do not provide, namely the explicit co-occurrences or basket information [167] on the recommendation platform. Note the synergistic advantages of such an approach; the resulting feature representations of items benefit the identification of suitable item choices for users and address the supply side fairness or performance concerns by matching sparse or under-reviewed items to the right users [1].

In the next chapter, we model the implicit association structure of items in the feedback data while simultaneously training recommender models in an architecture-agnostic manner, resulting in enhanced long-tail performance.

CHAPTER 4: REPRESENTING SPARSE ITEMS VIA SELF-SUPERVISED ASSOCIATION LEARNING

In recent times, deep neural networks have found success in Collaborative Filtering (CF) based recommendation tasks. By parametrizing the latent factor interactions of users and items with neural architectures, they achieve significant scalability and performance gains over matrix factorization. However, the long-tail phenomenon in recommender performance persists on online media or retail platforms' massive inventories. Given the diversity of neural architectures and applications, there is a need to develop a generalizable and principled strategy to enhance long-tail item coverage.

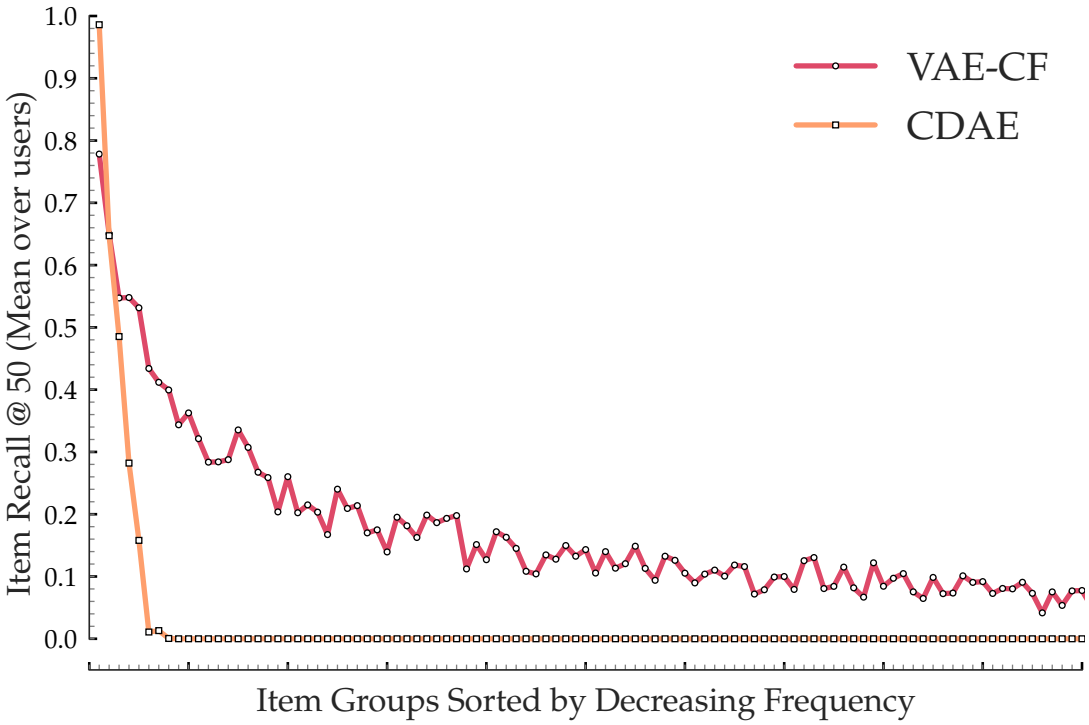
This chapter proposes a novel adversarial training strategy to enhance long-tail recommendations for users with Neural CF (NCF) models. The adversary network learns the implicit association structure of entities in the feedback data. Simultaneously, we train the NCF model to reproduce these associations and avoid the adversarial penalty, resulting in enhanced long-tail performance. Experimental results show that even without auxiliary data, adversarial training can boost long-tail recall of state-of-the-art NCF models by up to 25%, without trading-off overall performance. We evaluate our approach on two diverse platforms, content tag recommendation in Q&A forums and movie recommendation.

4.1 INTRODUCTION

Recommender systems play a pivotal role in sustaining massive product inventories on online media and retail platforms and reduce information overload on users. Collaborative filtering methods personalize item recommendations based on historical interaction data (implicit feedback setting), with matrix-factorization being the most popular approach [92]. In recent times, NCF methods [60, 115, 224] have transformed simplistic inner-product representations with non-linear interactions, parametrized by deep neural networks. Although performance gains over conventional approaches are significant, a closer analysis indicates skew towards popular items (Figure 4.3) with ample evidence in the feedback (overfit to popular items), resulting in poor niche (long-tail) item recommendations to users (see fig. 4.1). This stifles user experience and reduces platform revenue from niche products with high-profit margins.

Conventional effort to challenge the long-tail in recommendation has been two-fold [234]. First, integration with neighbor-based models [130] to capture inter-item, inter-user and cross

Figure 4.1: CDAE[224] and VAE-CF[115] recall for item-groups (decreasing frequency) in MovieLens (*ml-20m*). CDAE overfits to popular item-groups, falls very rapidly. VAE-CF has better long-tail recall due to representational stochasticity.



associations in the latent representations and second, incorporating auxiliary data (e.g. item descriptions) to overcome limited feedback [211] or hybrid methods [94, 155]. While neural models readily adapt auxiliary data [111], the association/neighbor-based path is relatively unexplored due to the heterogeneity of representations and architectures.

Given the diversity of NCF architectures and applications [60, 111, 115], architectural solutions may not generalize well. Instead, we propose to augment NCF training to levy penalties when the recommender fails to identify suitable niche items for users, given their history and global item co-occurrence. To achieve this, conventional neighbor models employ static pre-computed links between entities [130] to regularize the learned representations. While it is possible to add a similar term to the NCF objective, we aim to learn the association structure rather than imposing it on the model. Towards this goal, we introduce an adversary network to infer the inter-item association structures, unlike link-based models, guided by item co-occurrences in the feedback data. The adversary network is trained in tandem with the recommender. It can readily integrate auxiliary data and be extended to model inter-user or cross associations.

For each user, a penalty is imposed on the recommender if the suggested niche items do

not correlate with the user's history. The adversary is trained to distinguish the recommender's niche item suggestions against actual item pairings sampled from the data. The more confident this distinction, the higher the penalty imposed. As training proceeds, the adversary learns the inter-item association structure guided by the item pairs sampled from user records while the recommender incorporates these associations until mutual convergence. In summary, we make the following contributions:

- Unlike conventional neighbor models, our adversary model learns the association structure of entities rather than imposing pre-defined links on the recommender model.
- Our approach is architecture and application agnostic.
- Experimental results on two diverse platforms show substantial gains (by up to 25%) in long-tail item recall for state-of-the-art NCF models while not degrading overall results.

We now present our problem formulation, model details (sec. 4.2, 4.3) experimental results (sec. 4.4), and conclude in sec. 4.5.

4.2 PROBLEM DEFINITION

We consider the implicit feedback setting with binary interaction matrix $\mathcal{X} \in \mathbb{Z}_2^{M_\mathcal{U} \times M_\mathcal{I}}, \mathbb{Z}_2 = \{0, 1\}$ given users $\mathcal{U} = \{u_1, \ldots, u_{M_\mathcal{U}}\}$, items $\mathcal{I} = \{i_1, \ldots, i_{M_\mathcal{I}}\}$. Items $\mathcal{I}$ are partitioned apriori into two disjoint sets, $\mathcal{I} = \mathcal{I}^\mathcal{P}$ (*popular items*) $\cup$ $\mathcal{I}^\mathcal{N}$ (*niche/long-tail items*) based on their frequency in $\mathcal{X}$. We use the notation $\mathcal{X}_u$ to denote the set of items interacted by $u \in \mathcal{U}$, further split into popular and niche subsets $\mathcal{X}_u^\mathcal{P}$, $\mathcal{X}_u^\mathcal{N}$ respectively.

The base neural recommender model $\mathbf{G}$ learns a scoring function $f_\mathbf{G}(i \mid u, \mathcal{X}), i \in \mathcal{I}, u \in \mathcal{U}$ to rank items given u's history $\mathcal{X}_u$ and global feedback $\mathcal{X}$, by minimizing CF objective function $\mathcal{O}_\mathbf{G}$ over recommender $\mathbf{G}$'s parameters θ via stochastic gradient methods. Typically, $\mathcal{O}_\mathbf{G}$ is composed of a reconstruction loss (analogous to conventional inner product loss [92]) and a suitable regularizer depending on the architecture. We adopt $\mathbf{O}_\mathbf{G}$ as a starting point in our training process. Our goal is to enhance the long-tail performance of recommender $\mathbf{G}$ with emphasis on the niche items $\mathcal{I}^\mathcal{N}$.

4.3 MODEL

Most NCF models struggle to recommend niche items with limited click histories, owing to the reconstruction-based objective's implicit bias. Conventional neighbor models [130]

apply simplistic pre-defined associations such as Pearson correlation first, and then learn the social representations for recommendation. In contrast, our critical insight is that these two tasks are mutually dependent, namely generating item recommendations for user u, and modeling the associations of recommended niche items to his history $\mathcal{X}_u$. The adversarial network paradigm [53] fits our application well; we seek to balance the tradeoff between the popular item biased reconstruction objective against the recall and accuracy of long-tail item recommendations.

Towards the above objective, we introduce the adversary model $\mathbf{D}$ in our learning framework to learn the inter-item association structure in the feedback data and correlate $\mathbf{G}$'s niche item recommendations with popular items in the user's history, $\mathcal{X}_u^{\mathcal{P}}$. We associate $\mathbf{G}$'s niche item recommendations with u's popular item history since niche-popular pairings are the most informative (inter-popular pairs are redundant, inter-niche pairs are noisy). The adversary $\mathbf{D}$ is trained to distinguish "fake" or synthetic pairings of popular and niche items sampled from $X_u^{\mathcal{P}}$ and $f_{\mathbf{G}}(i \mid u, \mathcal{X})$ respectively, against "real" popular-niche pairs sampled from the global co-occurrence counts in $\mathcal{X}$. The more confident this distinction by $\mathbf{D}$, the stronger the penalty on $\mathbf{G}$. To overcome the applied penalty, $\mathbf{G}$ must produce niche item recommendations that are correlated with the user's history. The model converges when both the synthetic and true niche-popular pairs align with the association structure learned by $\mathbf{D}$. We now formalize the strategy.

True & Synthetic Pair Sampling

- **True Pairs :** "True" popular-niche pairs $(i^p, i^n) \in \mathcal{I}^{\mathcal{P}} \times \mathcal{I}^{\mathcal{N}}$ are sampled from their global co-occurrence counts in $\mathcal{X}$. To achieve efficiency, we use the alias table method [105] which has $O(1)$ amortized cost when repeatedly drawing samples from the same discrete distribution, compared to $O(\mathcal{I}^{\mathcal{P}} \times \mathcal{I}^{\mathcal{N}})$ for standard sampling. We will denote the true distribution of pairs from $\mathcal{X}$ as $p_{true}(i^p, i^n)$.

- **Synthetic Pairs :** Synthetic pairs $(\tilde{i^p}, \tilde{i^n}) \in \mathcal{I}^{\mathcal{P}} \times \mathcal{I}^{\mathcal{N}}$ are drawn on a per-user basis with $\tilde{i^n} \propto f_{\mathbf{G}}(\tilde{i^n} \mid u, \mathcal{X})$, and $\tilde{i^p}$ randomly drawn from $\mathcal{X}_u^{\mathcal{P}}$. The number of synthetic pairs drawn for each user u is in proportion to $|\mathcal{X}_u^{\mathcal{P}}|$. We denote the resulting synthetic pair distribution $p_\theta(\tilde{i^p}, \tilde{i^n} \mid u)$, conditioned on the user u and parameters θ of the recommender model $\mathbf{G}$.

Note that the above terminology is borrowed from standard adversarial literature [53]. The source distribution of item pairs is generated by the recommender model, while the target distribution is modeled by the discriminator, guided by the underlying item associations.

Discriminative Adversary Training

The adversary $\mathbf{D}$ takes as input the synthetically generated item pairs $(\tilde{i^p}, \tilde{i^n})$ across all users, and an equal number of true pairs (i^p, i^n) sampled as described above. It performs two tasks:

- $\mathbf{D}$ learns latent representations $\mathbf{V} = [\mathbf{v}_i, i \in \mathcal{I}]$ for the set of items with dimensionality d.
- Additionally, $\mathbf{D}$ learns a discriminator function $f_\phi(i^p, i^n)$ simultaneously with $\mathbf{V}$ to estimate the probability of a pair (i^p, i^n) being drawn from $p_{true}(i^p, i^n)$.

$$\mathbf{D}_\phi(i^p, i^n) = \sigma(f_\phi(i^p, i^n)) = \frac{1}{1 + \exp(-f_\phi(\mathbf{v}_{i^p}, \mathbf{v}_{i^n}))} \tag{4.1}$$

We implement $\mathbf{D}_\phi$ via two simple symmetric feedforward ladders followed by fully connected layers (Figure 4.2). With the parameters of $\mathbf{G}$ (*i.e.,* θ) fixed, ϕ and $\mathbf{V}$ are optimized by stochastic gradient methods to maximize the log-likelihood of the true pairs, while minimizing that of synthetic pairs with a balance parameter μ,

$$\begin{aligned} \phi^*, \mathbf{V}^* = \arg\max_\phi \sum_{u \in \mathcal{U}} & \mathbb{E}_{(i^n, i^p) \sim p_{true}(i^p, i^n)} \left[\sigma(f_\phi(i^p, i^n))\right] + \\ & \mu . \mathbb{E}_{(\tilde{i^p}, \tilde{i^n}) \sim p_\theta(\tilde{i^p}, \tilde{i^n} | u)} \left[\log(1 - \sigma(f_\phi(\tilde{i^p}, \tilde{i^n})))\right] \end{aligned} \tag{4.2}$$

Recommender Model Training

The more confident the distinction of the fake pairs generated as $(\tilde{i^p}, \tilde{i^n}) \sim p_\theta(\tilde{i^p}, \tilde{i^n} \mid u)$ by adversary $\mathbf{D}$, the stronger the penalty applied to $\mathbf{G}$. As previously described, synthetic pairs $(\tilde{i^p}, \tilde{i^n})$ are drawn as $\tilde{i^n} \propto f_\mathbf{G}(\tilde{i^n} \mid u, \mathcal{X})$, and $\tilde{i^p}$ randomly drawn from $\mathcal{X}_u^\mathcal{P}$. Thus,

$$p_\theta(\tilde{i^p}, \tilde{i^n} \mid u) \propto \frac{1}{|\mathcal{X}_u^\mathcal{P}|} f_\mathbf{G}(\tilde{i^n} \mid u, \mathcal{X}) \tag{4.3}$$

For sanity, we shrink $p_\theta(\tilde{i^p}, \tilde{i^n} \mid u)$ as $p_\theta(u)$ in the following equations. Our goal is to reinforce the associations of the niche items recommended by $\mathbf{G}$ to the popular items in user history. This is achieved when the synthetic pairs cannot be distinguished from the true ones, *i.e.*, $\mathbf{D}_\phi(\tilde{i^p}, \tilde{i^n})$ is maximized for the synthetic pairs sampled for each user. Thus, there are two terms in the recommender's loss, first the base objective $\mathcal{O}_\mathbf{G}$ and second, the adversary term with weight λ. Note that $\mathbf{D}$'s parameters $\phi, \mathbf{V}$, are now held constant as $\mathbf{G}$ is optimized (alternating optimization schedule).

$$\begin{aligned}\theta^* &= \arg\max_{\theta} -\mathcal{O}_{\mathbf{G}} + \lambda \sum_{u \in \mathcal{U}} \mathbb{E}_{(\tilde{i^p}, \tilde{i^n}) \sim p_\theta(u)} \left[\log D(\tilde{i^p}, \tilde{i^n})\right] \\ &= \arg\min_{\theta} \mathcal{O}_{\mathbf{G}} + \lambda \sum_{u \in \mathcal{U}} \mathbb{E}_{(\tilde{i^p}, \tilde{i^n}) \sim p_\theta(u)} \left[\log(1 - D(\tilde{i^p}, \tilde{i^n}))\right]\end{aligned} \tag{4.4}$$

Since the second term (adversary) involves discrete item samples drawn on a per-user basis, it cannot be directly optimized by standard gradient descent algorithms. We thus apply policy gradient based reinforcement learning (REINFORCE) [195, 212] to approximate the gradient of the adversary term for optimization. Let us denote the gradient of the second term of eq. (4.4) for $u \in \mathcal{U}$ as $\nabla_\theta J^{\mathbf{G}}(u)$,

$$\begin{aligned}\nabla_\theta J^{\mathbf{G}}(u) &= \nabla_\theta \mathbb{E}_{(\tilde{i^p}, \tilde{i^n}) \sim p_\theta(u)} \left[\log(1 - D(\tilde{i^p}, \tilde{i^n}))\right] \\ &= \sum_{(\tilde{i^p}, \tilde{i^n}) \in \mathcal{I}^{\mathcal{P}} \times \mathcal{I}^{\mathcal{N}}} \nabla_\theta p_\theta(u) \log(1 + \exp(f_\phi(\tilde{i^p}, \tilde{i^n})) \\ &= \sum_{(\tilde{i^p}, \tilde{i^n}) \in \mathcal{I}^{\mathcal{P}} \times \mathcal{I}^{\mathcal{N}}} p_\theta(u) \nabla_\theta \log(p_\theta(u)) \log(1 + \exp(f_\phi(\tilde{i^p}, \tilde{i^n})) \\ &= \mathbb{E}_{(\tilde{i^p}, \tilde{i^n}) \sim p_\theta(u)} \left[\nabla_\theta \log(p_\theta(u)) \log(1 + \exp(f_\phi(\tilde{i^p}, \tilde{i^n}))\right] \\ &\approx \frac{1}{K} \sum_{k=1}^{K} \nabla_\theta \log(p_\theta(u)) \log(1 + \exp(f_\phi(\tilde{i^p}, \tilde{i^n}))\end{aligned} \tag{4.5}$$

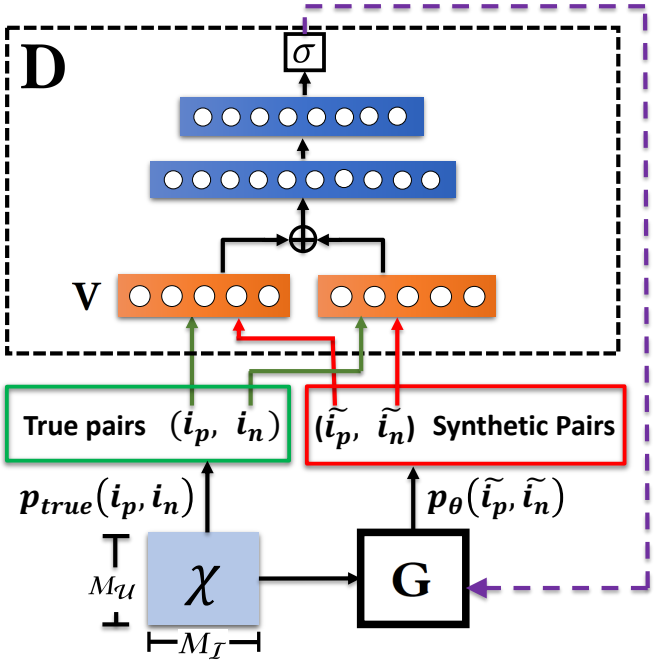
The last step introduces a sampling approximation, drawing K sample-pairs from $p_\theta(u)$. Before adversarial training cycles, the recommender $\mathbf{G}$ can be pre-trained with loss $\mathcal{O}_{\mathbf{G}}$, while $\mathbf{D}$ can be pre-trained with just the maximization term for true pairs. Our overall objective can be given by combining eq. (4.5), eq. (4.4),

$$\begin{aligned}\mathcal{O} = \min_{\theta} \max_{\phi} \mathcal{O}_{\mathbf{G}} + \lambda \sum_{u \in \mathcal{U}} &\mathbb{E}_{(i^p, i^n) \sim p_{true}(i^p, i^n)} \left[\log D_\phi(i^p, i^n)\right] + \\ &\mu . \mathbb{E}_{(\tilde{i^p}, \tilde{i^n}) \sim p_\theta(\tilde{i^p}, \tilde{i^n} | u)} \left[\log(1 - D_\phi(\tilde{i^p}, \tilde{i^n}))\right]\end{aligned} \tag{4.6}$$

On the whole, our framework employs a minimax strategy for iterative refinement: While the adversary progressively identifies finer distinctions between true and synthetic pairs, thus refining the learned inter-item association structure, the recommender incorporates it in the item recommendations made to users.

Also, note that the above iterative refinement process is architecture agnostic. Thus, we can integrate an appropriate recommender model depending on the application.

Figure 4.2: Architecture details for the discriminative adversary D trained in tandem with base recommender G.



4.4 EXPERIMENTS

In this section, we employ a Variational Auto-Encoder (VAE-CF) [115] and Denoising Auto-Encoder (CDAE) [224] as our base recommender models **G**. Results on the *ml-20m* dataset already indicate strong long-tail performance of stochastic VAE-CF (fig. 4.3) in comparison to deterministic CDAE [224]. Thus, performance gains in niche-item recall for VAE-CF with our adversarial training are particularly significant. We use two publicly available user-item datasets suitable for recommendation,

- **Movielens (*ml-20m*)**[1]: We binarized the available feedback matrix with a threshold of 5. Only users who watched atleast 10 movies were retained.
- **Ask-Ubuntu Stack Exchange**[2]: Tags were assigned to users if they Liked, Commented, Answered or asked a Question with the respective tags. Users with atleast 10 distinct tags were retained.

We employ strong generalization with train, test, and validation splits across the set of all users. Models are trained with all the user-item interactions of users in the training set,

[1]https://grouplens.org/datasets/movielens/20m/
[2]https://archive.org/details/stackexchange

Table 4.1: Composition of top-100 item recommendations to users in item popularity quartiles (Q1-Most popular Items, Q4 - Least popular items). Note the significant improvements in diversity for the CDAE base model which overfits to popular items in the inventory, resulting in only Q1 recommendations to all users. The augmented model exhbits recommendation compositions that better reflect item appearance.

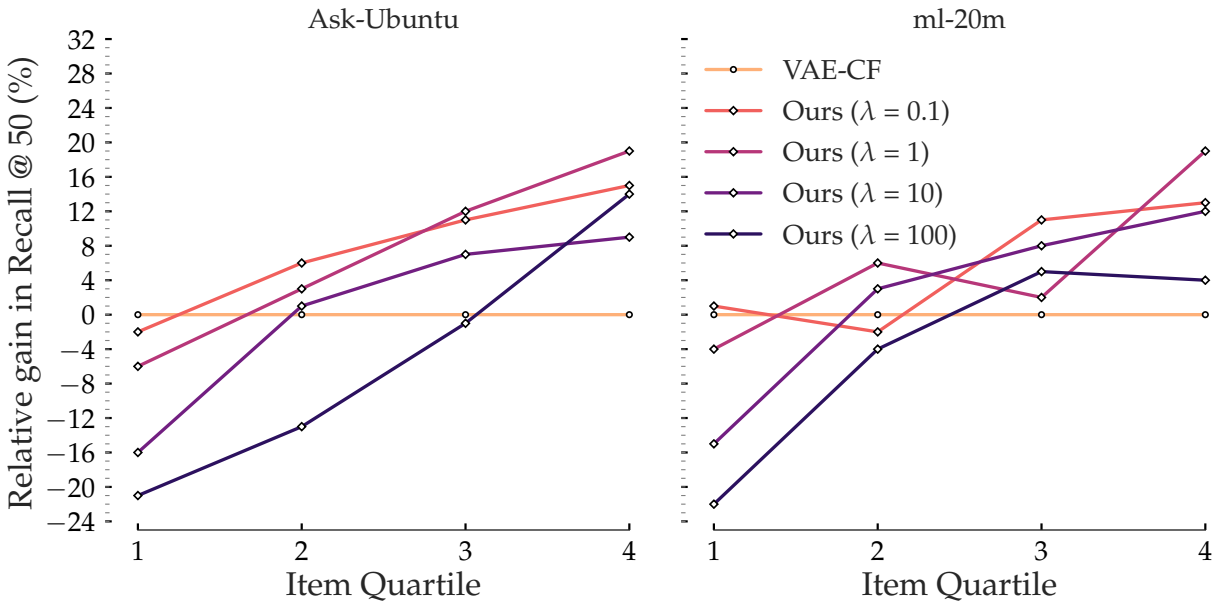
Method	**ml-20m**				**Ask-Ubuntu**			
	Q-1	**Q-2**	**Q-3**	**Q-4**	**Q-1**	**Q-2**	**Q-3**	**Q-4**
CDAE ($\mathbf{G}_1$)	74%	26%	0%	0%	97%	3%	0%	0%
$\mathbf{D}$+$\mathbf{G}_1(\lambda = 0.1)$	61%	23%	10%	6%	76%	14%	7%	3%
$\mathbf{D}$+$\mathbf{G}_1(\lambda = 1)$	62%	21%	11%	6%	73%	16%	6%	5%
$\mathbf{D}$+$\mathbf{G}_1(\lambda = 10)$	61%	19%	12%	8%	65%	19%	11%	5%
VAE-CF ($\mathbf{G}_2$)	64%	24%	8%	4%	60%	25%	9%	6%
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 0.1)$	58%	23%	12%	7%	53%	25%	12%	10%
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 1)$	59%	21%	13%	7%	55%	21%	13%	11%
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 10)$	59%	20%	13%	8%	54%	22%	14%	10%

while the interactions corresponding to the users in the validation and test sets are split in two. One subset is fed as input to the trained model, while the other is used to evaluate the system output (ranked list) on $NDCG@100, Recall@K$, $K = 20, 50$. The architecture and training procedure is adopted from [115] for comparison. We set tradeoff parameter λ to multiple values and explore it's effect on recommendation over different sets of items, grouped by popularity. The balance parameter μ was set to 1 and $\mathbf{D}$ used a feed-forward network with 2 hidden layers (300, 100) as in fig. 4.2 (*tanh* activations and sigmoid output layer) and 300-dimensional embedding layers. All items with less than 0.5% appearance (<1 `in` 200) were discarded, with negligible impact on results.

We will first analyze the composition of the `top` 100 recommendations of $\mathbf{D} + \mathbf{G}$, against $\mathbf{G}$ trained in isolation. All items are split into four quartiles based on their popularity. We demonstrate the effects of parameter λ on the `top` 100 items for the validation set users by analyzing the quartiles they appear from (Table 4.1). The recommendations from our model with higher values of λ improve the niche-tag coverage. Specifically, we show that the recommendation composition's significant changes do not degrade the overall recommender performance. This indicates a more balanced and diversified set of recommendations that do not rely on just the popular items to achieve high aggregate performance.

We analyze the overall recommendation performance against VAE-CF and CDAE in Table 7.5. Conventional baselines such as [68] have been shown to be significantly weaker than both our neural base recommender models in prior work [115, 224].

Figure 4.3: Relative improvement over VAE-CF with adversary training, measured for each item popularity quartile (R@50).



Note that CDAE does not make *any* niche item recommendations (Q3 and Q4). Integrating our adversary to train CDAE results in a significant jump in long-tail coverage. To further dissect the above results, we will now observe our relative gains in *Recall*@50 compared to VAE-CF for each item quartile (Figure 4.3). We chose VAE-CF for comparison due to it's stronger long-tail performance.

Gains by Quartile: As expected, our strongest gains are observed in Quartiles-3 and 4, which constitute long-tail items. Although there is a slight loss in popular item performance for $\lambda = 1$, this loss is not significant owing to the ease of recommending popular items with auxiliary models if required. We observe the values of tradeoff λ between 0.1 and 1 to generate balanced results.

We now analyze overall recommendation performance against VAE-CF and CDAE in Table 7.5 (**N** = NDCG, **R** = Recall). Even though our models recommend very different compositions of items (table 4.1), the results exhibit modest overall improvements for $\lambda = 0.1$ and $\lambda = 1$ over both the base recommenders.

The additional niche item recommendations to users are coherent since there is no aggregate recommender performance drop. However, larger λ parameter values hurt the aggregate recommender performance by over-penalizing minor distributional differences at the expense of relevance. It is thus essential to balance the adversary objective and base recommender to obtain strong overall results.

Table 4.2: Overall recommendation performance on the ml-20m and Ask-Ubuntu datasets is either superior to, or at par with the respective base models despite massive improvements in long-tail item appearance (Table 4.1).

Method	**ml-20m**			**Ask-Ubuntu**		
	N@100	**R@20**	**R@50**	**N@100**	**R@20**	**R@50**
CDAE ($\mathbf{G}_1$)	0.34	0.27	0.37	0.29	0.30	0.46
VAE-CF ($\mathbf{G}_2$)	0.51	0.44	0.57	0.42	0.45	0.59
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 0.1)$	0.53	0.45	0.59	0.43	0.46	0.61
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 1)$	0.52	0.44	0.58	0.42	0.46	0.59
$\mathbf{D}$+$\mathbf{G}_2(\lambda = 10)$	0.48	0.41	0.55	0.40	0.43	0.56
$\mathbf{D}$+$\mathbf{G}_2(\lambda$=100)	0.42	0.37	0.51	0.38	0.41	0.53

4.5 CONCLUSION AND FUTURE WORK

4.5.1 Chapter Summary

In this chapter, we developed and investigated a self-supervised adversarial learning framework to overcome sparsity in long-tail item recommendation and learn effective *neural / vector* representations of long-tail items. Our approach's strength lies in its ability to reweight each item-item association differentially. We contextually reweight the aggregate item co-occurrences to filter and adapt to each item's eccentricities.

Our approach generalizes conventional neighbor models [129] which adopt static association criteria to organize the item representation space. Instead of imposing static precomputed item-item metrics on the item representation space, we jointly learn the associated recommendation model and the task-focused association structure of the item-set, guided by the aggregate co-occurrence feedback. Our approach significantly improved the long-tail performance of VAE-CF [115]. This robust stochastic model outperforms alternate neural recommenders (CDAE [224]) by a significant margin on the long-tail items, even without adversarial augmentation.

4.5.2 Improvements to the Proposed Approach

We broadly categorize improvements to the proposed adversarial framework into two buckets. The first dimension includes the input variables to the source or target distributions. Integration of inter-user or cross associations across the user and item embedding spaces learned by the base recommender could prove valuable, in addition to the item-item asso-

ciations. A two-phase learning approach is also feasible. We can first execute the method to reproduce the contextually filtered item-item association structure, followed by a second phase to introduce the inter-user / user-item similarities in the learning objective. The second phase could incorporate a local bounded parameter search to avoid degeneracy or mode collapse challenges.

Although our empirical results indicate reasonable model convergence with two diverse neural collaborative filtering models, we plan to explore the Wasserstein metric [8] to improve and stabilize generator updates (i.e., neural recommender) when the critic outperforms the recommendation model. The linear-shaped gradients with the Wasserstein objective function minimize the vanishing gradient challenges observed with conventional adversarial models.

4.5.3 Extending Grouping Approaches to Multimodal Scenarios

In the previous two chapters, we discussed modeling solutions to target and mitigate skew and sparsity on both the user-side and the item-side by forming skew-aware groups of users and learning to represent inter-item associations, respectively. While both solutions admit a choice of user profiles and item feature representations, they do not account for simultaneous and independent data generation processes, i.e., a multimodal setting Section 1.2.1. A well-studied example of multimodal recommendation is the *social recommendation* problem [114], where users engage in both item purchases and user-user social interactions. In such settings, effectively representing each modality of user participation requires different modeling hypotheses. For instance, signed networks [34] necessitate polarity-aware representation models as opposed to unsigned social networks [89]. Further, we must independently evaluate user activity across the data-modalities to generate a joint representation.

The next chapter develops generalizable abstractions of multimodal user representation to combine data-modalities towards recommendation and inference tasks. We identify the implicit *adversarial problem* of learning to attribute each training sample to one among many data-modalities and address the learning problem in a model and modality agnostic manner. Note the implicit relationship between the grouping mechanisms proposed in the previous two chapters and the multimodal setting; We can first independently group users or items within each data-modality and then leverage the learned groupings towards the multimodal attribution problem.

In the next chapter, we combine and subsume the skew and sparsity-aware grouping mechanisms (developed in the previous chapters) across each data-modality towards a joint representation for each entity, independent of the data and platform specifics.

CHAPTER 5: AN ADVERSARIAL FRAMEWORK FOR MULTIMODAL RECOMMENDATION AND INFERENCE

This chapter proposes a novel framework to incorporate social regularization for item recommendation. Social regularization grounded in ideas of homophily and influence appears to capture latent user preferences. However, there are two key challenges: first, the importance of a specific social link depends on the context, and second, a fundamental result states that we cannot disentangle homophily and influence from observational data to determine the effect of social inference. Thus, we view the attribution problem as inherently adversarial, where we examine two competing hypotheses– social influence and latent interests–to explain each purchase decision.

We make two contributions. First, we propose a modular, adversarial framework that decouples the architectural choices for the recommender and social representation models, for social regularization. Second, we overcome degenerate solutions through an intuitive contextual weighting strategy that supports an expressive attribution to ensure informative social associations play a more significant role in regularizing the learned user interest space. Our results indicate significant gains (5-10% relative Recall@K) over state-of-the-art baselines across multiple publicly available datasets.

5.1 INTRODUCTION

This chapter proposes a novel framework to incorporate social regularization for item recommendation. The motivating idea is to leverage social relation structure to capture unseen user preferences appropriately. Social correlation theories such as homophily [136] and notions of influence or conversely, susceptibility [43, 135] lend support to the idea of social regularization.

The social recommendation problem has received significant attention in the research community. The social connections among users (in the form of explicit social networks) and among items (such as induced co-occurrence graphs [223]) can play a critical role in improving recommendation quality in the presence of data sparsity and in addressing long-tail concerns [95, 96, 234]. The use of homophily encodes the assumption that social connections share similar preferences [76, 129]. This assumption constrains our ability to combine user interests and social factors effectively [213].

Exposure models [114, 213] adopt a more nuanced *exposure precedes action* lens. Each user's exposure to his contacts' preferences limits her potential actions. The exposure approach's weakness is that it cannot explicitly prioritize specific preferences originating from

different contacts based on the available context. For instance, Alice may prefer Bob's suggestions on books but follow Mary (another connection) for music. Thus social contacts can vary in the extent of influence they assert. Their relative importance depends on a contextual mixture of factors that we can infer from their interest representations and social structure.

Shalizi and Thomas [183] proved a key negative result—homophily and influence are fundamentally confounded in observational studies. In other words, we cannot disentangle peer influence from latent interests using observational data. Thus, the attribution problem is inherently adversarial, where we examine two competing hypotheses– social influence and latent interests–to explain each purchase decision.

The social regularization problem is readily amenable to a Generative Adversarial Network (GAN) formulation, whereby the social and interest factors of each user complete to explain each user's observed actions. As a result of such a training process, the most contextually relevant social information regularizes each user's interest space.

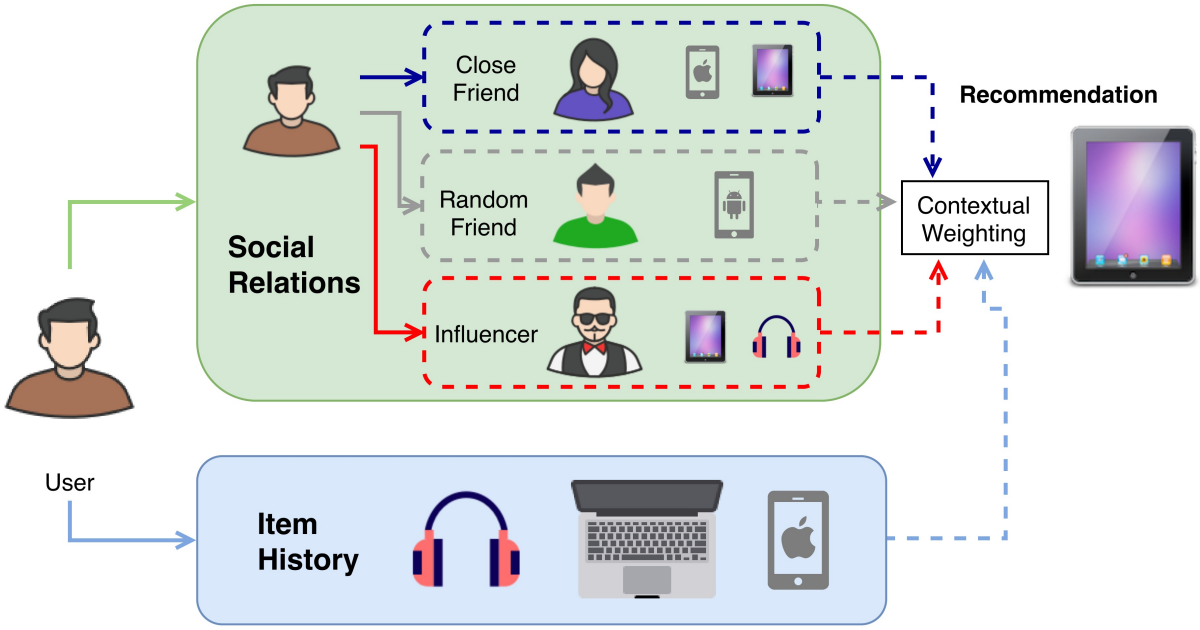
Furthermore, an adversarial formulation provides a modular framework to decouple the architectural choices for the recommender and social representation models, enabling a wide range of recommender applications. Degenerate solutions are a significant challenge in vanilla GAN implementations that lack a sufficiently expressive attribution strategy. We overcome this challenge through an intuitive contextual weighting strategy to ensure informative social associations play a larger role in regularizing the learned user interest space. Our contributions are as follows:

Modular Adversarial Formulation: To the best of our knowledge, ours is the first work to address the social recommendation problem with an architecture-agnostic formulation. In contrast to prior work, we integrate *state-of-the-art* recommender architectures and social representations models.

Expressive Attribution Strategy: We unify the interest and social distributions of users by contextually attributing their purchase decisions across these two representations. Thus, we incorporate diversity across users' social links and each link's varied impact on their purchase decisions, enabling a more expressive interest space. Our qualitative analysis in *Section* 7.6 indicates we can preferentially select important social relations to improve recommendations.

Robust Experimental Results: We integrate three state-of-the-art social-agnostic recommender models in our adversarial framework and observe significant gains with adversarial training across multiple public datasets (4-10% relative Recall@K). Further, we categorize and study the extent of regularization imposed by social samples. We find that relations between influential users tend to play an essential role in regularizing interests. Further, links

Figure 5.1: Social contacts and item histories of users must be contextually weighted to evaluate their potential impact on future purchases.



across peers (similar activity levels) are better regularizers than those with highly active users. Finally, our stochastic optimization approach is resilient to lossy social data.

We organize the rest of the chapter as follows. In Section 7.7 we discuss related work. We formally define the problem and propose our approach in Section 5.3 and Section 7.3. We then present our experimental results in Sections 5, perform qualitative analysis of our model in Section 5.5.4, Section 5.5.5, Section 5.5.6 and discuss it's limitations in Section 6.6.9, finally concluding in Section 7.8.

5.2 RELATED WORK

Historically, matrix factorization (MF) has been the most popular collaborative filtering approach [134, 143] and forms the basis for efficient modern recommenders [59] and effective deep-learning strategies [60, 115, 224]. Prior efforts to integrate social structure in the latent interest space employed static hypotheses [76, 129] that do not incorporate additional context. Incorrect prioritization of social links could hurt recommendation quality. A second line of work has looked at transfer learning [150], auxiliary facet integration in MF [113] and trust propagation [72]. While these approaches augment [134], they are expensive and incompatible with neural methods [60].

More recently, *exposure* models [114, 213] view user actions as subsets of their social exposure. However, they do not separate sources of exposure; an item exposed by a subject expert is likely to have a greater impact; for instance. Wu et al. [223] propose a multi-armed bandit (MAB) solution to contextually pick *one-of-many* factors to explain purchases. Although it incorporates context, it is intuitive to explore a continuous version of Wu et al. [223] that differentially combines factors rather than pick just one.

In recent times, neural social-agnostic recommenders obtained *state-of-the-art* results with user-item rating information [115, 198, 224]. Further, a wide range of formulations and convolutional models have been proposed to effectively embed social networks [34, 89, 182, 206] with diverse link semantics. Our work unifies these two lines of work. While we address the weaknesses of static social integration models with a dynamic contextual regularization approach, our primary focus is to enable diverse recommenders to effortlessly integrate with the most suitable social models, enabling more interesting and relevant recommendations.

5.3 PROBLEM AND MODEL FORMULATION

In this section, we describe relevant preliminaries and formalize our problem definition. We discuss the implications of structurally regularizing user representations and provide an intuitive solution to avoid converging to degenerate solutions. Finally, in Section 5.3.4, we describe our approach with a modular adversarial framework for social recommendation.

5.3.1 Preliminaries

We consider the implicit feedback setting with users $\mathcal{U}$, items $\mathcal{I}$ and binary user-item interaction matrix $\mathcal{Z} \in \mathbb{B}^{|\mathcal{U}|\times|\mathcal{I}|}$ ($\mathbb{B} = \{0, 1\}$). Further, $\mathcal{N} \in \mathbb{B}^{|\mathcal{U}|\times|\mathcal{U}|}$ denotes the explicit social link matrix between the users, we abuse $\mathcal{N}$ to denote both, the social network and its user adjacency matrix. Although we assume undirected social links, the extension to the directed case is straightforward. The total number of user-item interactions and social links are denoted $|\mathcal{Z}|$, $|\mathcal{N}|$ respectively.

Latent-factor social recommenders learn the latent social and item interest representations for each user. Without loss of generality, let us denote the social embedding matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{U}|*d_S}$ and the interest embeddings $\mathbf{X} \in \mathbb{R}^{|\mathcal{U}|*d_X}$. Note that $\mathbf{X}_u, \mathbf{S}_u$ denote the rows for user u. Further, we denote item embeddings $\mathbf{I} \in \mathbb{R}^{|\mathcal{I}|*d_I}$. Given any user embedding matrix $\mathbf{E}$, we can compute user-user similairities in $\mathbf{E}$'s latent space as,

$$p_{\mathbf{E}}(u, v) \propto \sigma(\mathbf{E}_u \cdot \mathbf{E}_v) \tag{5.1}$$

where $u, v \in \mathcal{U}$ and $\sigma(x) = 1/(1 + e^{-x})$. The social and interest embedding spaces $\mathbf{S}$, $\mathbf{X}$ model the social neighborhoods and item interactions of users, and thus induce different user-user proximities p_S, p_X when placed in Equation (5.1). Social regularization of interest space $\mathbf{X}$ is achieved by introducing a shared coordinate structure between $\mathbf{S}$ and $\mathbf{X}$. At the heart of this problem is the choice of a suitable distance metric in the embedding space. Historically metric learning approaches have learned effective distance functions in similarity, distance-based tasks [99], and recently in Collaborative Filtering [65]. Thus, the question follows,

5.3.2 Can we Learn a Distance Metric to Regularize Interest Embeddings $\mathbf{X}$ with Social Structure $\mathbf{S}$?

Let us consider the embeddings to lie in metric space $\mathbb{M}$ with any metric distance measure $\mathbf{D}_M$. This is the most general form with no constraint on the form of $\mathbf{D}_M$. To transfer structure under metric $\mathbf{D}_M$, for each user-item interaction $(u, i) \in \mathcal{Z}$ we obtain pairwise loss $\|\mathbf{X}_u - \mathbf{I}_i\|_{\mathbf{D}_M} \to 0$ (with user interest embeddings $\mathbf{X}$ and item embeddings $\mathbf{I}$). Similarly, for social links $(u, v) \in \mathcal{N}$, we obtain $\|\mathbf{S}_u - \mathbf{S}_v\|_{\mathbf{D}_M} \to 0$ (with social embeddings $\mathbf{S}$).

When we convert the above pairwise losses to equalities, it is easy to show that we obtain an over-specified system with only degenerate solutions (i.e., assigning the same interest embedding $\mathbf{X}_u$ to all $u \in \mathcal{U}$) due to the identity property of any $\mathbf{D}_M$.

Note the fundamental adversarial nature of the regularization problem in any metric embedding space. No solution can perfectly satisfy the above system if any pair of connected users have different item ratings. The continuous loss version of this system (optimized via gradient methods) moves towards some degenerate solution with user embeddings $\mathbf{X}_u$ collapsing inwards. The resulting loss in the expressivity of interest space $\mathbf{X}$ causes reduced diversity in recommendations (especially for users sharing first, second-order connections in $\mathcal{N}$). We refer to this as *interest space collapse*.

5.3.3 Can we Transfer the Structure of $\mathbf{S}$ to $\mathbf{X}$ without Affecting Interest Space Expressivity?

The user-user similarities (or pairwise proximities) $p_{\mathbf{S}}(u, v)$ and $p_{\mathbf{X}}(u, v)$ from Equation (5.1) represent the structures of the embedding spaces $\mathbf{S}$ and $\mathbf{X}$. Ideally, we must converge $p_{\mathbf{S}}$ and $p_{\mathbf{X}}$ to a *meaningful*, i.e. *non-degenerate* equilibrium to avoid interest space collapse.

We avoid the over-specification problem in section 5.3.2 by introducing pair-specific translations for each pairwise constraint, i.e, the system is now of the form $\|\mathbf{S}_u - \mathbf{S}_v\|_{\mathbf{D}_M} \to w(u, v)$

where w is a learned function of the user context. This added expressivity enables a non-degenerate encoding in interest space $\mathbf{X}$, while retaining a contextually transformed version of the social structure via $w(u, v)$.

We now describe and motivate our modular stochastic approach to solve the continuous version of the above regularization problem in an adversarial framework similar to GANs [53]. Social regularization is naturally amenable to such an approach due to the competing interest and social spaces. Further, we can socially regularize any gradient optimizable recommender model with our approach, agnostic to its architecture.

5.3.4 Adversarial Social Regularization

The Generator ($\mathbf{G}$) in the GAN framework is a neural model that synthesizes data samples, $\mathbf{y}_G \in \mathbb{R}^d$, drawn from the source distribution $P_G(\mathbf{Y})$ over $\mathbb{R}^d$ induced by $\mathbf{G}$. The Discriminator ($\mathbf{D}$), on the other hand, attempts to construct a decision boundary to distinguish synthetic samples $\mathbf{y}_G$ drawn from the source distribution against true (positive labeled) samples drawn from an unknown target distribution. The generator is trained to synthesize data points that mimic target samples, hence encoding the target distribution.

In our formulation, the social-agnostic base recommender model learns a scoring function $f_{\mathbf{G}}(i \mid u, \mathcal{Z}), i \in \mathcal{I}, u \in \mathcal{U}$ to rank items given u's history $\mathcal{Z}_u$ by minimizing continuous, differentiable objective $\mathcal{O}_{\mathbf{G}}$ over its parameters θ_G. As a result, it learns the interest embeddings $\mathbf{X}$, and the source user-user similarity $p_{\mathbf{X}}(u, v)$ in the interest space $\mathbf{X}$ (Equation (5.1)). We will refer to the base recommender as the generator $\mathbf{G}$ in our formulation.

On the other hand, social network $\mathcal{N}$ induces a target user-user similarity that the generator must learn to imitate to regularize its interest space $\mathbf{X}$. To compute the target user-user similarity, we apply a Graph Auto-Encoder [88] on network $\mathcal{N}$ and place the learned embeddings in Equation (5.1). We will denote this as $p_{\mathcal{N}}(u, v)$, the target or *true* user-user similarity from $\mathcal{N}$.

Finally, discriminator $\mathbf{D}$ learns an independent social embedding space $\mathbf{S}$ for users separate from social network $\mathcal{N}$. The discriminator induces social proximity, $p_{\mathbf{S}}(u, v)$ of users in its latent social space, forming the link between the target $p_{\mathcal{N}}(u, v)$ and source $p_{\mathbf{X}}(u, v)$, and attempts to move them closer. We highlight two key advantages of the adversarial regularization strategy —

1) It enables our modular optimization strategy (Section 5.3.5), providing flexibility in the recommender $\mathbf{G}$ and discriminator $\mathbf{D}$'s architectures. In our experiments, we substitute and show gains for multiple strong neural recommenders as $\mathbf{G}$ with a convolutional discriminator [89] to capture social representations.

2) We enable pair-specific expressivity in Section 5.3.6 as motivated in section 5.3.3 to provide a wider choice of target $p_{\mathbf{X}}$ given source $p_{\mathcal{N}}$, hence reducing the likelihood of interest-space collapse and providing contextual social structure integration in $\mathbf{X}$.

5.3.5 Structure Regularization

We propose a robust stochastic approach to represent source $p_{\mathcal{X}}$ and target $p_{\mathcal{N}}$ with a finite number of user-user pair samples drawn from each space. We evaluate the likelihood of each sampled user pair (u,v) with the discriminator embeddings $\mathbf{S}$, i.e., $p_{\mathbf{S}}(u, v)$.

Ideally, the discriminator should assign higher likelihoods to the *true-pairs* sampled from the target distribution $p_{\mathcal{N}}$ (denoted (u_+, v_+)) modeled by the discriminator, and lower likelihoods to *fake-pairs* sampled from the source $p_{\mathbf{X}}$ (denoted (u_-, v_-)), while the generator's goal is to confuse the discriminator, i.e., maximize expected fake-pair likelihood $\mathbb{E}(p_{\mathbf{S}}(u_-, v_-))$. Thus, we obtain overall objective $\mathcal{O}$,

$$\mathcal{O} = \min_{\mathbf{X}} \max_{\mathbf{S}} \Big(\mathbb{E}_{(u_+,v_+)\sim p_{\mathcal{N}}} \log p_{\mathbf{S}}(u_+, v_+) + \mu . \mathbb{E}_{(u_-,v_-)\sim p_{\mathbf{X}}} \log \big(1 - p_{\mathbf{S}}(u_-, v_-)\big) \Big) \tag{5.2}$$

where μ is the balance parameter. When we optimize $\mathcal{O}$, $\mathbf{G}$ learns X so that *fake-pairs* $(u_-, v_-) \sim p_{\mathbf{X}}$ confuse the discriminator i.e., maximize $\log p_{\mathbf{S}}(u_-, v_-)$.

Conversely, the discriminator maximizes the expected *true-pair* likelihood $\log p_{\mathbf{S}}(u_+, v_+)$ and minimize *fake-pair* likelihood $\log p_{\mathbf{S}}(u_-, v_-)$. The expectations $E_{(u,v)}$ are averaged over ϵ *fake* and *true-pair* samples each to compute the gradient updates to the model parameters (policy-gradient approximation) [212, 221].

We find in Section 5.5.5 that the number of *fake* and *true* user pair samples ϵ required for robust convergence is $\leq 2\%$ of the distinct user pair count ($|\mathcal{U}|^2$), enabling much faster training than Coordinate Transfer Learning [150]. Further, our approach is observed to be robust to lossy social data (Figure 5.9). We perform stratified sampling to equally represent all users in the *fake* and *true-pair* sample sets, denoted ϵ_-, ϵ_+ respectively ($|\epsilon_-| = |\epsilon_+| = \epsilon$).

Equation (5.2) stochastically moves the user interest structure in $p_{\mathbf{X}}$ closer to $p_{\mathcal{N}}$. However, it may still lead to a partial collapse of the interest space $\mathbf{X}$ since it lacks the pairwise expressivity defined in Section 5.3.3. We now describe an intuitive pair weighting strategy to enable a wider choice of the target $p_{\mathbf{X}}$ by learning to prioritize the most important parts of $p_{\mathcal{N}}$ (contextual social regularization).

5.3.6 User Pair Weighting to Avoid Interest Space Collapse

In our formulation, interest space collapse can cause $\mathbf{G}$ to learn interest space $\mathbf{X}$ with shallow variety, moving towards degenerate solutions to the Min-Max game in Equation (5.2). We can prevent interest space collapse by varying the regularization induced by each user pair sample, thus increasing model expressivity. This effectively differentiates social and interest context at the pair sample level, such as close friend links vs. celebrity-follower links, correlation of the interersts of each social contact to a user, expertise etc. The augmented Min-Max objective is as follows —

$$\mathcal{O} = \min_{\mathbf{X}} \max_{\mathbf{S}} \Big(\mathbb{E}_{(u_+,v_+)\sim p_{\mathcal{N}}} \log p_{\mathbf{S}}(u_+, v_+) + \mu . \mathbb{E}_{(u_-,v_-)\sim p_{\mathbf{X}}} w(u_-, v_-) \log \big(1 - p_{\mathbf{S}}(u_-, v_-)\big) \Big) \tag{5.3}$$

Note that the above transformation regularizes the product $w(u, v) \times p_{\mathbf{X}}(u, v)$ against $p_{\mathbf{S}}$ (instead of just $p_{\mathbf{X}}$ against $p_{\mathbf{S}}$), enabling a much wider choice of $\mathbf{X}$. The contextual weighting function $w(u, v)$ accounts for diverse social relations with varying levels of interest sharing. Also note that contextually weighting *fake-pairs* is sufficient to expand the expressivity of $\mathbf{X}$, we do not need to weight the *true-pairs*. Thus, $w(u, v)$ needs to be computed only for the ϵ *fake-pairs* in sample set ϵ_- and adds limited overhead ($\epsilon \ll |\mathcal{U}|^2$).

5.4 MODEL DETAILS

We now describe the architectural details of $\mathbf{G}$, contextual pair weighting function $w(u, v)$, discriminator $\mathbf{D}$ and an alternating optimization approach to train these modules.

5.4.1 Generator Architecture

We limit our architectural assumptions on the generator (or recommender) model to the most general hypotheses, namely $\mathbf{G}$ learns the user interest embeddings $\mathbf{X}$ (and any other parameters θ_G) by optimizing a differentiable continuous objective function $\mathcal{O}_{\mathbf{G}}$. In our experiments, we demonstrate generalizability by showing social regularization gains on the three best-performing neural recommender baselines in our framework.

Fake-pair Sampling: *Fake-pairs* (u_-, v_-) are sampled by first choosing u_-, and then sampling $v_- \propto p_{\mathbf{X}}(u_-, v_-)$. We stratify the samples per user, so that each user appears in at least $\epsilon/|\mathcal{U}|$ pairs.

True pair Sampling: True pairs are representative of the underlying social network structure. They are sampled similar to the fake pairs above by replacing the generator embeddings with Graph Auto-Encoder [88] embeddings from social network $\mathcal{N}$.

We now describe the parametrization of the contextual weighting function $w(u_-, v_-)$.

5.4.2 Attentive Hadamard Weighting

Multiplicative cross-factors between the context features of a pair of users are natural indicators of homogeneity and heterogeneity. For instance, the multiplicative cross-factors across appropriate dimensions of interest embeddings $\mathbf{X}_u$ and $\mathbf{X}_v$ can help us infer shared interests and differences between pair (u, v). A similar intuition generalizes across other user features.

Towards this transformation, we propose a simple Hadamard projection approach to achieve low-rank bilinear pooling of user features in the contextual weight function $w(u, v)$. We learn a projector matrix $\mathbf{P} \in \mathbb{R}^{N * d_w}$, where d_w is the dimensionality of contextual user features. Each row of the projector matrix, $\mathbf{P}_i, i \in [1, ..., N]$, represents a unique transformation on the user context. For each user pair sample (u, v), the input representations are projected as (using interest embeddings $\mathbf{X}_u$ as the contextual features) —

$$\mathbf{X}_u^i = \mathbf{X}_u \odot \mathbf{P}_i,\ \mathbf{X}_v^i = \mathbf{X}_v \odot \mathbf{P}_i \tag{5.4}$$

where $\odot$ denotes the Hadamard product operation. We then compute attention weights for each projector to represent the alignment of the users under it's projected dimensions, i.e.,

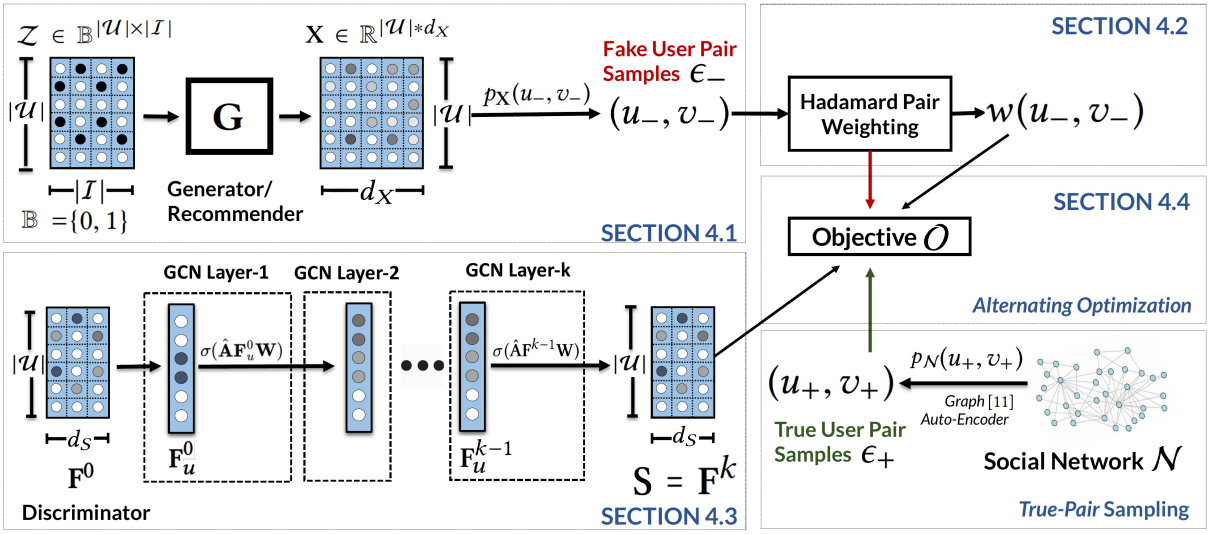
$$a_n(u, v) = \frac{\exp(\mathbf{X}_u^n . \mathbf{X}_v^n)}{\sum_{i=1}^{N} \exp(\mathbf{X}_u^i . \mathbf{X}_v^i)} \tag{5.5}$$

The higher weight $a_n(u, v)$, stronger the multiplicative cross-factors for pair (u, v) across dimensions projected by $\mathbf{P}_n$. We then compute pair alignment vector $\mathbf{A}(u, v)$ as a weighted projector sum,

$$\mathbf{A}(u, v) = \sum_{n=1}^{N} a_n(u, v) \mathbf{P}_n \tag{5.6}$$

Alignment vector $\mathbf{A}(u, v)$ denotes the nature of the relation between users (u, v). It is then transformed to the pair weight value $w(u, v)$ through a single feed-forward layer. Additionally, we introduce a batch sparsity regularizer across the N projectors to incentivize

Figure 5.2: Architecture diagram illustrating the model components and computation of the loss terms that appear in the adversarial objective in Equation (5.3). We do not place any restrictions on the architecture of recommender $\mathbf{G}$.



sparsity and diversity in their projected dimensions.

There is a loss in expressivity moving from $\mathbf{A}(u, v)$ to weight $w(u, v)$ for a user pair. We can address this by transforming each projection and their interactions separately to obtain a fine-grained joint expression. We leave this investigation to future work.

5.4.3 Discriminator Architecture

The discriminator architecture $\mathbf{D}$ learns social representations $\mathbf{S}$ by optimizing the Min-Max objective in eq. (5.3). It hence parametrizes the proximity $p_{\mathbf{S}}(u, v)$. We explore a few simple architectural choices to keep the computational overhead to a minimum—

Inner Product Discriminator: The inner product discriminator parametrizes the likelihood $p_{\mathbf{S}}(u, v)$ as $1/(1 + e^{-\mathbf{S}_u.\mathbf{S}_v})$. We also expore a bilinear form $p_{\mathbf{S}}(u, v) = 1/(1 + e^{-(\mathbf{S}_u)^{\mathbf{T}}\mathbf{W}_{\mathbf{B}}\mathbf{S}_v})$. Thus the embeddings $\mathbf{S}$ (and bilinear weight parameter $\mathbf{W}_{\mathbf{B}}$) are learned directly by optimizing eq. (5.3) with these functional forms of $p_{\mathbf{S}}$.

MLP: We apply a RelU bi-layer perceptron to encode the normalized Laplacian matrix $\mathbf{L}$ of the social network $\mathcal{N}$ to the latent social embeddings $\mathbf{S}$. Note that $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{1/2}$ where $\mathbf{A}$ and $\mathbf{D}$ denote the adjacency and degree matrices of $\mathcal{N}$. Once again, $p_{\mathbf{S}}(u, v) = 1/(1 + e^{-\mathbf{S}_u.\mathbf{S}_v})$ where $\mathbf{S}_u = \mathbf{MLP}(\mathbf{L}_u)$.

Graph Convolutional Network: The convolution operations on the social network $\mathcal{S}$

is given by the product of input user features $\mathbf{F}_u \in \mathbb{R}^n$ with learned filter g_θ in the fourier domain,

$$g_\theta * \mathbf{F}_u = g_\theta(\mathbf{Q}\Lambda\mathbf{Q}^T)\mathbf{F}_u = \mathbf{Q}g_\theta(\Lambda)\mathbf{Q}^T\mathbf{F}_u \tag{5.7}$$

where rows of $\mathbf{Q}$ are the eigenvectors of Laplacian $\mathbf{L}$.

To circumvent the expensive eigen-decomposition of the Laplacian, Defferrard et al. [32] proposed to approximate filter $g_\theta(\Lambda)$ with truncated Chebyshev polynomials $T_k(x)$ to the k^{th} order. This approximation results in k-localization, i.e. node representations incorporate k-hop neighborhoods. Kipf and Welling [89] further simplified this to a first-order linear form (GCN). We stack k GCN layers to condition $\mathbf{S}$ on the k-hop social neighborhoods of users.

The feature inputs to the k^{th} GCN layer are the user representations from the previous layer, $\mathbf{F}^{k-1} \in \mathbb{R}^{|\mathcal{U}| \times d_{k-1}}$, where d_{k-1} is the dimensionality of the $(k\text{-}1)^{th}$ GCN layer. Thus,

$$\mathbf{F}^k = \sigma(\hat{\mathbf{A}}\mathbf{F}^{k-1}\mathbf{W}),\ \ \hat{\mathbf{A}} = \mathbf{D}^{-1/2}\hat{\mathbf{A}}\mathbf{D}^{-1/2} + \mathbf{I} \tag{5.8}$$

Note that inputs $\mathbf{F}^0$ are the node features of the users in $\mathcal{N}$. We used one-hot feature inputs in our experiments. The social embedding matrix $\mathbf{S}$ is k^{th} layer ouput i.e., $\mathbf{S} = \mathbf{F}^k$. Thus,

$$p_{\mathbf{S}}(u, v) = 1/(1 + e^{-\mathbf{F}^k_u . \mathbf{F}^k_v}) \tag{5.9}$$

We set the dimensions of all GCN layers F^k and social embeddings S to the same value d_S. We find two and three-layer GCNs (k=2,3) to outperform the Inner-product and MLP variants significantly in our experiments. Although we expect further improvements with architectures such as Graph Attention [205], we leave this investigation to future work.

5.4.4 Model Optimization

We now describe our alternation optimization approach and the specific objective functions for each of the previous three modules. The optimization objective for each module is obtained by separating out the relevant terms from Equation (5.3).

Generator Objective: In the absence of our adversarial framework, recommender (generator) $\mathbf{G}$ optimizes $\mathcal{O}_{\mathbf{G}}$ to learn $\mathbf{X}$ and associated parameters θ_D. The adversarial term optimizes the discriminator likelihood of G's *fake-pair* samples,

$$\mathbf{X}, \theta_D = \arg\min_{\mathbf{X}, \phi} \left(\mathcal{O}_{\mathbf{G}} + \frac{\lambda}{\epsilon} \sum_{\epsilon_-} w(u_-, v_-) \log\left(1 - p_{\mathbf{S}}(u_-, v_-)\right) \right) \tag{5.10}$$

Note that constant λ controls the adversary weight (i.e., overall regularization strength). Also note that minimizing the second term is equivalent to maximizing $w_(u_-, v_-) \log \Big(p_{\mathbf{S}}(u_-, v_-)\Big)$. The generator updates $\mathbf{X}$ to increase the likelihood of generating *fake-pairs* with higher contextual weights and discriminator likelihoods.

Discriminator Objective: The discriminator learns social space $\mathbf{S}$ and associated parameters θ_D, to maximize the similarity or likelihood $p_{\mathbf{S}}$ of the *true-pairs* and minimize that of the *fake-pairs* sampled from the generator's interest space $\mathbf{X}$,

$$\mathbf{S}, \phi = \arg\max_{\mathbf{S}, \phi`} \Big(\frac{1}{\epsilon} \sum_{\epsilon_+} \log p_{\mathbf{S}}(u_+, v_+) + \frac{\mu}{\epsilon} \sum_{\epsilon_-} w_{(u_-, v_-)} \log \big(1 - p_{\mathbf{S}}(u_-, v_-)\big) \Big) \tag{5.11}$$

As a result, the discriminator progressively learns finer distinctions between samples from $p_{\mathbf{X}}$ and $p_{\mathcal{N}}$. In response, $\mathbf{G}$ selectively embeds the social structure to generate harder *fake-pair* samples. Note that pair weighting enables, in theory, an infinitely wide choice for $p_{\mathbf{X}}$ to differ from $p_{\mathcal{N}}$. In practice, however, model expressivity depends on the context features provided to the weighting module.

Pair Weighting Objective: The Hadamard network learns to prioritize pairs that result in minimizing $\mathbf{G}$'s loss while keeping $\mathbf{X}, \mathbf{S}$ fixed. This translates to the following objective.

$$\mathbf{P}, \theta_w = \arg\min_{\mathbf{P}, \theta_w} \left(\frac{\lambda}{\epsilon} \sum_{\epsilon} w(u_-, v_-) \log \big(1 - p_{\mathbf{S}}(u_-, v_-)\big) \right) + \sum_{n=1}^{N} \|\mathbf{P}_n\|_2 \tag{5.12}$$

We impose group Lasso (each $\mathbf{P}_n$ is a group) regularization to avoid over-fitting and incentivize sparse projectors. By combining objectives section 5.4.4, eq. (5.11), eq. (5.12) we can re-obtain eq. (5.3) with minor modifications. Each module is trained alternately holding the other two constant via *ADAM* gradient updates [87].

Computational Complexity: On the whole, our model complexity is $O(\mathbf{G}) + O(|\mathcal{N}| \times d_{\mathbf{S}}) + O(\epsilon \times N \times d_w)$, where $O(\mathbf{G})$ is the recommender complexity, $|\mathcal{N}|$ is the social link count, N is the number of Hadamard projectors, $d_{\mathbf{S}}$ the social space dimensionality, d_w the user context feature dimensionality and ϵ is the *fake/true-pair* sample count. In practice, our modules are highly parellel and the discriminator model $\mathbf{D}$ is implemented with sparse optimizations [89].

In practice, the discriminator $\mathbf{D}$ and pair weight module $w(u, v)$ add 50% overhead to train

Auto-Encoder based recommenders [115, 224] (less percentage overhead for more complex recommender architectures), if the dimensions of $\mathbf{S}$, $\mathbf{X}$ are equal, i.e., $d_{\mathbf{S}} = d_{\mathbf{X}}$. The overheads are reduced further if $d_{\mathbf{S}} < d_{\mathbf{X}}$.

5.5 EXPERIMENTAL RESULTS

In this section, we present extensive quantitative and qualitative analyses of our model. We begin by introducing datasets and baseline methods in Section 6.6.1, followed by the primary recommendation task in Section 6.6.6, and quantitative results by integrating three diverse neural recommenders in our framework (Table 5.2). Then in Section 5.5.3, we analyze the user segments where our model exhibits gains and study the pair samples that were important in the model's learning process in Section 5.5.4. In Section 5.5.5 we examine the empirical results and inferences to two important questions: Q1—*What is the effect of adversary weight λ on interest space collapse and does this depend on the generator architecture?* and Q2—*Is adversarial training robust to missing social or item history user data?* Finally, we analyze parameter sensitivity in Section 5.5.6 and discuss limitations in Section 6.6.9.

5.5.1 Datasets and Baselines

We evaluated all models over five publicly available datasets, *Delicious*, *Ciao*, *Epinions*, *Ask-Ubuntu* and *Yelp*.

Ciao[1]: The Ciao dataset contains user's ratings on DVDs, the user social network, and DVD category data.

Epinions[1]: The Epinions dataset provides user ratings to purchased items, the user social network, and item categories.

Ask-Ubuntu[2]: Ask-Ubuntu is a popular online Q&A forum. We predict tags for users' posts. Social links are interactions between users via comments, answers, or edits.

Delicious[3]: The Delicious dataset contains user bookmarks, social links, and tags. We predict bookmarks in our experiments.

Yelp[4]: The Yelp dataset contains user ratings to restaurants and their social network.

We pre-process smaller datasets (*Ciao, Epinions, Delicious*) to retain users and items with ten or move reviews. For *Ask-Ubuntu* and *Yelp*, we set the threshold to 30. We compare

[1]https://www.cse.msu.edu/ tangjili/datasetcode/truststudy.htm
[2]https://archive.org/details/stackexchange
[3]https://grouplens.org/datasets/hetrec-2011/
[4]https://www.yelp.com/dataset/challenge

our framework against recent state-of-the-art baselines. We present gains by integrating the three strongest social-agnostic recommender baselines as the generators in our framework.

BPR [166]: BPR is a first-cut baseline for all implicit feedback recommendation methods.

SBPR [239]: SBPR augments personalized ranking by assuming users assign higher ranks to their friends' preferences.

NCF [60]: NCF is a state-of-the-art neural ranking model combining matrix factorization and neural representation learning. NCF outperforms most conventional baselines.

SNCF: We modify NCF by concatenating social network embedding representations (as in [198]) in the neural inputs. We refer to this variant as Social NCF (SNCF).

Social-GCN [222]: Social-GCN convolves user neighbor features and optimizes a personalized ranking objective function.

SEREC [213]: SEREC assumes users are exposed to items reviewed by their contacts, some leading to purchases. SEREC is competitive on most datasets due to its flexible item choices.

CB [223]: Contextual-Bandit (CB) uses dual graph-attention networks to compute user interest and social embeddings and selects one of the factors to explain each purchase.

DAE [224]: Denoising Auto-Encoders learn a low-dimensional user interest representation by decoding a noised version of his item history. We incorporate DAE as an adversarial variant.

VAE-CF [115]: Variational Auto-Encoders eliminate noisy inputs by introducing stochasticity in the user interest space. We incorporate VAE and evaluate its gains in our framework.

LRML [198]: LRML is a memory network architecture to learn relation vectors between user-item pairs. We incorporate LRML in our adversarial framework.

We tested our framework by incorporating **DAE**, **VAE-CF** and **LRML** as generators **G** in our framework. We refer to these variants as **Asr-DAE**, **Asr-VAE** and **Asr-LRML** (**Asr** denotes adversarial social regularization). Experiments were performed on a Nvidia Tesla V100 GPU with *TensorFlow* implementations on the Linux platform. Our implementations are publicly available[5].

5.5.2 Social Recommendation Task

To evaluate the performance of the recommender models listed above, we compute the $NDCG@K$ ($N@K$) and $Recall@K$ ($R@K$) metrics [114]. $Recall@K$ is a measure of the percentage of relevant items in the top-K recommendations to each user; it considers true and false positives in the list and is thus more descriptive than Precision. The $NDCG@K$ metric

[5] `https://github.com/CrowdDynamicsLab/Adversarial-Social-Recommendation`

Table 5.1: Aggregate recommendation result table for the three smaller datasets. $R@K$ and $N@K$ denote the Recall and NDCG metrics for all models. Our models outperform competing baselines by upto 35% Recall@50 and 25% NDCG@50. Asr-VAE was found to be the best overall model. Owing to the small inventory size for the *Delicious* dataset, we only report the *@20* metrics.

	Smaller Datasets									
Method	**Epinions**				**Ciao**				**Delicious**	
	R@20	*N@20*	*R@50*	*N@50*	*R@20*	*N@20*	*R@50*	*N@50*	*R@20*	*N@20*
Social-Agnostic Recommenders										
BPR [166]	0.264	0.141	0.440	0.176	0.232	0.128	0.428	0.162	0.363	0.271
NCF [60]	0.310	0.138	0.462	0.181	0.282	0.147	0.471	0.193	0.498	0.283
DAE [224]	0.324	0.164	0.498	0.198	0.290	0.143	0.493	0.191	0.572	0.340
VAE-CF [115]	0.336	0.161	0.510	0.204	0.299	0.152	0.496	0.197	0.585	0.327
LRML [198]	0.329	0.173	0.509	0.219	0.317	**0.165**	0.526	0.206	0.482	0.310
Social Recommenders										
SBPR [239]	0.271	0.138	0.446	0.185	0.217	0.140	0.439	0.174	0.381	0.292
SNCF	0.306	0.189	0.468	0.202	0.284	0.151	0.478	0.196	0.520	0.296
SGCN [222]	0.318	0.153	0.481	0.198	0.275	0.142	0.470	0.179	0.546	0.295
CB [223]	0.337	0.171	0.436	0.202	0.288	0.153	0.491	0.180	0.572	0.287
SEREC [213]	0.348	0.167	0.496	0.213	0.303	0.158	0.513	0.202	0.589	0.314
Adversarial Social Recommenders (Ours)										
Asr-DAE	0.339	0.168	0.513	0.207	0.301	0.144	0.519	0.189	0.603	0.322
Asr-VAE	**0.358**	0.173	0.532	0.216	0.312	0.138	0.528	0.196	**0.617**	**0.379**
Asr-LRML	0.340	0.166	0.527	0.220	**0.328**	0.160	**0.544**	**0.214**	0.495	0.357

* The Asr variants denote the DAE, VAE-CF, and LRML base models integrated as the generator in our adversarial framework. Our model can substitute recommender (generator) and discriminator architectures owing to the modular formulation. The performance numbers in bold numerals indicate statistically significant gains over the second-best model at $p = 0.05$. When there are two or more strong performers under a specific metric, we underline them. Our adversarial variants exhibit strong gains over competing social recommenders as well as their respective base models.

Table 5.2: Aggregate recommendation result table for the two larger datasets. $R@K$ and $N@K$ denote the Recall and NDCG metrics for all models. Our models outperform competing baselines by upto 35% Recall@50 and 25% NDCG@50. Asr-VAE was found to be the best overall model.

Method	Larger Datasets							
	Ask-Ubuntu				**Yelp**			
	R@20	*N@20*	*R@50*	*N@50*	*R@20*	*N@20*	*R@50*	*N@50*
Social-Agnostic Recommenders								
BPR [166]	0.377	0.199	0.514	0.264	0.228	0.125	0.431	0.170
NCF [60]	0.420	0.215	0.538	0.281	0.196	0.118	0.488	0.209
DAE [224]	0.416	0.301	0.569	0.392	0.270	0.158	0.473	0.213
VAE [115]	0.408	0.317	0.576	0.383	0.281	0.164	0.479	0.208
LRML [198]	0.405	0.366	0.564	0.405	0.272	0.160	0.483	0.196
Social Recommenders								
SBPR [239]	0.368	0.206	0.528	0.287	0.230	0.143	0.449	0.196
SNCF	0.414	0.371	0.541	0.403	0.198	0.103	0.493	0.202
SGCN [222]	0.397	0.343	0.526	0.395	0.288	0.160	0.492	0.176
CB [223]	0.399	0.365	0.559	0.382	0.282	0.154	0.471	0.196
SEREC [213]	0.415	0.362	0.584	0.414	<u>0.306</u>	<u>0.173</u>	**0.508**	0.211
Adversarial Social Recommenders (Ours)								
Asr-DAE	<u>0.434</u>	0.347	<u>0.585</u>	0.412	0.272	0.158	0.489	0.201
Asr-VAE	<u>0.431</u>	0.350	<u>0.592</u>	0.401	<u>0.298</u>	0.161	0.496	0.218
Asr-LRML	0.411	**0.375**	0.578	**0.419**	0.287	<u>0.172</u>	0.481	**0.233**

* The Asr variants denote the DAE, VAE-CF, and LRML base models integrated as the generator in our adversarial framework. Our model can substitute recommender (generator) and discriminator architectures owing to the modular formulation. The performance numbers in bold numerals indicate statistically significant gains over the second-best model at $p = 0.05$. When there are two or more strong performers under a specific metric, we underline them. Our adversarial variants exhibit strong gains over competing social recommenders as well as their respective base models.

is position sensitive and considers the order of the ranked list against the ideal case (only relevant items placed at the top). We evaluate each ranked list at $K = 20, 50$ (Table 5.2).

We randomly split each dataset into Training (80%), Validation (10%), and Test (10%). We tune the baselines with parameter ranges centered at the author-provided values to obtain the best performance on our datasets. For a fair comparison, we set the representation dimensions to 128 for all models. For our model, adversary weight λ, balance μ were both tuned in the range $(0, 10]$ and we set Hadamard projectors $N = 10$ across all experiments.

Comparative Analysis : We make several observations from the experimental results obtained with the baseline recommenders and our adversarial variants (Table 5.2). First, conventional social recommenders are outperformed by social-agnostic neural methods that efficiently leverage the rating information. Non-linear transformations of interest representations are more expressive than linear or bi-linear operations [60].

Second, expressive interest spaces (like in **DAE** [224]) benefit more from social regularization than conventional interest representations. The gains achieved by integrating neural models in our framework are stronger than those adding social information to older methods (e.g., *R@50* gains of **SBPR** vs. **BPR** are smaller on average than those of **Asr-VAE** vs. **VAE**). Also, note that a direct integration of pre-trained embeddings (as in **SNCF**) does not produce a noticeable gain in performance. Pre-trained graph embeddings cannot contextually distinguish the influence of a user's neighbors by their interests.

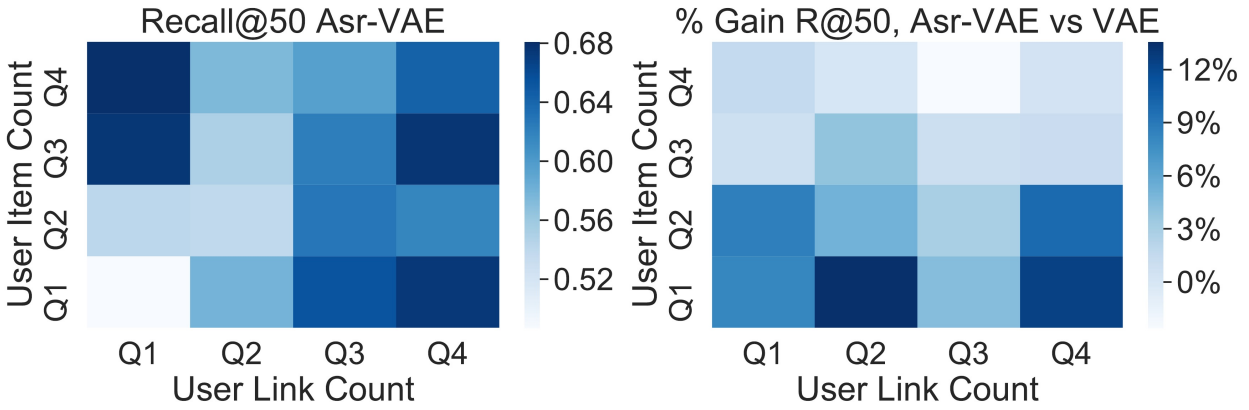
We find our adversarial variants and **SEREC** to outperform older social recommender baselines by significant margins. While **SEREC** permits for the exposed item set to be prioritized differently, **CB** [223] flexibly attributes purchases, however picking a single factor (interest vs. social) instead of a contextual combination. **Asr-VAE** was found to achieve the best overall performance. The **VAE** user representations are inherently stochastic unlike **DAE** and **LRML**, we also observed greater recommendation diversity (less interest space collapse) with **Asr-VAE** (Section 5.5.5).

Neighbor Diversity: Unlike exposure models, we condition each item on the specific social context, i.e., a phone exposed by an android expert has a greater effect than from other social contacts. To verify this, we measure the diversity of each user's friends. As an example, if a user's four friends have 5, 10, 15, and 20 items, their item distribution is $(5/50, 10/50, 15/50, 20/50)$. We estimate the KL-divergence of this distribution against the uniform case to measure diversity. We then split all users into four quartiles based on their neighbor diversity (Q4 has users with high neighbor diversity) and compare *R@50* relative gains of **Asr-VAE** over **SEREC** on samples from each quartile. Ideally, we expect our model to make gains on later quartiles since context is more important to distinguish diverse social contacts.

Table 5.3: Performace gains of **Asr-VAE** against **SEREC** on user neighbor diversity. We see stronger gains for quartile Q4 (high neighbor item-count diversity).

Neighbor Diversity Quartile	**Q1**	**Q2**	**Q3**	**Q4**
% Gain R@50 (Asr-VAE vs. SEREC)	3.82%	3.16%	3.45%	4.23%

Figure 5.3: Overall Performance and Percentage Gains of Asr-VAE (by R@50), measured across social link count and item count user quartiles (Q1 = lowest values, Q4 = highest values). Heatmap values are averaged over the smaller datasets (*Ciao*, *Epinions*, *Delicious*).



(a) Asr-VAE *Recall*@50

(b) % Gain Asr-VAE vs. VAE *R*@50

5.5.3 Interpreting our Results

We now study our results more closely to understand the source of **Asr-VAE**'s gains over base recommender **VAE**. We observe the *R@50* performance values of **Asr-VAE** against the base recommender **VAE** to observe the source of our gains. We analyze users along three axes -

Item Count Quartile: We separate the test users into four quartiles based on the number of items in their histories.

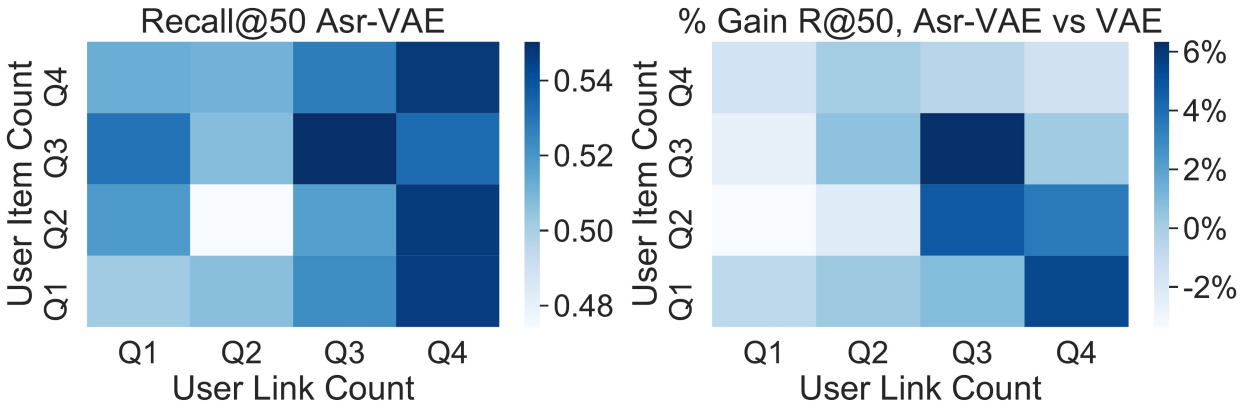
Social Links Quartile: We again separate test users into four quartiles depending on their social link counts.

User Coherence Quartile: We define user coherence as the mean pair-wise correlation of item categories purchased by the user. Thus, if a user were to purchase items that are often bought together, he receives greater coherence. We partition test users into four quartiles by coherence scores. We can compute coherence only for the *Epinions* and *Ciao* datasets.

We first study the overall performance variations and performance gains for users in different social and item count quartiles.

Overall Results: The heatmaps on the left in Figure 5.3, Figure 5.4 indicate the per-

Figure 5.4: Overall Performance and Percentage Gains of Asr-VAE (R@50), mean over larger datasets (*Ask-Ubuntu*, *Yelp*).



(a) Asr-VAE *Recall*@50

(b) % Gain Asr-VAE vs. VAE *R*@50

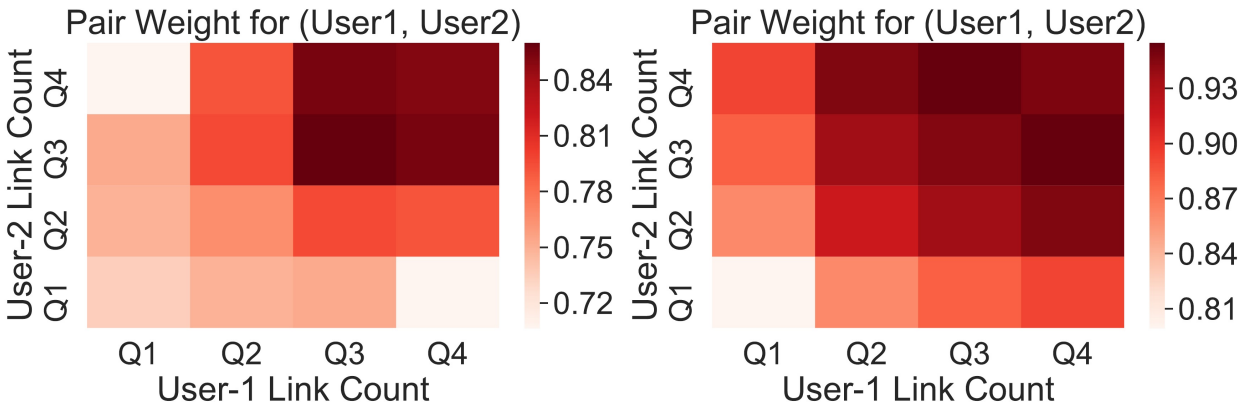
formance (*R@50*) achieved by Asr-VAE for users grouped under each quartile (Q1 - Lower values), averaged over the smaller and larger datasets respectively. We observe weaker performance for users at the bottom-left of the plot (i.e., users with sparse links and items). For the small datasets, stronger results appear at the other three corners (i.e., users who have either have a long item history, or several social connections). On the large datasets, results are concentrated towards users with greater link counts (social link quartiles Q3 and Q4). These gains are consistent with our intuitions, users with a large item history obtain accurate interest representations while those with more social links can socially regularize their interest embeddings.

Difference between Asr-VAE and VAE-CF: The heatmaps on the right of each figure (Figure 5.3, Figure 5.4) indicate the relative performance gains of **Asr-VAE** against its base recommender **VAE** for users in the respective quartiles.

Dissecting Performance Gains: We observe stronger performance gains in terms of item recall in the bottom half of each heatmap (Figure 5.4 and Figure 5.3), indicating improvements for users in the 25% and 50% user item count quartiles (i.e., users with sparse interest representations).

Social Regularization: Social regularization especially benefits users with limited purchase histories by padding their interest representations with the interest representations of other users in their extended ego networks. Surprisingly, we also see gains in the bottom left corner for the smaller datasets. A likely reason for this observation is that users in these quartiles have fewer informative social links (since the datasets are smaller and lack sufficient peer-to-peer social links), thus achieving modest performance gains in **Asr-VAE** vs. **VAE**.

Figure 5.5: We measure the Pair-Weight allocations to sampled pairs of users by our weight module. The x and y-axis denote the social link count quartiles of each user in pair (User-1, User-2), Q1 contains the lower values. E.g., The top-right box of the heatmap is the average weight alloted to samples where both users have many social links (Q4, Q4).



(a) Smaller Datasets (*Ciao, Epinions, Del*)

(b) Large Datasets (*Ask-Ubuntu, Yelp*)

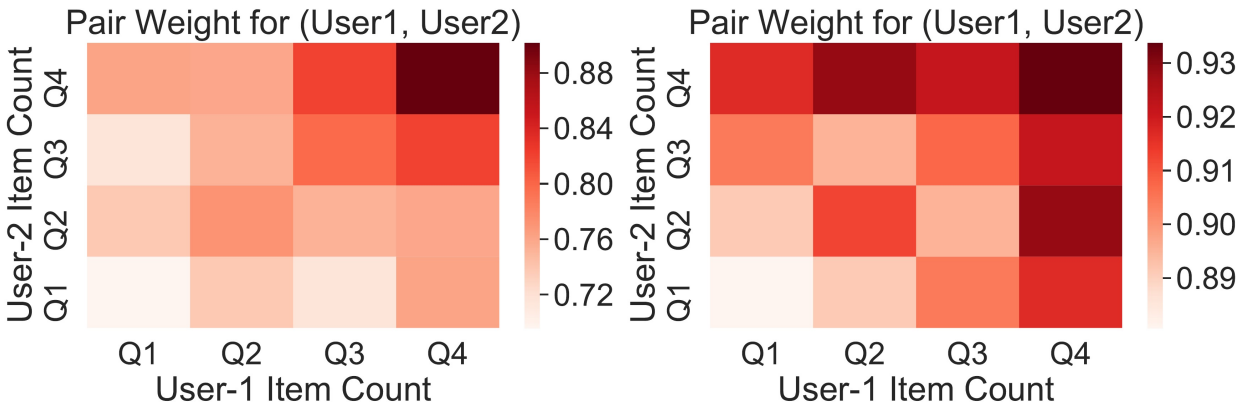
5.5.4 Pair-Weight Allocations

The Hadamard projection vectors in our weighting function $w(u, v)$ are hard to interpret, since we do not know what each latent dimension is, however pair weights assigned to pair samples can be aggregated to analyze the training process.

We observe from Figure 5.5, Figure 5.6 that our model prioritizes pairs of users where both users have numerous social connections or longer item histories to regularize their neighborhoods. Intuitively, pair samples where both users are influencers or prolific consumers are likely to regularize their social and interest neighborhoods (they may act as cluster centers). We observe a similar trend against user coherences in the *Ciao* dataset (Figure 5.7). In *epinions*, the model also prioritizes quartiles where one user in a pair has more coherent purchases than the other (note that we can only compute coherence for the *Ciao, Epinions* datasets using their item category labels).

Finally, we also analyze pair weights by considering differences within user pairs. We look at the difference in the number of social counts and length of item histories of the two users. Figure 5.8 indicates a slight drop in pair weights at the extreme right of each plot (significant difference in social link count). Such connections are unlikely to represent friend-friend links and hence may not effectively regularize preferences. However, in *Yelp*, *Ask-Ubuntu* we observe a more uniform distribution of pair weights, potentially due to users' information-seeking requirements these websites.

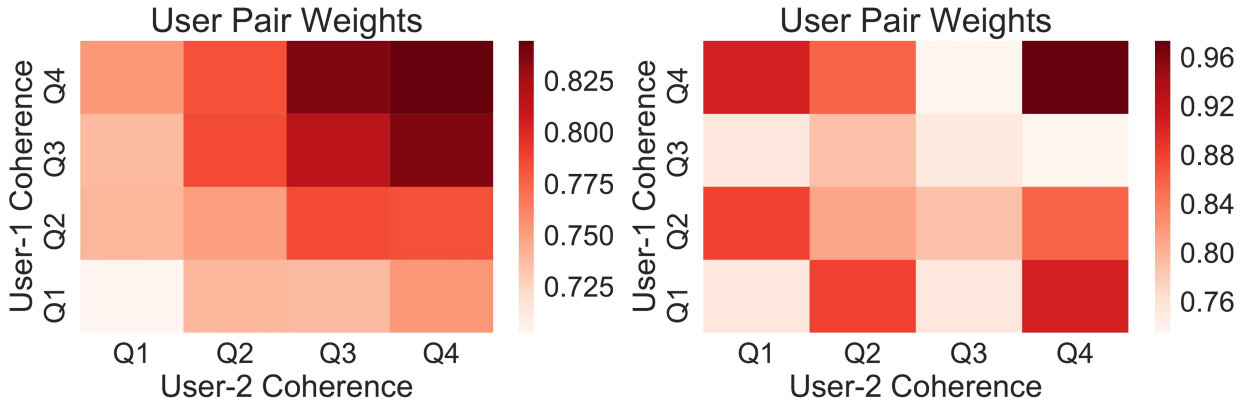
Figure 5.6: We create these heatmaps similar to Figure 5.5 with user item count quartiles, i.e., Q4 denotes long item histories.



(a) Small Datasets (*Ciao, Epinions, Del*)

(b) Large Datasets (*Ask-Ubuntu, Yelp*)

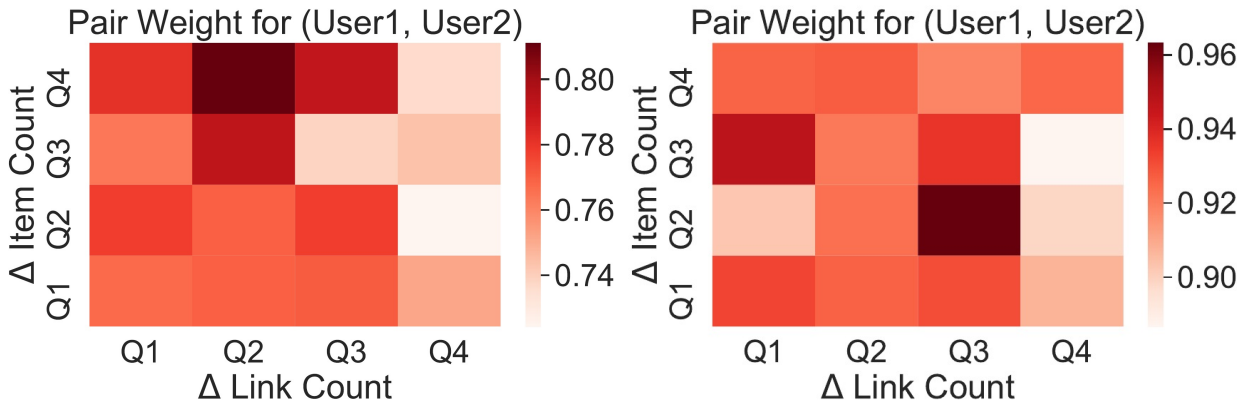
Figure 5.7: Pair weights against user coherence for pair samples in the *Ciao* and *Epinions* datasets.



(a) *Ciao* dataset

(b) *Epinions* dataset

Figure 5.8: Each pair sample (User 1, User 2) is binned in quartiles by item and social link count differences between the two users. We then plot the average pair weights assigned to the pair samples within the respective quantiles.



(a) Small Datasets (*Ciao, Epinions, Del*) (b) Large Datasets (*Ask-Ubuntu, Yelp*)

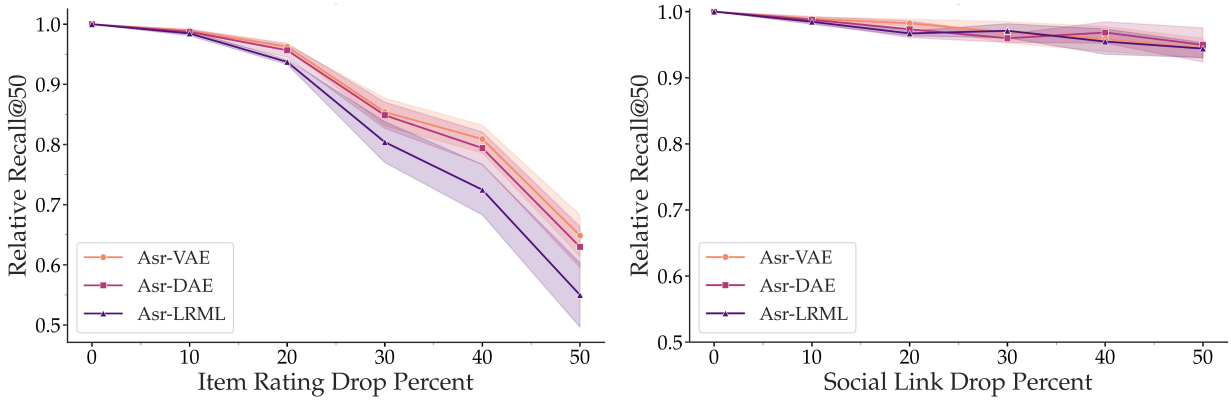
5.5.5 Robustness and Interest Space Collapse

We study the robustness of each adversarial model to lossy data by separately sub-sampling the social links and item ratings of each user in the respective training sets (Figure 5.9). Performance is measured as a fraction of the peak performance (e.g., 0.98 indicates the model degraded by 2%). We observe an average performance degradation $\leq 3\%$ by *R@50* with 10% item ratings dropped and $\leq 6\%$ at 20% drop, indicating our models are reasonably robust to lossy item ratings. Asr-LRML shows a slightly steeper drop compared to the auto-encoder variants. Further, we observe our models are highly robust to social link drop, degrading by 5% *R@50* even with 50% social links dropped, owing to their stochastic pair sample-based gradient updates.

We also analyze the effect of adversary weight λ on the diversity of items recommended to users (Figure 5.10). Specifically, we apply k-means clustering to the GAE [88] embeddings for each social network, pick the median user cluster by average degree, and measure recommendation diversity as the union of their top-50 recommendations. λ_{opt} indicates the optimal λ setting by *R@50* for each dataset. As λ is varied, the variation in diversity is measured as a percentage of the largest union set obtained (i.e., less diversity implies a smaller union set and hence, a lower percentage).

In general, larger values of λ result in less diverse recommendations. **Asr-VAE**'s recommendations are slightly more diverse at greater values of λ owing to the stochasticity of the user representations in the **VAE** generator. On the opposite end, smaller multiples of λ also produce lower recommendation diversity by over-fitting to the supervised loss term $\mathcal{O}_{\mathbf{G}}$ in

Figure 5.9: We observe $\leq 6\%$ *R@50* degradation at 20% item drop indicating our models are fairly robust in practice. Dropping social links results in much smaller performance drops, indicating the effectiveness of stochastic user pair sampling. Performance values are averaged across datasets.



(a) Performance with Item Drop

(b) And Social Link Drop

Figure 5.10: Recommendation diversity is observed to drop on either side of λ_{opt}, but due to different causes. The smaller λ multiples result in overfitting to the supervised term $\mathcal{O}_{\mathbf{G}}$, while larger multiple result in interest space collapse, i.e., less diverse recommendations to socially clustered users. Diversity values are averaged across datasets.

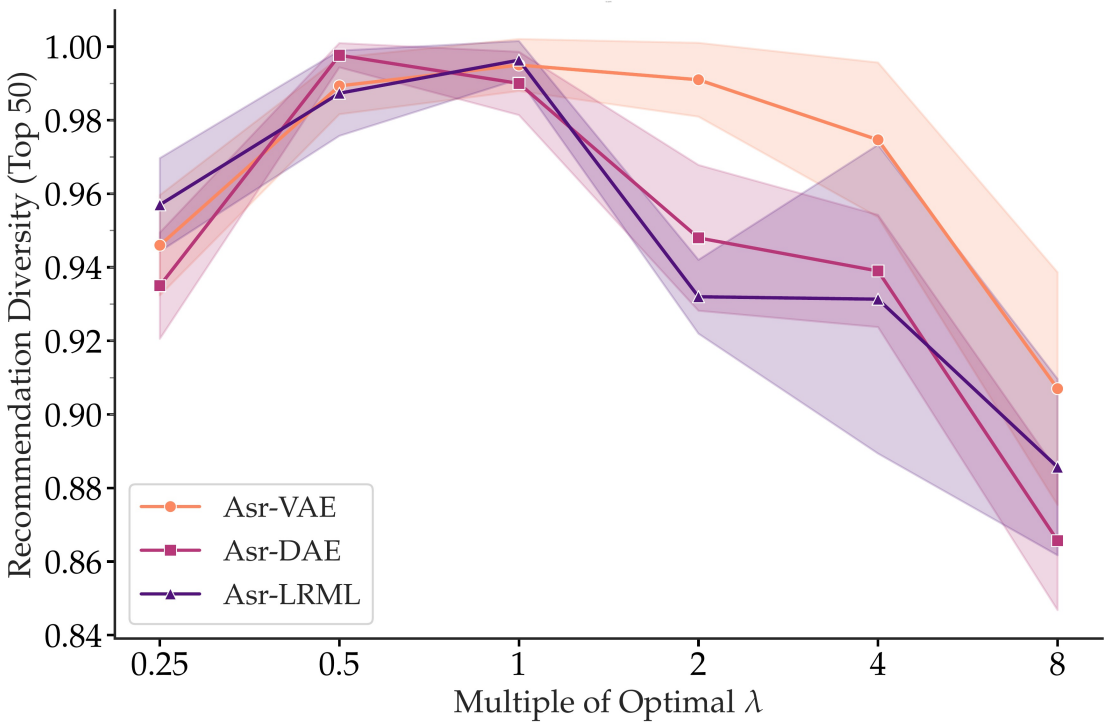
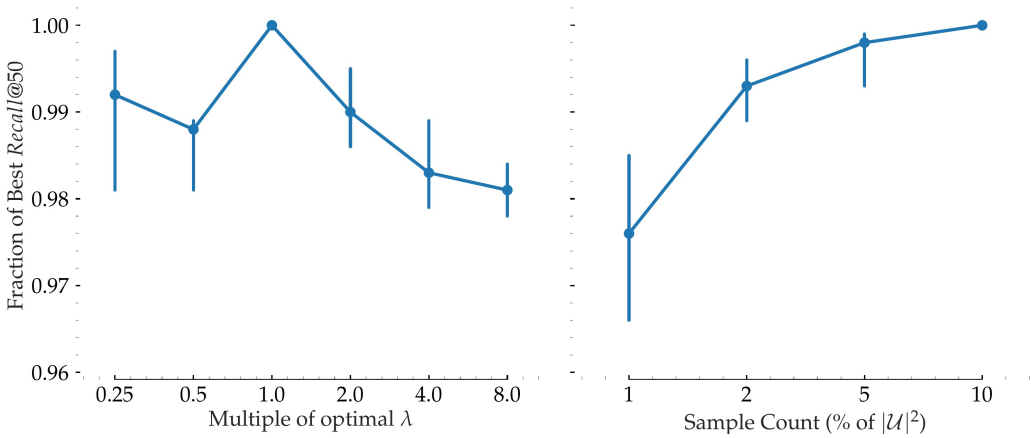


Figure 5.11: Asr-VAE is fairly robust in a wide range of values ($\leq 2.5\%$ *R@50* variation). λ is varied as multiples of the best performing value λ_{opt}, larger multiples result in a performance drop. Robust performance is obtained for user pair sample count $\epsilon = 0.02|\mathcal{U}|^2$, further samples provide small gains ($\leq 1\%$). *R@50* values are averaged across datasets.



the generator objective in Equation (5.10). Values close to λ_{opt} produce the most diverse set of top-50 recommendations.

5.5.6 Parameter Sensitivity

In this section, we study the sensitivity of our model to two key parameter values, first the adversary weight λ, and second the user pair sample count ϵ (Figure 5.11) measured as a fraction of the total number of unique user pairs (e.g., 5% denotes $0.05 \times \mathcal{U}^2$).

Varying the adversary weight λ results in a performance drop on either side of the optimal value. We find $\epsilon = 0.02 \times \mathcal{U}^2$ to provide an efficient tradeoff between compute-cost and performance. In practice, ϵ does not significantly change the overall compute time since the pair weight module is inexpensive. Also note that λ_{opt} varies across datasets, with values on either side of λ_{opt} resulting in weaker and less diverse recommendations (Figure 5.10).

5.5.7 Limitations

We identify a few key limitations of our model. First, although the model performance is stable around multiples of the optimal values of λ_{opt}, the optimal weight varies across datasets and applications. The optimal strength of social regularization depends on the data semantics as well the generator and discriminator architectures. Second, the pair-weighting

strategy performs best when the provided context features are meaningfully correlated to the interests and social indicators of users. Thus, depending on the application, context features should be picked to enhance social inference and prevent diversity loss in the generated recommendations.

5.6 CONCLUSION AND NEXT STEPS

5.6.1 Chapter Summary

In this chapter, we formulate the multimodal representation learning task as an adversarial attribution problem. The data modalities compete to accurately represent each user's preferences or item characteristics towards a recommendation or inference task. We leverage the widespread social recommendation problem to demonstrate the utility of the proposed framework. Unlike prior work, we develop a modular architecture-agnostic framework that enables us to address a broad range of multimodal recommendation applications, the corresponding user and item data-modalities, and a wide range of gradient updated models to represent each data modality.

Further, we show that a direct application of metric-learning approaches or equivalent formulations may result in generator / user preference space collapse owing to the strong pairwise correlation constraints across the two representation spaces. Instead, we propose a stochastic pair-weighting approach that allows us to assess each user independently and enhance the user interest representation via contextual integration of their social structure. Extensive experimental results over five real-world datasets reveal the strengths of our approach.

5.6.2 Improvements to the Proposed Framework

When training adversarial models, not all samples are equally important for the generator and discriminator updates; At every stage of the training process, most data points can be safely ignored without significant changes to the parameter trajectories. For instance, the variance heuristic or importance sampling [84] discards points to reduce the variance of the gradient estimates and enable smooth convergence.

Our framework aims to enable consistent and smooth generator updates to improve the quality of recommendations made to users; In this context, variance reduction strategies can enhance gradient updates' smoothness independent of the precise generator / recommender

architecture. Thus, we identify multiple promising avenues to improve performance - developing sampling strategies to identify and reweight informative fake-pairs to regularize the interest space, either by enhancing contextual weighting with a non-linear combination of the context projections or by developing efficient and expressive discriminator architectures tailored to handle specific classes of recommender models.

5.6.3 From Knowledge Extraction to Knowledge Transfer

The three chapters, Chapter 3, Chapter 4, and Chapter 5, in unison, describe strategies to tackle skewed and sparse data towards multimodal recommendation and inference tasks. Learning more informative embedding spaces improves the trained models' sample efficiency and enables more accurate inferences for users with limited data. The proposed methods generalize across diverse data modalities and the corresponding knowledge representation models. Thus, we can summarize our work until this point: frameworks to represent long-tail users and items towards supervised and unsupervised learning objectives with unimodal or multimodal data. In other words, we developed strategies to extract task-dependent knowledge with limited user-item interaction histories despite long-tail data challenges.

In the following two chapters (Chapter 6 and Chapter 7), we focus on augmenting and supplementing the proposed knowledge extraction strategies via knowledge transfer across multiple correlated recommendation and inference tasks and knowledge transfer from alternate recommendation domains incorporating similar recommendation tasks (even with disjoint sets of entities). We can simultaneously apply enhanced knowledge extraction strategies and knowledge transfer strategies to improve sample efficiency in machine learning problems incorporating sparse and skewed training data.

Specifically, we now refer to the recommendation domain definition in Section 1.2.2. Numerous challenges arise in the multi-domain setting, such as geographic disparities in data quality and volume, where recommendation domains (geographic regions in the example) do not explicitly share users or items that permit cross-domain inference [98]. When an entire domain (or geographic region in this case) lacks data, grouping mechanisms are insufficient. Furthermore, grouping mechanisms assume sufficient data for a subset of the population to define groups and hence, do not directly apply to a few-shot / cold-start scenario [101, 192].

In the next chapter, we extend our overall sparsity and skew mitigation strategy to the multi-domain setting via interaction context (Section 1.2.1). Unlike user or item data-modalities, interaction context is specific to each user-item interaction. Interaction context may vary across two data-points even if they involve the same interacting user and item. We describe a generalizable solution that extends our grouping strategy in the cross-domain

scenario. We develop behavioral invariants for users via pooled contextual combinations representing users' and items' preferred interaction strategies. These invariants, once learned, can be used to make few-shot inferences about users in a sparse-domain by pooling them with similar users in the dense/source-domain (and likewise for items).

The proposed strategy extends the utility of the prior techniques to the cross-domain setting in the following manner: We learn user and item representations in the source domain and correlate them with their preferred contextual invariants. Subsequently, we can leverage the context data for target domain users and items to link them to the source domain representations. In this manner, we additionally enable significant scalability gains since few-shot inferences are computationally inexpensive compared to fitting new models to the target-domains.

CHAPTER 6: LEARNING CONTEXTUAL INVARIANTS FOR CROSS-DOMAIN RECOMMENDATION AND INFERENCE

The rapid proliferation of new users and items on the social web has aggravated the gray-sheep user/long-tail item challenge in recommender systems. Historically, cross-domain co-clustering methods have successfully leveraged shared users and items across dense and sparse domains to improve inference quality. However, they rely on shared rating data and cannot scale to multiple sparse target domains (i.e., the one-to-many transfer setting).

The need to scale to several target domains without shared users or items, combined with the increasing adoption of neural recommender architectures, motivates us to develop scalable neural layer-transfer approaches for cross-domain transfer learning. Our key intuition is to guide neural collaborative filtering with domain-invariant components shared across the dense and sparse domains, improving the user and item representations learned in the sparse domains. We leverage contextual invariances across domains to develop these shared modules and demonstrate that we can *learn-to-learn* informative representation spaces even with sparse interaction data using user-item interaction context. We show our approach's effectiveness and scalability on two public datasets and a massive transaction dataset from Visa, a global payments technology company (19% Item Recall, 3x faster vs. training separate models for each domain). Our approach is applicable to both *implicit* and *explicit* feedback settings.

6.1 INTRODUCTION

This chapter's focus is to learn to build expressive neural collaborative representations of users and items with sparse interaction data. The problem is essential: neural recommender systems are crucial to suggest useful products, services, and content to users online. Sparsity, or the long tail of user interaction, remains a central challenge to traditional collaborative filtering, as well as new neural collaborative filtering (NCF) approaches [60]. Sparsity challenges have become pronounced in neural models [95] owing to generalization and overfitting challenges, motivating us to *learn-to-learn* effective embedding spaces in such a scenario.

Cross-domain transfer learning is a well-studied paradigm to address sparsity in recommendation tasks. However, how recommendation domains are defined plays a key role in deciding the algorithmic challenges. In the most common pairwise cross-domain setting, we can employ cross-domain co-clustering via shared users or items [132, 218], latent structure alignment [48], or hybrid approaches using both [67, 152]. However, recommendation domains with limited user-item overlap are pervasive in real-world applications, such as

geographic regions with disparities in data quality and volume (e.g., restaurant recommendation in cities vs. sparse towns). Historically, there is limited work towards such a *few-dense-source*, *multiple-sparse-target* setting, where entity overlap approaches are ineffective. Further, sharing user data entails privacy concerns [47].

Simultaneously, context-aware recommendation has become an effective alternative to traditional methods owing to the extensive multi-modal feedback from online users [137]. Combinations of contextual predicates prove critical in *learning-to-organize* the user and item latent spaces in recommendation settings. For instance, an *Italian wine restaurant* is a good recommendation for a *high spending* user on a *weekend evening*. However, it is a poor choice for a *Monday afternoon*, when the user is at work. The intersection of restaurant type (an attribute), historical patterns (historical context), and interaction time (interaction context) jointly describe the likelihood of this interaction.

Our key intuition is to infer such *behavioral invariants* from a *dense-source* domain (where we have ample interaction histories of users with wine restaurants) and apply or adapt these learned invariants to improve inference in *sparse-target* domains. Clustering users who interact under covariant combinations of contextual predicates in different domains lets us better incorporate their behavioral similarities and analogously for the item sets. The user and item representations in sparse domains can be significantly improved when we combine these transferrable covariances.

Guiding neural representations is also a central theme in gradient-based meta-learning. Recent work [45, 112] measures the plasticity of a base-learner via gradient feedback for few-shot adaptation to multiple semantically similar tasks. However, the base-learner is often constrained to simpler architectures (such as shallow neural networks) to prevent overfitting [192] and requires multi-task gradient feedback at training time [45]. This strategy does not scale to the embedding learning problem in NCF, especially in the *many sparse-target* setting.

Instead, we propose incorporating the core strengths of meta-learning and transfer learning by defining transferrable neural layers (or meta-layers) via contextual predicates, working in tandem with and guiding domain-specific representations. Further, we develop a novel adaptation approach via regularized residual learning to incorporate new target domains with minimal overheads. Only residual layers and user/item embeddings are learned in each domain while transferring meta-layers, limiting sparse domain overfit. In summary, we make the following contributions:

Contextual Invariants for Disjoint Domains: We identify the shared task of *learning-to-learn* NCF embeddings via cross-domain contextual invariances. We develop a novel class of pooled contextual predicates to learn descriptive representations in sparse recommenda-

tion domains without sharing users or items.

Tackling the *One-Dense, Many-Sparse* Scenario: Our model infers invariant contextual associations via user-item interactions in the dense source domain. Unlike gradient-based meta-learning, we do not sample all domains at train time. We show that it suffices to transfer the source layers to new target domains with an inexpensive and effective residual adaptation strategy.

Modular Architecture for Reuse: Contextual invariants describing user-item interactions are geographically and temporally invariant. Thus we can reuse our meta-layers while only updating the user and item spaces with new data, unlike black-box gradient strategies [45]. This also lets us embed new users and items without retraining the model from scratch.

Strong Experimental Results: We demonstrate strong experimental results with transfer between dense and sparse recommendation domains in three different datasets - (Yelp Challenge Dataset[1], Google Local Reviews[2]) for benchmarking purposes and a large financial transaction dataset from Visa, a major global payments technology company.

We demonstrate performance and scalability gains on multiple sparse target regions with low interaction volumes and densities by leveraging a single dense source region.

We now summarize related work, formalize our problem, describe our approach, and evaluate the proposed framework.

6.2 RELATED WORK

We briefly summarize a few related lines of work that apply to the sparse inference problem in recommendation:

Sparsity-Aware Cross-Domain Transfer: Structure transfer methods regularize the user and item subspaces via principal components [107, 150], joint factorization [79, 118], shared and domain-specific cluster structure [48, 152] or combining prediction tasks [97, 179] to map user-item preference manifolds. They explicitly map correlated cluster structures in the subspaces. Instead, co-clustering methods use user or item overlaps as anchors for sparse domain inference [132, 218], or auxiliary data [77, 208] or both [67]. It is hard to quantify the volume of users/items or shared content for effective transfer. Further, both overlap-based methods and pairwise structure transfer do not scale to *many sparse-targets*.

Neural Layer Adaptation: A wide-array of layer-transfer and adaptation techniques use convolutional invariants on semantically related images [125, 191] and graphs [177].

[1] `https://www.yelp.com/dataset/challenge`

[2] `http://cseweb.ucsd.edu/~jmcauley/datasets.html`

However, unlike convolutional nets, latent collaborative representations are neither interpretable nor permutation invariant [60, 115]. Thus it is much harder to establish principled layer-transfer methods for recommendation. We develop our model architecture via novel contextual invariants to enable cross-domain layer transfer and adaptation.

Meta-Learning in Recommendation: Prior work has considered algorithm selection [26], hyper-parameter initialization [40, 44], shared scoring functions across users [203] and meta-curriculums to train models on related tasks [40, 101]. Across these threads, the primary challenge is scalability in the multi-domain setting. Although generalizable, they train separate models (over users in [203]), which can be avoided by adapting or sharing relevant components.

6.3 PROBLEM DEFINITION

Consider recommendation domains $\mathbb{D} = \{\mathbf{D}_i\}$ where each $\mathbf{D}_i$ is a tuple $\{\mathcal{U}_{\mathbf{D}_i}, \mathcal{V}_{\mathbf{D}_i}, \mathcal{T}_{\mathbf{D}_i}\}$, with $\mathcal{U}_{\mathbf{D}_i}$, $\mathcal{V}_{\mathbf{D}_i}$ denoting the user and item sets of $\mathbf{D}_i$, and interactions $\mathcal{T}_{\mathbf{D}_i}$ between them. There is no overlap between the user and item sets of any two domains $\mathbf{D}_i$, $\mathbf{D}_j$.

In the implicit feedback setting, each interaction $t \in \mathcal{T}_{\mathbf{D}_i}$ is a tuple $t = (u, \mathbf{c}, v)$ where $u \in \mathcal{U}_{\mathbf{D}_i}, v \in \mathcal{V}_{\mathbf{D}_i}$ and context vector $\mathbf{c} \in \mathbb{R}^{|\mathbf{C}|}$. For the explicit feedback setting, $\mathcal{T}_{\mathbf{D}_i}$ is replaced by ratings $\mathcal{R}_{\mathbf{D}_i}$, where each rating is a tuple $r = (u, \mathbf{c}, v, r_{uv})$, with the rating value r_{uv} (other notations are the same). For simplicity, all interactions in all domains have the same set of context features. In our datasets, the context feature set $\mathbf{C}$ contains three different types of context features, interactional features $\mathbf{C_I}$ (such as time of interaction), historical features $\mathbf{C_H}$ (such as a user's average spend), and attributional features $\mathbf{C_A}$ (such as restaurant cuisine or user age). Thus each context vector $\mathbf{c}$ contains these three types of features for that interaction, i.e., $\mathbf{c} = [\mathbf{c_I}, \mathbf{c_H}, \mathbf{c_A}]$.

Under implicit feedback, we rank items $v \in \mathcal{V}_{\mathbf{D}}$ given user $u \in \mathcal{U}_{\mathbf{D}}$ and context $\mathbf{c}$. In the explicit feedback scenario, we predict rating r_{uv} for $v \in \mathcal{V}_{\mathbf{D}}$ given $u \in \mathcal{U}_{\mathbf{D}}$ and $\mathbf{c}$. Our transfer objective is to reduce the rating or ranking error in a set of disjoint sparse target domains $\{\mathbf{D}_t\} \subset \mathbb{D}$ given the dense source domain $\mathbf{D}_s \in \mathbb{D}$.

6.4 OUR APPROACH

This section describes a scalable, modular architecture to extract pooled contextual invariants and employ them to guide the learned user and item embedding spaces.

6.4.1 Modular Architecture

We achieve context-guided embedding learning via four synchronized neural modules with complementary semantic objectives:

- **Context Module $\mathcal{M}^1$:** Extracts contextual invariants driving user-item interactions in the dense source domain.
- **Embedding Modules $\mathcal{M}^2_{\mathcal{U}}, \mathcal{M}^2_{\mathcal{V}}$:** Domain-specific user and item embedding spaces ($\mathcal{U}$, $\mathcal{V}$ denote users and items).
- **Context-Conditioned Clustering Modules $\mathcal{M}^3_{\mathcal{U}}, \mathcal{M}^3_{\mathcal{V}}$:** $\mathcal{M}^3_{\mathcal{U}}$ and $\mathcal{M}^3_{\mathcal{V}}$ reorient the user and item embeddings with the contextual invariants extracted by $\mathcal{M}^1$ respectively.
- **Mapping/Ranking Module $\mathcal{M}^4$:** Generate interaction likelihoods with the context-conditioned representations of $\mathcal{M}^3$.

Context-driven modules $\mathcal{M}^1$, $\mathcal{M}^3$ and $\mathcal{M}^4$ contain the meta-layers that are transferred from the dense to the sparse domains (i.e., shared or meta-modules). In contrast, $\mathcal{M}^2$ contains the domain-specific user and item representations. Our architecture provides a separation between the domain-specific $\mathcal{M}^2$ module and shared context-based transforms in the other modules (Figure 6.1). We now detail each module in our overall architecture.

6.4.2 Context Module Description (Module $\mathcal{M}^1$)

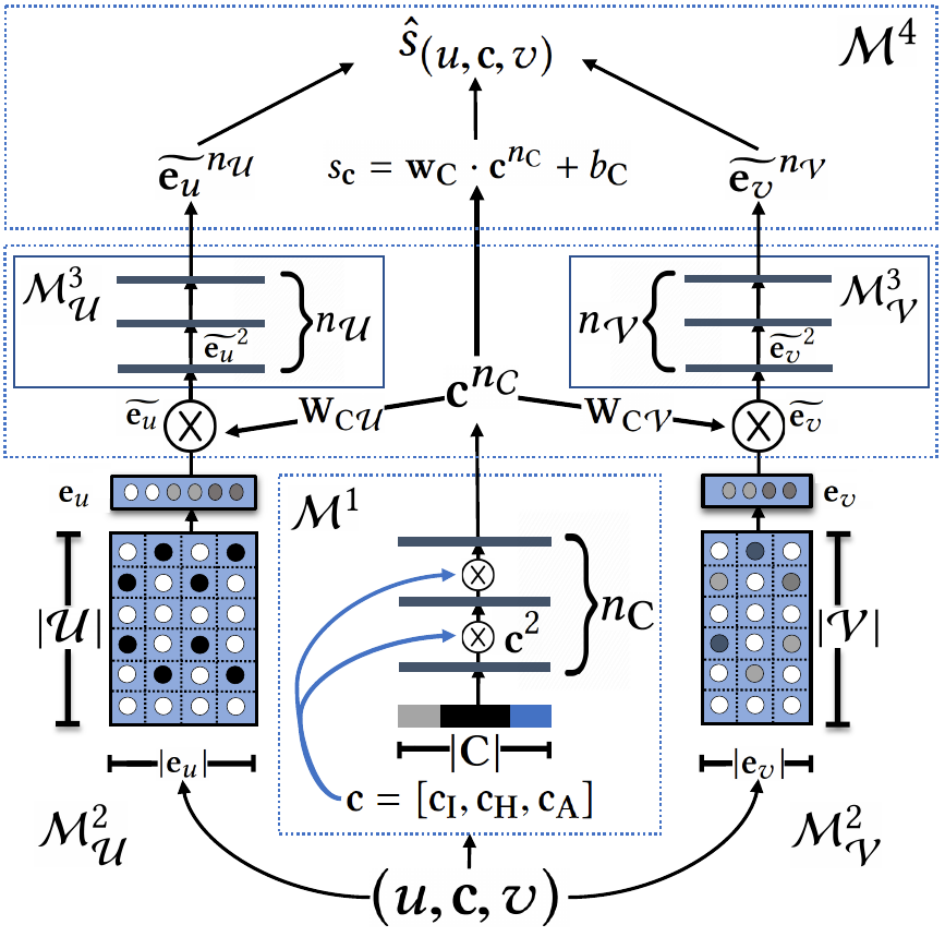
User-item interactions are driven by context feature intersections that are inherently *multiplicative* (i.e., assumptions of independent feature contributions are insufficient). They are often missed in the Naive-Bayes assumption of additive models such as feature-attention [15, 58]. Inspired by the past success of low-rank feature pooling [15, 86], our context module extracts low-rank multi-linear combinations of context to describe interactions and build expressive representations. The first layer in $\mathcal{M}^1$ transforms context $\mathbf{c}$ of an interaction $(u, \mathbf{c}, v)$ as follows:

$$\mathbf{c}^2 = \sigma(\underbrace{\mathbf{W}^2\mathbf{c} \oplus (\mathbf{b}^2 \otimes \mathbf{c})}_{\text{Weighted linear transform}}) \otimes \underbrace{\mathbf{c}}_{\text{Element-wise interaction}} \tag{6.1}$$

where $\oplus, \otimes$ denote element-wise product and sum, i.e.,

$$\mathbf{c}^2_i \propto \mathbf{c}_i \times \sigma(\mathbf{b}^2_i \mathbf{c}_i + \sum_j \mathbf{W}^2_{ij} \mathbf{c}_j) \tag{6.2}$$

Figure 6.1: Our overall recommender architecture, highlighting all four modules, $\mathcal{M}^1$ to $\mathcal{M}^4$.



Thus, $\mathbf{c}_i^2$ (i^{th}-component of $\mathbf{c}^2$) incorporates a weighted bivariate interaction between $\mathbf{c}_i$ and other context factors $\mathbf{c}_j$, including itself. We then repeat this transformation over multiple stacked layers with each layer using the previous output:

$$\mathbf{c}^n = \sigma(\mathbf{W}^n \mathbf{c}^{n-1} \oplus (\mathbf{b}^n \otimes \mathbf{c}^{n-1})) \otimes \mathbf{c} \tag{6.3}$$

Each layer interacts *n-variate* terms from the previous layer with $\mathbf{c}$ to form *n+1-variate* terms. However, since each layer has only $|\mathbf{C}|$ outputs (i.e., low-rank), $\mathbf{W}^n$ prioritizes the most effective *n-variate* combinations of $\mathbf{c}$ (typically, a very small fraction of all combinations is useful). We can choose the number of layers $n_\mathbf{C}$ depending on the required order of the final combinations $\mathbf{c}^{n_C}$.

Multimodal Residuals for Discriminative Correlation Mining: In addition to discovering the most critical context combinations, we incorporate the information gain associated

Table 6.1: Modules and Parameter Notations.

Modules	**Learned Parameters**
Domain-Specific $(\mathcal{M}^2_{\mathcal{U}}, \mathcal{M}^2_{\mathcal{V}})$	**Embeddings** $\mathbf{e}_u \forall u \in \mathcal{U}_{\mathbf{D}}$, $\mathbf{e}_v \forall v \in \mathcal{V}_{\mathbf{D}}$ **Biases** (only under explicit feedback) $s, s_u \forall u \in \mathcal{U}_{\mathbf{D}}, s_v \forall v \in \mathcal{V}_{\mathbf{D}}$
Shared Modules $(\mathcal{M}^1, \mathcal{M}^3, \mathcal{M}^4)$	$\mathcal{M}^1$ eq. (6.3) $(\mathbf{W}^i, \mathbf{b}^i) \forall i = [1, \cdots, n_{\mathbf{C}}]$ $\mathcal{M}^1$ eq. (6.5) $\mathbf{s_I}, \mathbf{s_H}, \mathbf{s_A}; \mathbf{W_I}, \mathbf{W_H}, \mathbf{W_A}$ $\mathcal{M}^3$ eq. (6.7) $\mathbf{W_{C\mathcal{U}}}, \mathbf{W_{C\mathcal{V}}}$ $\mathcal{M}^3_{\mathcal{U}}$ eq. (6.9) $(\mathbf{W}^i_{\mathcal{U}}, \mathbf{b}^i_{\mathcal{U}}) \forall i = [1, \cdots, n_{\mathcal{U}}]$ $\mathcal{M}^3_{\mathcal{V}}$ eq. (6.9) $(\mathbf{W}^i_{\mathcal{V}}, \mathbf{b}^i_{\mathcal{V}}) \forall i = [1, \cdots, n_{\mathcal{V}}]$ $\mathcal{M}^4$ eq. (6.10) $\mathbf{W_C}, \mathbf{b_C}$

with pairwise interactions of context features [196]. For instance, the item cost feature is more informative in interactions where users deviate from their historical spending patterns. Specifically, pairs of signals (e.g., cost & user history) enhance or diminish each other's impact, i.e.,

$$\mathbf{c}_i = \mathbf{c}_i + \Sigma_j \delta_{\mathbf{c}_i|\mathbf{c}_j} \tag{6.4}$$

We simplify Equation (6.4) by only considering cross-modal effects across interactional, historical, and attributional context, i.e.,

$$\delta_{\mathbf{c_I}|\mathbf{c_H},\mathbf{c_A}} = \underbrace{\mathbf{s_I}}_{\text{Scaling factor}} \otimes \underbrace{tanh(\mathbf{W_{IH}} \times \mathbf{c_H} + \mathbf{W_{IA}} \times \mathbf{c_A} + \mathbf{b_I})}_{\text{Info gain/loss}} \tag{6.5}$$

and likewise for $\delta_{\mathbf{c_H}}$, $\delta_{\mathbf{c_A}}$. Information gains are computed before $\mathbf{c}^2$ to cascade to further layers.

6.4.3 Context Conditioned Clustering Module Description ($\mathcal{M}^3$)

We combine domain-specific embeddings $\mathcal{M}^2$ with the context combinations extracted by $\mathcal{M}^1$ to generate context-conditioned user and item representations. Specifically, we introduce the following bilinear transforms,

$$\widetilde{\mathbf{e}_u} = \mathbf{e}_u \otimes \sigma(\mathbf{W_{C\mathcal{U}}} \times \mathbf{c}^{n_{\mathbf{C}}}) \tag{6.6}$$

$$\widetilde{\mathbf{e}_v} = \mathbf{e}_v \otimes \sigma(\mathbf{W_{C\mathcal{V}}} \times \mathbf{c}^{n_{\mathbf{C}}}) \tag{6.7}$$

where, $\mathbf{W_{C\mathcal{U}}} \in \mathbb{R}^{|\mathbf{e}_u| \times |\mathbf{C}|}$, $\mathbf{W_{C\mathcal{V}}} \in \mathbb{R}^{|\mathbf{e}_v| \times |\mathbf{C}|}$ are learned parameters that map the most

relevant context combinations to the user and item embeddings. We further introduce $n_{\mathcal{U}}$ feedforward *RelU* layers to cluster the representations,

$$\widetilde{\mathbf{e}_u}^2 = \sigma(\mathbf{W}^2_{\mathcal{U}}\widetilde{\mathbf{e}_u} + \mathbf{b}^2_{\mathcal{U}}) \tag{6.8}$$

$$\widetilde{\mathbf{e}_u}^n = \sigma(\mathbf{W}^n_{\mathcal{U}}\widetilde{\mathbf{e}_u}^{n-1} + \mathbf{b}^n_{\mathcal{U}}) \tag{6.9}$$

Analogously, we obtain context-conditioned item representations $\widetilde{\mathbf{e}_v}^2, \cdots, \widetilde{\mathbf{e}_v}^{n_{\mathcal{V}}}$ with $n_{\mathcal{V}}$ feedforward *RelU* layers.

The bilinear transforms in eq. (6.7) introduce *dimension alignment* for both $\widetilde{\mathbf{e}_u}^{n_{\mathcal{U}}}$ and $\widetilde{\mathbf{e}_v}^{n_{\mathcal{V}}}$ with the context output $\mathbf{c}^{n_{\mathbf{C}}}$. Thus, when $\mathcal{M}^3$ and $\mathcal{M}^1$ layers are transferred to a sparse target domain, we can directly backpropagate to guide the target domain user and item embeddings with the target domain interactions.

6.4.4 Source Domain Training Algorithm

In the source domain, we train all modules and parameters (Table 6.1) with ADAM optimization [87] and dropout regularization [188].

Self-Paced Curriculum via Contextual Novelty: Focusing on harder data samples accelerates and stabilizes stochastic gradients [25, 127]. Since our learning process is grounded on context, novel interactions display uncommon or *interesting* context combinations. Let $\mathcal{L}_{(u,\mathbf{c},v)}$ denote the loss function for an interaction $(u, \mathbf{c}, v)$. We propose an inverse novelty measure referred as the context-bias, $s_{\mathbf{c}}$, which is self-paced by the context combinations learned by $\mathcal{M}^1$ in Equation (6.3),

$$s_{\mathbf{c}} = \mathbf{w}_{\mathbf{C}} \cdot \mathbf{c}^{n_{\mathbf{C}}} + b_{\mathbf{C}} \tag{6.10}$$

We then attenuate the loss $\mathcal{L}_{(u,\mathbf{c},v)}$ for this interaction as,

$$\mathcal{L}'_{(u,\mathbf{c},v)} = \mathcal{L}_{(u,\mathbf{c},v)} - s_{\mathbf{c}} \tag{6.11}$$

The resulting novelty loss $\mathcal{L}'_{(u,\mathbf{c},v)}$ decorrelates interactions [29, 81] by emulating variance-reduction in the *n-variate* pooled space of $\mathbf{c}^{n_{\mathbf{C}}}$. $\mathcal{L}'_{(u,\mathbf{c},v)}$ determines the user and item embedding spaces, inducing a novelty-weighted training curriculum focused on harder samples as training proceeds. We now describe loss $\mathcal{L}_{(u,\mathbf{c},v)}$ for the explicit and implicit feedback scenarios.

Ranking our Recommendations: In the *implicit feedback setting*, predicted likelihood $\hat{s}_{(u,\mathbf{c},v)}$ is computed with the context-conditioned embeddings (Equation (6.9)) and context-

bias (Equation (6.11)) as,

$$\hat{s}_{(u,\mathbf{c},v)} = \widetilde{\mathbf{e}}_u^{\,n_\mathcal{U}} \cdot \widetilde{\mathbf{e}}_v^{\,n_\mathcal{V}} + s_\mathbf{c} \tag{6.12}$$

The loss for all the possible user-item-context combinations in domain $\mathbf{D}$ is,

$$\mathcal{L}_\mathbf{D} = \sum_{u \in \mathcal{U}_\mathbf{D}} \sum_{v \in \mathcal{V}_\mathbf{D}} \sum_{\mathbf{c} \in \mathbb{R}^{|\mathbf{c}|}} ||\mathbb{I}_{(u,\mathbf{c},v)} - \hat{s}_{(u,\mathbf{c},v)}||^2 \tag{6.13}$$

where $\mathbb{I}$ is the binary indicator $(u, \mathbf{c}, v) \in \mathcal{T}_\mathbf{D}$. $\mathcal{L}_\mathbf{D}$ is intractable due to the large number of contexts $\mathbf{c} \in \mathbb{R}^{|\mathbf{c}|}$. We develop a negative sampling approximation for implicit feedback with two learning objectives - identify the likely item given the user and interaction context, and identify the likely context given the user and the item. We thus construct two negative samples for each $(u, \mathbf{c}, v) \in \mathcal{T}_\mathbf{D}$ at random: Item negative with the true context, $(u, \mathbf{c}, v^-)$ and context negative with the true item, $(u, \mathbf{c}^-, v)$. $\mathcal{L}_\mathbf{D}$ then simplifies to,

$$\mathcal{L}_\mathbf{D} = \sum_{\mathcal{T}_\mathbf{D}} ||1 - \hat{s}_{(u,\mathbf{c},v)}||^2 + \sum_{(u,\mathbf{c},v^-)} ||\hat{s}_{(u,\mathbf{c},v^-)}|| + \sum_{(u,\mathbf{c}^-,v)} ||\hat{s}_{(u,\mathbf{c}^-,v)}|| \tag{6.14}$$

In the explcit feedback setting, we introduce two additional bias terms, one for each user, s_u and one for each item, s_v. These terms account for user and item rating eccentricities (e.g., users who always rate well), so that the embeddings are updated with the relative rating differences. Finally, global bias s accounts for the rating scale, e.g., 0-5 vs. 0-10. Thus the predicted rating is given as,

$$\hat{r}_{(u,\mathbf{c},v)} = \widetilde{\mathbf{e}}_v^{\,n_\mathcal{V}} \cdot \widetilde{\mathbf{e}}_u^{\,n_\mathcal{U}} + s_\mathbf{c} + s_u + s_v + s \tag{6.15}$$

Negative samples are not required in the explicit feedback setting,

$$\mathcal{L}_\mathbf{D}^{explicit} = \sum_{(u,\mathbf{c},v,r_{uv}) \in \mathcal{R}_\mathbf{D}} ||r_{uv} - \hat{r}_{(u,\mathbf{c},v)}||^2 \tag{6.16}$$

We now detail our approach to transfer the shared modules from the source domain to sparse target domains.

6.5 TRANSFER TO TARGET DOMAINS

Our formulation enables us to train the shared modules $(\mathcal{M}^1)_\mathbf{S}, (\mathcal{M}^3)_\mathbf{S}$ and $(\mathcal{M}^4)_\mathbf{S}$ on a dense source domain $\mathbf{S}$, and transfer them to a sparse target domain $\mathbf{T}$ to guide its embedding module $(\mathcal{M}^2)_\mathbf{T}$. Each shared module $\mathcal{M}$ encodes inputs $\mathbf{x}_\mathcal{M}$ to generate output

representations $\mathbf{y}_{\mathcal{M}}$. In each domain $\mathbf{T}$, module $(\mathcal{M})_{\mathbf{T}}$ determines the joint input-output distribution,

$$p_{\mathbf{T}}(\mathbf{y}_{\mathcal{M}}, \mathbf{x}_{\mathcal{M}}) = p_{\mathbf{T}}(\mathbf{y}_{\mathcal{M}}|\mathbf{x}_{\mathcal{M}}) \times p_{\mathbf{T}}(\mathbf{x}_{\mathcal{M}}) \tag{6.17}$$

where the parameters of $(\mathcal{M})_{\mathbf{T}}$ determine the conditional $p_{\mathbf{T}}(\mathbf{y}_{\mathcal{M}}|\mathbf{x}_{\mathcal{M}})$ and $p_{\mathbf{T}}(\mathbf{x}_{\mathcal{M}})$ describes the inputs to module $(\mathcal{M})_{\mathbf{T}}$ in domain $\mathbf{T}$. **Adaptation:** There are two broad strategies to adapt module $\mathcal{M}$ to a new target domain $\mathbf{T}$:

- **Parameter Adaptation:** We can retrain the parameters of module $\mathcal{M}$ for target domain $\mathbf{T}$ thus effectively changing the conditional $p_{\mathbf{T}}(\mathbf{y}_{\mathcal{M}}|\mathbf{x}_{\mathcal{M}})$ in eq. (8.2), or,
- **Input Adaptation:** Modify the input distribution $p_{\mathbf{T}}(\mathbf{x}_{\mathcal{M}})$ in each domain $\mathbf{T}$ without altering the parameters of $\mathcal{M}$.

We now explore module transfer with both types of adaptation strategies towards achieving three key objectives. First, the transferred modules must be optimized to be effective on each target domain $\mathbf{T}$. Second, we aim to minimize the computational costs of adapting to new domains by maximizing the reuse of module parameters between the source $\mathbf{S}$ and target domains $\mathbf{T}$. Finally, we must avoid overfitting the transferred modules to the samples in the sparse target domain $\mathbf{T}$.

6.5.1 Direct Layer-Transfer

We first train all four modules on the source $\mathbf{S}$ and each target domain $\mathbf{T}$ in isolation. We denote these pre-trained modules as $(\mathcal{M}^i)_{\mathbf{S}}$ and $(\mathcal{M}^i)_{\mathbf{T}}$ for source domain $\mathbf{S}$ and a target domains $\mathbf{T}$ respectively. We then replace the shared modules in all the target domain models with the source-trained version, i.e., $(\mathcal{M}^1)_{\mathbf{T}} = (\mathcal{M}^1)_{\mathbf{S}}$, $(\mathcal{M}^3)_{\mathbf{T}} = (\mathcal{M}^3)_{\mathbf{S}}$, $(\mathcal{M}^4)_{\mathbf{T}} = (\mathcal{M}^4)_{\mathbf{S}}$, while the domain-specific embeddings $(\mathcal{M}^2)_{\mathbf{T}}$ are not changed in the target domains. Clearly, direct layer-transfer involves no overhead and trivially prevents overfitting. However, we need to adapt the transferred modules for optimal target performance, i.e., either adapt the parameters or the input distributions for the transferred modules in each target $\mathbf{T}$. We now develop these adaptation strategies building on layer-transfer.

6.5.2 Simulated Annealing

Simulated annealing is a stochastic local-search algorithm that implicitly thresholds parameter variations in the gradient space by decaying the gradient learning rates [90]. As a

Table 6.2: Comparing the objectives in Section 6.5 addressed by our meta-transfer approaches for sparse target domains.

Adaptation Method	**Target Adaptation**	**Resists Overfitting**	**Extra compute per target**	**Extra parameters per target**
Layer-Transfer	No adaptation	Yes, trivially	None	None, module params reused
Simulated Annealing	Yes, module parameters	Yes, stochastic updates	Update costs for all parameters	All parameters (Table 6.1)
Regularized Residuals	Yes, module inputs	Yes, via distributional consistency	Residual layer updates with distributional regularization	Residual layer parameters

simple and effective adaptation strategy, we anneal each transferred module $\mathcal{M}$ in the target domain $\mathbf{T}$ with exponentially decaying learning rates to prevent overfitting stochastically:

$$(m)_{b+1} = (m)_b + \eta_b \frac{\partial \mathcal{L}_b}{\partial m},\ \eta_b = \eta_0 e^{-\lambda b} \tag{6.18}$$

where m denotes any parameter of transferred module $\mathcal{M}$ (Table 6.1), b is the stochastic gradient batch index in the target domain and $\mathcal{L}_b$ is the batch loss for batch b. Our annealing strategy stochastically generates a robust parameter search schedule for transferred modules $\mathcal{M}^1, \mathcal{M}^3, \mathcal{M}^4$, with η_b decaying to zero after one annealing epoch. While annealing the transferred modules, domain-specific module $\mathcal{M}^2$ is updated with the full learning rate η_0. Clearly, annealing modifies the conditional $p_{\mathbf{T}}(\mathbf{y}_{\mathcal{M}}|\mathbf{x}_{\mathcal{M}})$ in eq. (8.2) via parameter adaptation. However, annealing transferred modules in each target domain is somewhat expensive, and the annealed parameters are not shareable, thus causing scalability limitations in the one-to-many transfer scenario. We now develop a lightweight residual adaptation strategy to achieve input adaptation without modifying any shared module parameters in the target domains to overcome the above scalability challenges.

6.5.3 Distributionally Regularized Residuals

We now develop an approach to reuse the source modules with target-specific input adaptation, thus addressing the scalability concerns of parameter adaptation methods.

Enabling Module Reuse with Residual Input Adaptation: In eq. (8.2), module $\mathcal{M}$ implements the conditional $p(\mathbf{y}_{\mathcal{M}}|\mathbf{x}_{\mathcal{M}})$. To maximize parameter reuse, we share these modules across the source and target domains (i.e., $p_{\mathbf{T}}(\mathbf{y}_{\mathcal{M}}|\mathbf{x}_{\mathcal{M}}) = p_{\mathbf{S}}(\mathbf{y}_{\mathcal{M}}|\mathbf{x}_{\mathcal{M}})$) and introduce

target-specific residual perturbations to account for their eccentricities [124] by modifying the input distributions $p_{\mathbf{T}}(\mathbf{x}_{\mathcal{M}})$. Target-specific input adaptation overcomes the need for an expensive end-to-end parameter search. Our adaptation problem thus reduces to learning an input modifier for each target domain $\mathbf{T}$ and shared module $\mathcal{M} \in [\mathcal{M}^1, \mathcal{M}^3, \mathcal{M}^4]$, i.e., for each $\mathcal{M}, \mathbf{T}$.

Residual transformations enable the flow of information between layers without the gradient attenuation of inserting new non-linear layers, resulting in numerous optimization advantages [56]. Given the module-input $\mathbf{x}_{\mathcal{M}}$ to the shared module $\mathcal{M}$ in target domain $\mathbf{T}$, we learn a module and target specific residual transform:

$$\mathbf{x}_{\mathcal{M}} = \mathbf{x}_{\mathcal{M}} + \delta_{\mathcal{M},\mathbf{T}}(\mathbf{x}_{\mathcal{M}}) \tag{6.19}$$

The form of the residual function δ is flexible. We chose a single non-linear residual layer, $\delta(\mathbf{x}) = tanh(\mathbf{W}\mathbf{x} + \mathbf{b})$. We can intuitively balance the complexity and number of such residual layers. Note that the above residual strategy involves learning the $\delta_{\mathcal{M},\mathbf{T}}$ layers with feedback from only the sparse target domain samples. To avoid overfitting, we need a scalable regularization strategy to regularize $p_{\mathbf{T}}(\mathbf{x}_M)$ in each target domain. We propose to leverage the source input distribution as a common baseline for all the target domains, i.e., intuitively, $p_{\mathbf{S}}(\mathbf{x}_M)$ provides a common center for $p_{\mathbf{T}}(\mathbf{x}_M)$ in the different target domains. This effectively anchors the residual functions and prevents overfitting to noisy samples.

Scalable Distributional Regularization for Residual Learning: Learning pairwise regularizers between each $p_{\mathbf{T}}(\mathbf{x}_M)$ and the source input distribution $p_{\mathbf{S}}(\mathbf{x}_M)$ is not a scalable solution. Instead we train a universal regularizer for each module $\mathcal{M}$ on the source $p_{\mathbf{S}}(\mathbf{x}_{\mathcal{M}})$, and apply this pre-trained regularizer when we fit the residual layers $\delta_{\mathcal{M},\mathbf{T}}$ in each target domain. Our key intuition is to treat the regularizer for the inputs of each module $\mathcal{M}$ as a one-class decision-boundary [173], described by the dense regions in the source domain, i.e., $p_{\mathbf{S}}(\mathbf{x}_{\mathcal{M}})$. Unlike adversarial models that are trained with both the source and target distributions [162], we propose a novel approach to learn distributional input regularizers for the shared modules with just the source domain inputs.

For each shared module, the learned regularizer anticipates hard inputs across the target domains without accessing the actual samples. We introduce a variational encoder $\mathcal{E}_{\mathcal{M}}$ with RelU layers to map inputs $\mathbf{x}_{\mathcal{M}} \sim p_{\mathbf{S}}(\mathbf{x}_{\mathcal{M}})$ to a lower-dimensional reference distribution $\mathbf{N}(0, \mathbb{I})$ [36]. Simultaneously, we add poisoning model $\mathcal{P}_{\mathcal{M}}$ to generate sample-adaptive noise $\mathcal{P}_{\mathcal{M}}(\mathbf{x}_{\mathcal{M}})$ to generate poisoned samples $\widetilde{\mathbf{x}}_{\mathcal{M}} = \mathbf{x}_{\mathcal{M}} + \mathcal{P}_{\mathcal{M}}(\mathbf{x}_{\mathcal{M}})$ with the source domain inputs $\mathbf{x}_{\mathcal{M}} \sim p_{\mathbf{S}}(\mathbf{x}_{\mathcal{M}})$. We define the encoder loss to train $\mathcal{E}_{\mathcal{M}}$ as follows:

$$\mathcal{L}_{\mathcal{E}_\mathcal{M}} = D(p(\mathcal{E}_\mathcal{M}(\mathbf{x}_\mathcal{M})) \parallel \mathbf{N}(0, \mathbb{I})) - D(p(\mathcal{E}_\mathcal{M}(\widetilde{\mathbf{x}}_\mathcal{M})) \parallel \mathbf{N}(0, \mathbb{I})) \tag{6.20}$$

where $D(p \parallel q)$ denotes the *KL-Divergence* of distributions p and q. The above loss enables $\mathcal{E}_\mathcal{M}$ to separate the true and poisoned samples across the $\mathbf{N}(0, \mathbb{I})$ hypersphere in its encoded space. Since $\mathcal{E}_\mathcal{M}(\mathbf{x}_\mathcal{M})$ involves a stochastic sampling step, gradients can be estimated with a *reparametrization trick* using random samples to eliminate stochasticity in the loss $\mathcal{L}_{\mathcal{E}_\mathcal{M}}$[36]. Conversely, the loss for our poisoning model $\mathcal{P}_\mathcal{M}$ is given by,

$$\mathcal{L}_{\mathcal{P}_\mathcal{M}} = D(p(\mathcal{E}_\mathcal{M}(\widetilde{\mathbf{x}_\mathcal{M}} \parallel \mathbf{N}(0, \mathbb{I})) - \log ||\mathcal{P}_\mathcal{M}(\mathbf{x}_\mathcal{M})|| \tag{6.21}$$

Note the first term in Equation (6.21) attempts to confuse $\mathcal{E}_\mathcal{M}$ into encoding poisoned examples $\widetilde{\mathbf{x}_\mathcal{M}} = \mathbf{x}_\mathcal{M} + \mathcal{P}_\mathcal{M}(\mathbf{x}_\mathcal{M})$ in the reference distribution, while the second term prevents the degenerate solution $\mathcal{P}_\mathcal{M}(\mathbf{x}_\mathcal{M}) = 0$. Equation (6.20) and Equation (6.21) are alternatingly optimized, learning sharper decision boundaries as training proceeds. With the above alternating optimization, we pre-train the encoders $\mathcal{E}_\mathcal{M}$ for the three shared modules on the source domain $\mathbf{S}$. We now describe how we use these encoders to regularize the residual layers $\delta_{\mathcal{M},\mathbf{T}}$ in each target domain $\mathbf{T}$.

Distributionally-Regularized Target Loss: For each target domain $\mathbf{T}$, we learn three residual layers for the module inputs $\mathbf{c}^2$, $\widetilde{\mathbf{e}_u}$ and $\widetilde{\mathbf{e}_v}$ for $\mathcal{M}_1, \mathcal{M}_3^\mathcal{U}, \mathcal{M}_3^\mathcal{V}$ respectively. The inputs to $\mathcal{M}_4$, $\widetilde{\mathbf{e}_u}^{n_\mathcal{U}}$, $\widetilde{\mathbf{e}_v}^{n_\mathcal{V}}$ are not adapted. Thus, we learn three variational encoders in the source domain as described in Section 6.5.3, $\mathcal{E}_\mathbf{C}, \mathcal{E}_\mathcal{U}$ and $\mathcal{E}_\mathcal{V}$ for $\mathbf{c}^2$, $\widetilde{\mathbf{e}_u}$ and $\widetilde{\mathbf{e}_v}$ respectively. Consider target interactions $(u, \mathbf{c}, v) \in \mathcal{T}_\mathbf{T}$. In the absence of distributional regularization, the loss is identical to the first term in Equation (6.14). However, we now apply regularizers to $\mathbf{c}^2$, $\widetilde{\mathbf{e}_u}, \widetilde{\mathbf{e}_v}$:

$$\begin{aligned}\mathcal{L}_{\mathcal{T}_\mathbf{T}}^{reg} = \mathcal{L}_{\mathcal{T}_\mathbf{T}} + D(p_\mathbf{T}(\mathcal{E}_\mathcal{U}(\widetilde{\mathbf{e}_u})) \parallel \mathbf{N}(0, \mathbb{I})) + \\ D(p_\mathbf{T}(\mathcal{E}_\mathcal{V}(\widetilde{\mathbf{e}_v})) \parallel \mathbf{N}(0, \mathbb{I})) + D(p_\mathbf{T}(\mathcal{E}_\mathbf{C}(\mathbf{c}^2)) \parallel \mathbf{N}(0, \mathbb{I}))\end{aligned} \tag{6.22}$$

Again, the gradients can be estimated with the *reparametrization trick* on the stochastic *KL-divergence* terms[36] as in Section 6.5.3. The residual layers are then updated as in Section 6.4.4 with $\mathcal{L}_{\mathcal{T}_\mathbf{T}}^{reg}$ replacing the first term in Equation (6.14).

6.6 EXPERIMENTAL RESULTS

In this section, we present experimental analyses on diverse multi-domain recommendation datasets and show two key results. First, when we adapt modules trained on a rich source

Table 6.3: *Source* and *Target* statistics for each of our datasets. Source states denoted S have more interactions and interaction density per user than target states denoted $\mathbf{T}_i$. Note that $|\mathbf{C}|$ denotes the length of the context feature vector in each domain, while **k** and **m** denotes thousands and millions of interactions respectively.

Dataset		State	Users	Items	Interactions
	S	**Bay-Area CA**	**1.20 m**	**8.90 k**	**25.0 m**
FT-Data	$\mathbf{T}_1$	Arkansas	0.40 m	3.10 k	5.20 m
$\|\mathbf{C}\| = 220$	$\mathbf{T}_2$	Kansas	0.35 m	2.90 k	5.10 m
	$\mathbf{T}_3$	New-Mexico	0.32 m	2.80 k	6.20 m
	$\mathbf{T}_4$	Iowa	0.30 m	3.00 k	4.80 m
	S	**Pennsylvania**	**10.3 k**	**5.5 k**	**170 k**
Yelp	$\mathbf{T}_1$	Alberta, Canada	5.10 k	3.5 k	55.0 k
$\|\mathbf{C}\| = 120$	$\mathbf{T}_2$	Illinois	1.80 k	1.05 k	23.0 k
	$\mathbf{T}_3$	S.Carolina	0.60 k	0.40 k	6.20 k
Google	**S**	**California**	**46 k**	**28 k**	**320 k**
Local	$\mathbf{T}_1$	Colorado	10 k	5.7 k	51.0 k
$\|\mathbf{C}\| = 90$	$\mathbf{T}_2$	Michigan	7.0 k	4.0 k	29.0 k
	$\mathbf{T}_3$	Ohio	5.4 k	3.2 k	23.0 k

domain to the sparse target domains, we significantly reduce the computational costs and improve performance in comparison to learning directly on the sparse domains. Second, our model is comparable to *state-of-the-art* baselines when trained on a single domain without transfer.

6.6.1 Datasets and Baselines

We evaluate our recommendation model both with and without module transfer over the publicly available *Yelp*[3] and *Google Local Reviews*[4] datasets for benchmarking purposes. Reviews are split across U.S and Canadian states in these datasets. We treat each state as a separate recommendation domain for training and transfer purposes. There is no *user or item overlap* across the states (recommendation domains) in any of our datasets. We repeat our experiments with a large-scale restaurant transaction dataset obtained from Visa (referred to as *FT-Data*), also split across U.S. states.

Google Local Reviews Dataset: (*Explicit feedback*)[5][57, 151]: Users rate businesses on a 0-5 scale with temporal, spatial, and textual context available for each review. We

[3] `https://www.yelp.com/dataset/challenge`

[4] `http://cseweb.ucsd.edu/~jmcauley/datasets.html`

[5] `http://cseweb.ucsd.edu/~jmcauley/datasets.html`

also infer additional context features - users' preferred locations on weekdays and weekends, spatial patterns and preferred product categories.

Yelp Challenge Dataset (*Explicit feedback*) [6]: Users rate restaurants on a 0-5 scale, reviews include similar context features as the Google Local dataset. Further, *user check-ins* and restaurant attributes (e.g., *accepts-cards*) are available.

FT-Data (*Implicit feedback*): Contains the credit/debit card payments of users to restaurants in the U.S, with spatial, temporal, financial context features, and inferred transaction attributes. We leverage transaction histories also to infer user spending habits, restaurant popularity, peak hours, and tipping patterns.

In each dataset, we extract the same context features for every state with statewise normalization, either with min-max normalization or quantile binning. We retain users and items with three or more reviews in the Google Local dataset and ten or more reviews in the Yelp dataset. In *FT-Data*, we retain users and restaurants with over ten, twenty transactions, respectively, over three months. In each dataset, we choose a dense state with ample data as the source domain where all modules are trained, and multiple sparse states as target domains for module transfer from the source.

6.6.2 Source to Target Module Transfer

We evaluate the performance gains obtained when we transfer or adapt modules $\mathcal{M}^1, \mathcal{M}^3$ and $\mathcal{M}^4$ from the source state to each target state, in comparison to training all four modules directly on the target. We also compare target domain gains with *state-of-the-art* meta-learning baselines:

LWA [203]: Learns a shared meta-model across all domains, with a user-specific linear component.

NLBA [203]: Replaces LWA's linear component with a neural network with user-specific layer biases.

$\mathbf{s^2}$-Meta [40]: Develops a meta-learner to instantiate and train recommender models for each scenario. In our datasets, scenarios are the different states.

Direct Layer-Transfer (Our Variant): Transfers source-trained meta-modules to the target-trained models as in Section 6.5.1.

Anneal (Our Variant): We apply simulated annealing to adapt the transferred meta-modules to the target as in Section 6.5.2.

DRR - Distributionally Regularized Residuals: **(Our Main Approach)** Adapts the inputs of each transferred module with separate residual layers in each target state (as

[6] https://www.yelp.com/dataset/challenge

described in Section 6.5.3).

6.6.3 Single Domain Recommendation Performance

We also evaluate the performance of our models independently without transfer on the source and target states in each dataset. We compare with the following *state-of-the-art* recommendation baselines:

NCF [60]: *State-of-the-art* non context-aware model for comparisons and context validation.

CAMF-C [11]: Augments Matrix Factorization to incorporate a context-bias term for item latent factors. This version assumes a fixed bias for a given context feature for all items.

CAMF [11]: CAMF-C with separate context bias values for each item. We use this version for comparisons.

MTF [83]: Obtains latent representations via decomposition of the User-Item-Context tensor. This model scales very poorly with the size of the context vector.

NFM [58]: Employs a bilinear interaction model applied to the context features of each interaction for representation.

AFM [225]: Incorporates an attention mechanism to reweight the bilinear pooled factors in the NFM model. Scales poorly with the number of pooled contextual factors.

AIN [137]: Reweights the interactions of user and item representations with each contextual factor via attention.

MMT-Net (Our Main Approach): We refer to our model with all four modules as Multi-Linear Module Transfer Network (MMT-Net).

FMT-Net (Our Variant): We replace $\mathcal{M}^1$s layers with feedforward *RelU* layers to demonstrate the importance of multiplicative context invariants.

MMT-Net Multimodal (Our Variant): **MMT-Net** with the information-gain terms described in Equation (6.5). Only applied to *FT-Data* due to lack of interactional features in other datasets.

6.6.4 Experiment Setup

We tune each baseline in parameter ranges centered at the author provided values for each dataset and set all embedding dimensions to 200 for uniformity. We split each state in each dataset into training (80%), validation (10%) and test (10%) sets for training, tuning and testing purposes. For the *implicit feedback* setting in *FT-Data*, we adopt the standard

Table 6.4: Comparing the expressivity aspects incorporated by baseline recommendation models against our proposed **MMT-Net** approach.

	Bi-Linear Pooling	**Multi-Linear Pooling**	**Low-Rank**	**Factor Weights**	**Θ(Context)**
NFM	Yes	No	No	No	Linear
AFM	Yes	No	No	Yes	Quadratic
AIN	No	No	Yes	Yes	Linear
FMT	No	No	Yes	Yes	Linear
MMT	Yes	Yes	Yes	Yes	Linear

Table 6.5: Percentage improvements (**% Hit-Rate@1**) on *FT-Data* target states with module transfer approaches and meta-learning baselines against training all modules on the target state directly as in Table 6.8.

Dataset		**Direct %H@1**	**Anneal %H@1**	**DRR %H@1**	**LWA %H@1**	**NLBA %H@1**	**s^2-Meta %H@1**
FT-Data	T_1	2%	**19%**	18%	6%	x	x
	T_2	0%	**16%**	**16%**	8%	x	x
	T_3	3%	**18%**	**18%**	6%	x	x
	T_4	-1%	**14%**	12%	11%	x	x

negative-sample evaluation [60] and draw one-hundred negatives per positive, equally split between item and context negatives similar to the training process in Section 6.4.4. We then evaluate the average **Hit-Rate@K** (**H@K**) metric for $K = 1, 5$ in Table 6.8, indicating if the positive sample was ranked highly among the negative samples. For the *explicit feedback* setting in the other two datasets, we follow the standard **RMSE** and **MAE** metrics in Table 6.7 [11, 137] (no negative samples required). All models were implemented with *Tensorflow* and tested on a *Nvidia Tesla V100 GPU*.

6.6.5 Transferring Modules to Sparse Target States

We evaluate module transfer methods by the percentage improvements in the **Hit-Rate@1** for the implicit feedback setting in *FT-Data* (Table 6.5), or the drop in **RMSE** (Table 6.6) for the explicit feedback datasets when we transfer the $\mathcal{M}^1, \mathcal{M}^3$ and $\mathcal{M}^4$ modules from the source state rather than training all four modules from scratch on that target domain. Similarly, meta-learning baselines were evaluated by comparing their joint meta-model performance on the target state against our model trained only on that state. The performance numbers for training our model on each target state without transfer are recorded in Ta-

Table 6.6: Percentage RMSE improvements on the Yelp and Google Local target states with module transfer approaches and meta-learning baselines against training all modules on the target state directly as in Table 6.7.

Dataset		Direct %RMSE	Anneal %RMSE	DRR %RMSE	LWA %RMSE	NLBA %RMSE	$\mathbf{s^2}$-Meta %RMSE
Yelp	T_1	-2.2%	**7.7%**	7.2%	2.6%	4.1%	3.7%
	T_2	-2.6%	**9.0%**	7.9%	1.8%	3.6%	3.1%
	T_3	0.8%	**8.5%**	8.1%	0.3%	5.3%	1.8%
Google Local	T_1	-1.2%	**11.2%**	11.0%	3.3%	4.3%	3.1%
	T_2	-1.7%	**12.1%**	10.9%	4.6%	4.9%	2.8%
	T_3	-2.0%	**9.6%**	8.8%	2.4%	6.3%	3.9%

ble 6.7, Table 6.8.

We could not scale the NLBA, LWA and **$\mathbf{s^2}$-Meta** approaches to *FT-Data* owing to the costs of training the meta-models on all users combined across the source and multiple target domains. In Table 6.6, we demonstrate the percentage reduction in RMSE with module transfer for Google Local, Yelp, and in Table 6.5, we demonstrate significant improvements in the hit-rates for *FT-Data*. We start with an analysis of the training process for module transfer with simulated annealing and DRR adaptation.

Transfer Details: On each target state in each dataset, all four modules of our MMT-Net model are pretrained over two gradient epochs on the target samples. The layers in modules $\mathcal{M}^1, \mathcal{M}^3$ and $\mathcal{M}^4$ are then replaced with those trained on the source state, while retaining module $\mathcal{M}^2$ without any changes (in our experiments $\mathcal{M}^2$ just contains user and item embeddings, but could also include neural layers if required). This is then followed by either simulated annealing or DRR adaptation of the transferred modules. We analyze the training loss curves in Section 6.6.7 to better understand the fast adaptation of the transferred modules.

Invariant Quality: A surprising result was the similar performance of *direct layer-transfer* with no adaptation to training all modules on the target state from scratch (Table 6.6). The transferred source state modules were directly applicable to the target state embeddings. This helps us validate the generalizability of context-based modules across independently trained state models even with no user or item overlap.

Computational Gains: We also plot the total training times including pretraining for DRR and annealing against the total number of target state interactions in Figure 6.5. On the target states, module transfer is 3x faster then training all the modules from scratch. On the whole, there is a significant reduction in the overall training time and computational

Table 6.7: We evaluate recommendation performance on each state (no transfer) with RMSE, MAE metrics for *explicit feedback* against the ground-truth ratings. Metrics were averaged over five runs, $*$ indicates statistical significance (paired *t-test*, p=0.05). On average, models incorporating both pooling and reweighting in Table 6.4 exhibit significant relative gains (i.e., **AFM**, **MMT**).

Dataset	**State**	**CAMF** [11]		**MTF** [83]		**NCF** [60]		**NFM** [58]	
		RMS	**MAE**	**RMS**	**MAE**	**RMS**	**MAE**	**RMS**	**MAE**
	S	1.21	0.94	1.13	0.87	1.18	1.04	1.02	0.83
Yelp	T_1	1.56	1.20	1.41	1.12	1.39	0.99	1.29	1.01
	T_2	1.33	1.04	1.36	0.98	1.26	1.02	1.19	1.05
	T_3	1.49	1.13	1.50	1.08	1.35	1.08	1.31	0.96
	S	1.36	1.01	1.21	0.90	1.04	0.89	0.80	0.73
Google	T_1	1.49	1.20	1.38	1.14	1.27	1.05	1.10	0.99
Local	T_2	1.37	1.16	1.31	1.20	1.36	1.17	1.21	1.05
	T_3	1.39	1.23	1.20	1.07	1.19	0.98	1.13	0.92
Dataset	**State**	**AFM** [225]		**AIN** [137]		**FMT-Net**		**MMT-Net**	
		RMS	**MAE**	**RMS**	**MAE**	**RMS**	**MAE**	**RMS**	**MAE**
Yelp	**S**	0.96	0.78	0.98	0.75	1.02	0.76	**0.94**	**0.73**
	T_1	1.27	0.94	1.36	0.91	1.34	0.95	**1.24**$*$	**0.88**$*$
	T_2	1.16	**0.90**	1.17	0.95	1.15	0.98	**1.13**$*$	0.91
	T_3	**1.20**$*$	0.93	1.25	0.98	1.29	1.02	**1.20**$*$	**0.89**$*$
	S	**0.77**	0.63	0.85	0.64	0.91	0.68	**0.77**	**0.64**
Google	T_1	0.94	0.85	1.22	0.90	1.31	0.96	**0.89**	**0.76**$*$
Local	T_2	**1.14**$*$	0.98	1.19	1.01	1.28	1.07	1.16	**0.93**$*$
	T_3	1.09	0.91	1.08	0.94	1.14	0.98	**1.02**$*$	**0.85**$*$

effort in the *one-to-many* setting. Simulated annealing and DRR adaptation converge in fewer epochs when applied to the pre-trained target model, and outperform the target-trained model by significant margins (Table 6.6). These computational gains potentially enable a finer target domain granularity (e.g., adapt to towns or counties rather than states).

6.6.6 Single Domain Recommendation

We draw attention to the most relevant features of the baselines and our variants in Table 6.4. We highlight our key observations from the experimental results obtained with the baseline recommenders and our FMT-Net and MMT-Net variants (Table 6.8, Table 6.7). Note that methods with some form of context pooling significantly outperform methods that do not consider pooled factors, indicating the importance of multi-linear model expressivity.

Table 6.8: We evaluate recommendation performance on each state (no transfer) with the *H@1, 5* metrics for *implicit feedback* in *FT-Data*. Metrics were averaged over five runs, * indicates statistical significance (paired *t-test*, p=0.05). On average, feature-pooling methods AFM, NFM and MMT outperform additive models AIN, FMT. x indicates timed-out or memory limit exceeded.

Dataset	**State**	**CAMF** [11]		**MTF** [83]		**NCF** [60]		**NFM** [58]	
		H@1	**H@5**	**H@1**	**H@5**	**H@1**	**H@5**	**H@1**	**H@5**
	S	x	x	x	x	0.42	0.77	0.52	0.91
FT-Data	T_1	x	x	x	x	0.36	0.71	0.41	0.83
	T_2	x	x	x	x	0.25	0.64	0.30	0.77
	T_3	x	x	x	x	0.26	0.70	0.31	0.78
	T_4	x	x	x	x	0.29	0.72	0.32	0.74

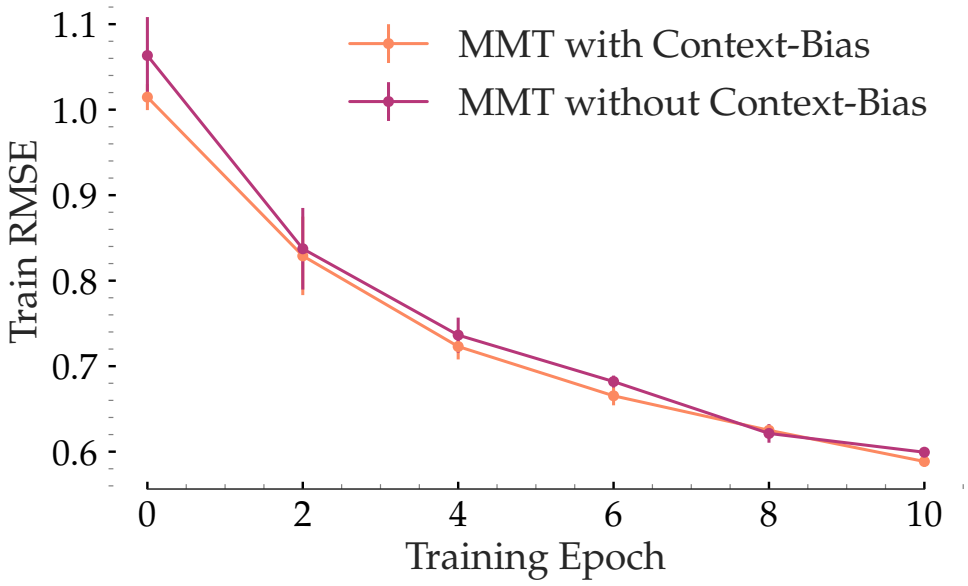
Dataset	**State**	**AFM** [225]		**AIN** [137]		**FMT-Net**		**MMT-Net**		**MMT-m**	
		H@1	**H@5**	**H@1**	**H@5**	**H@1**	**H@5**	**H@1**	**H@5**	**H@1**	**H@5**
	S	x	x	0.44	0.89	0.37	0.76	**0.56***	**0.94**	**0.56***	0.93
FT-Data	T_1	x	x	0.34	0.76	0.32	0.75	0.45	0.84	**0.47***	**0.86***
	T_2	x	x	0.30	0.72	0.26	0.72	**0.34***	**0.79**	**0.34***	0.77
	T_3	x	x	0.29	0.74	0.28	0.74	0.33	**0.82***	**0.34**	0.80
	T_4	x	x	0.32	0.78	0.21	0.69	0.37	0.80	**0.38**	**0.83***

Also observe that AFM performs very competitively owing to its ability to reweight terms similar to our approach (Table 6.7), but fails to scale to the larger *FT-Data*. NFM is linear with context size in practice owing to a simple algebraic re-organization, and thus scales to *FT-Data*, however losing the ability to reweight pairwise context product terms [58].

Also note the differences between our FMT and MMT variants, demonstrating the importance of the pooled multi-linear formulation for the contextual invariants. These performance differences are more pronounced in the implicit feedback setting (Table 6.8). This can be attributed to the greater relevance of transaction context (e.g., transactions provide accurate temporal features while review time is a proxy to the actual visit) and more context features in *FT-Data* vs. Google Local and Yelp (220 vs. 90,120 respectively), magnifying the importance of feature pooling for *FT-Data*.

The lack of pooled feature expressivity in the FMT-Net model impacts the training process as seen in Figure 6.4, demonstrating the importance of context intersection. The NFM and MMT models converge faster to a smaller Train-RMSE in Figure 6.4 and outperform FMT on the test data (Table 6.8, Table 6.7). We also observe models incorporating pooled factors to outperform the inherently linear attention-based AIN model, although the performance gap is less pronounced in the smaller review datasets (Table 6.7).

Figure 6.2: MMT-Net trained with & without context-bias (Equation (6.11)) on the Google Local source exhibits similar Train-RMSE, but registers $> 10\%$ drop in test performance.



We now qualitatively analyze our results to interpret module transfer/adaptation as well as our overall performance gains on the target domains.

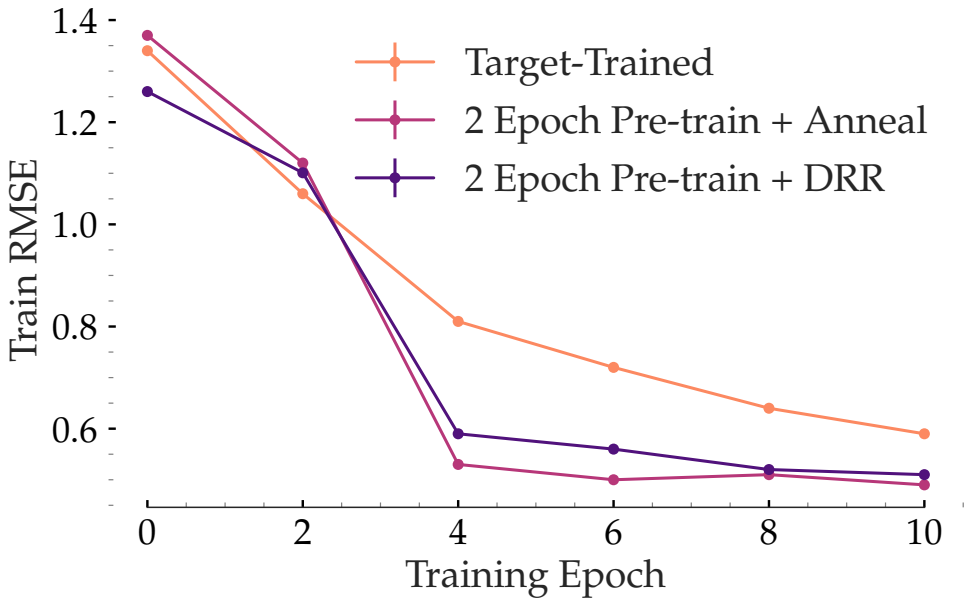
6.6.7 Qualitative Analysis

We analyze our model from the model training and convergence perspective for the module transfer adaptation methods. We observe consistent trends across the direct layer-transfer, annealing, and DRR adaptation approaches.

Training without Context-Bias: To understand the importance of decorrelating training samples in the training process, we repeat the performance analysis on our MMT-Net model with and without the adaptive context-bias term in the training objective in Section 6.4.4. We observe a 15% performance drop across the Yelp and Google Local datasets, although this does not reflect in the Train-RMSE convergence (Figure 6.2) of the two variations. In the absence of context-bias, the model overfits uninformative transactions to the user and item bias terms (s_u, s_v) in Equation (6.15), Equation (6.16) and thus achieves comparable Train-RMSE values. However, the overfit user and item terms are not generalizable, resulting in the observed drop in test performance.

Model Training and Convergence Analysis: We compare the Train-RMSE convergence for the MMT-Net model fitted from scratch to the Google Local target state, Colorado ($\mathbf{T}_1$) vs. the training curve under DRR and annealing adaptation with two pre-

Figure 6.3: MMT-Net convergence under target-training vs. Annealing/DRR adaptation after 2 epochs of pretraining on the Google Local Colorado target.



training epochs on the target state in Figure 6.3. Clearly, the target-trained model takes significantly longer to converge to a stable Train-RMSE in comparison to the Anneal and DRR adaptation.

Although the final Train-RMSE is comparable (Figure 6.5), there is a significant performance difference between the two approaches on the test dataset, as observed in Table 6.6. Training loss convergence alone is not indicative of the final model performance; the target-only training method observes lower Train-RMSE by overfitting to the sparse data. We also compare the Train-RMSE convergence for target-trained models with and without pooled context factors (MMT-Net, NFM vs. FMT-Net) in Figure 6.4. We observe the NFM, MMT-Net models to converge faster to a better optimization minima than FMT-Net. This also reflects in their test performance in Table 6.8.

6.6.8 Scalability and Robustness Analysis

We demonstrate the scalability of our two transfer learning algorithms (*simulated annealing* and *distributionally regularized residuals*) with the number of transactions in the target domain in Figure 6.5 (i.e., transferring a pre-trained source model with the respective algorithms) against training separate models for the source and target domains.

Our previous observations in Section 6.6.5 also validate the ability of our approach to scale deeper neural recommendation architectures to a large number of target domains while also

Figure 6.4: MMT-Net convergence compared to NFM and FMT-Net on the Google Local Colorado target.

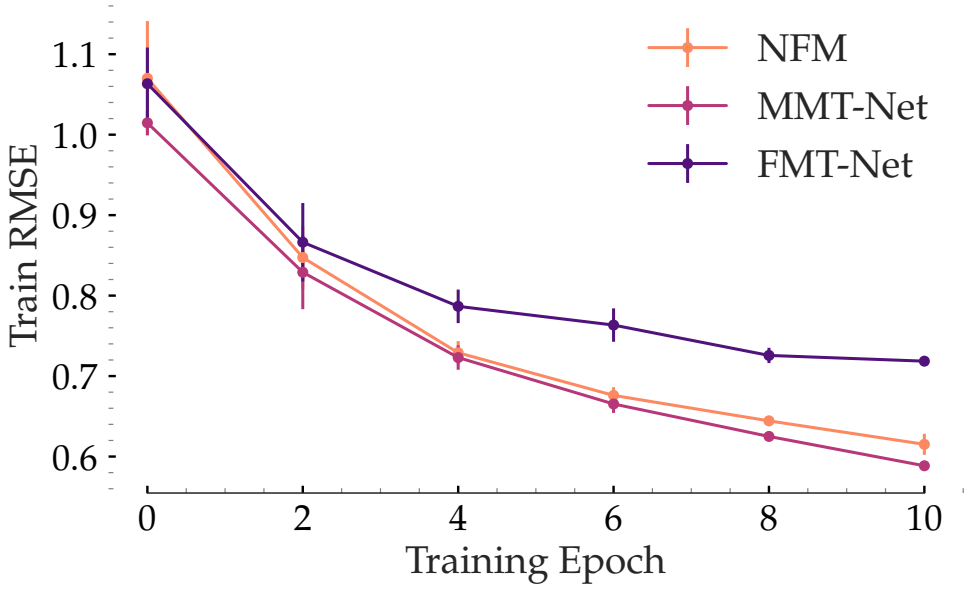


Table 6.9: MMT-Net performance degradation was measured by the decrease in **H@1** or increase in **RMSE**, averaged over target states with random context feature dropout.

Context Drop	**5%**	**10%**	**15%**	**20%**
FT-Data	1.1%	2.6%	4.1%	6.0%
Google Local	3.9%	4.2%	7.0%	8.8%
Yelp	1.8%	3.2%	5.4%	7.3%

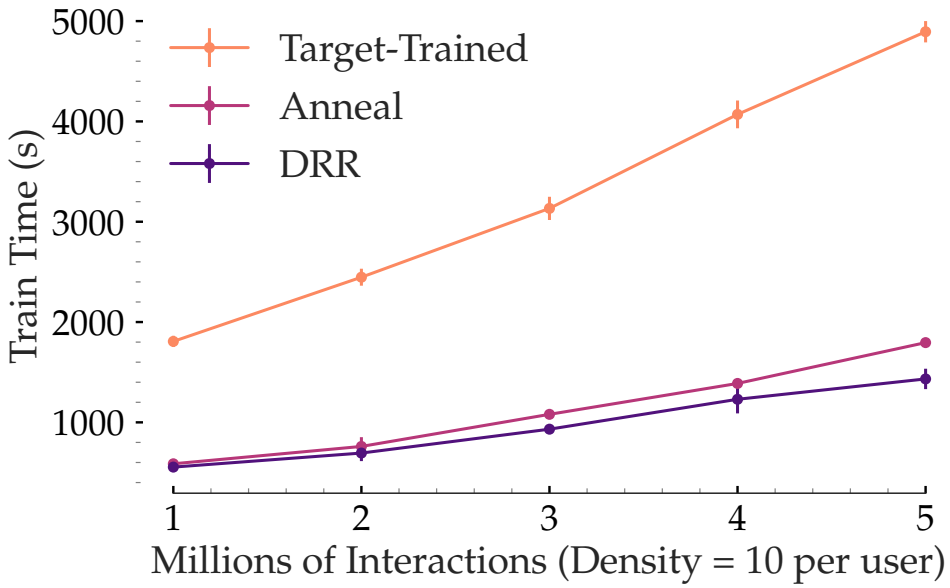
enabling a finer resolution for the selection of target domains.

Towards tackling incomplete or potentially incorrect context feature data, we also evaluated the robustness of the shared context layers by randomly dropping up to 20% of the context features in each interaction at train and test time for both, the source and target states in Table 6.9.

6.6.9 Limitations and Discussion

We identify a few fundamental limitations of our model. While our approach presents a scalable and effective solution to bridge the weaknesses of gradient-based meta learning and co-clustering via user or item overlaps, contextual invariants do not extend to cold-start users or items. Second, our model does not trivially extend to the case where a significant number of users or items are shared across recommendation domains. We separate the

Figure 6.5: MMT-Net training duration with and without module transfer vs. target domain interaction volume.



embeddings and *learn-to-learn* aspect which improves modularity, but prevents direct reuse of representations across domains, since only the transformation layers are shared. Depending on the application, context features could potentially be filtered to enhance social inference and prevent loss of diversity in the generated recommendations.

6.7 CONCLUSION AND NEXT STEPS

6.7.1 Chapter Summary

This chapter developed a novel contextual invariant approach to address the sparsity problem in the cross-domain setting via neural model transfer. We leverage the broad meta-transfer paradigm grounded on an expressive context pooling strategy to learn effective invariants. The invariants themselves and the resulting soft clusters of users and items constitute the set of meta-parameters that enable cross-domain learning. Further, we develop two complementary approaches (*parameter updates via annealing* vs. *distributional input adaptation*) to optimize the transferred neural models and parameters to each sparse recommendation domain.

Our context-invariant approach is highly scalable in the one-to-many setting, especially when combined with the residual adaptation strategy. We incur minimal residual parameter overheads and reduced training costs for each new target domain compared to fitting a new

model, both of which constitute significant advantages over gradient-based meta-learning approaches. We contrast the two adaptation strategies that broadly characterize neural layer transfer - *input adaptation* and *parameter adaptation* - and exhibit the effectiveness of both strategies while highlighting the advantages of reducing parameter overheads.

6.7.2 Improvements to the Proposed Framework

The proposed framework hinges on the availability of similar contextual features across the source and each target domain of interest. While distributional changes are accounted for by the input adaptation strategy, we expect the essential recommendation context to overlap.

We consider two potential approaches to overcome the above limitation and extend the learned models' transferability in our framework. First, we aim to develop meta-invariants associated with each context feature, e.g., statistical representations, induced gradient representations associated with each context feature. Second, apply the residual adaptation strategy to the meta-invariants to account for the contextual invariants' higher-order characteristics.

A few valuable extensions also include updating representation with streaming data and incorporating knowledge priors on expected behavior patterns (e.g., if we knew what combinations of context are more likely to dictate interactions) to benefit the learned context transformation space.

6.7.3 From Multi-Domain Single-Task to Single-Domain Multi-Task Invariants

In combination, the preceding chapters handled the sparsity and skew challenge in unimodal, multimodal, and cross-domain settings and can be applied simultaneously towards a joint recommendation or user inference goal. However, we identify a significant avenue for progress with the proposed methods - that of *cross-task similarities*. While the preceding methods offer grouping and representation strategies incorporating users, items, datamodalities, interaction context, and domains, they do not directly leverage cross-task correlations.

Task-correlations are often platform-dependent, i.e., the set of inference and recommendation tasks that a recommendation platform offers to or infers across its user and item populations. Prior work usually handles each task in isolations by developing predictive models that encode specific aspects of the task in the form of *inductive biases*. While each task-model's exact inductive nature might differ, the tasks' outcomes are often correlated due

to users' aligned behavioral propensities and thus benefit from a joint treatment. Towards this goal, we describe a domain-agnostic generalizable solution to leverage shared aspects across multiple predictions, inference, and recommendation tasks to mitigate skewed and sparse behavioral data in the next chapter.

While Chapter 6 handles the multi-domain, single-task setting, in the next chapter, we handle the complementary single-domain, multi-task scenario.

CHAPTER 7: RESIDUAL-AUGMENTED KNOWLEDGE REPRESENTATIONS FOR MULTI-TASK RECOMMENDATION AND INFERENCE

This chapter proposes MuTATE, a Multi-Task Augmented paradigm to learn Transferrable Embeddings of knowledge graphs. Prior research efforts in knowledge graph representations assume that a given knowledge graph is complete and apply it to augment machine learning models; or apply geometric, relation-based, and path-based hypotheses to enrich the knowledge graph and learn informative node embeddings.

In contrast to these efforts, we propose a novel bidirectional framework to unify model training with knowledge graph completion and enrichment. We unify diverse task-models that predict associations between distinct subsets of nodes in the knowledge graph vis-a-vis an underlying shared node embedding space, thus permitting multi-directional knowledge transfer: model→graph, graph→model, and model→model. We achieve this by learning task-specific residual functions to augment the node embeddings, motivated by counterfactual domain-shift theory. With experiments on two public datasets, we show strong results on knowledge graph link prediction (5% relative to *state-of-the-art* embedding baselines). We show significant potential for the above types of knowledge transfer across distinct task-models.

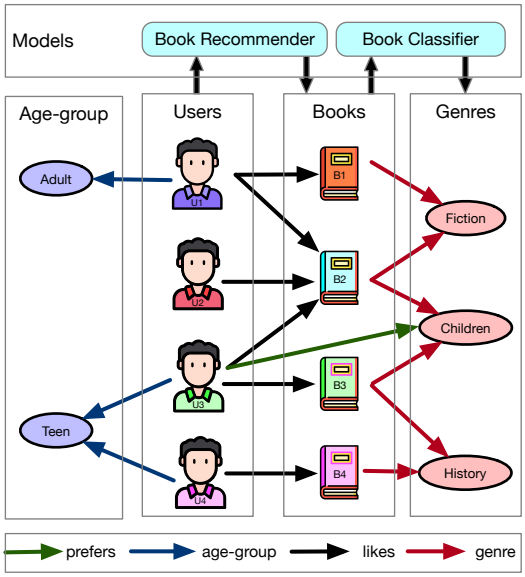
7.1 INTRODUCTION

The modern-day rise of context-driven AI has sparked a renewed interest in knowledge graph (KG) representations of data. Knowledge graphs represent vast amounts of domain-specific information (ranging from linguistics [220], biomedicine [42] to finance [27]) via interacting entities (nodes) and relationships (edges)—see Figure 7.1. Knowledge graphs are semantically enriched by the rich transitive association structure across diverse interacting entities, hence enhancing a wide range of inferencing applications, e.g., intelligent assistants (*Apple Siri*, *Amazon Alexa*), question answering on search engines (*Google, Microsoft Bing entity graphs*), and product recommendation/discovery on online marketplaces (*Amazon, eBay*).

Figure 7.1 demonstrates a sample knowledge graph capturing user and book attributes, e.g., *age-group,* genre, and their relationships. The knowledge graph is impacted by the characteristics of the underlying data, such as sparsity for some users or missing attributes), and distributional skew in their relations (e.g., most readers prefer a popular genre).

We also demonstrate two task-models in Figure 7.1, book recommendations to users,

Figure 7.1: Toy example of a user-item knowledge graph with four interacting entity types: *users*, *books*, *age-groups*, and *book-genres*. Entities are linked via four relations: user *prefers* genre, user *in* age-group, user *likes* book, and book *is genre*. *Sample task-models include recommendation (*Book Recommender*) and book-genre prediction (*Book Classifier*).*



and book genre prediction. Both models can be trained with the available factual links in the graph, i.e., graph→model knowledge transfer. However, once trained, the specialized inductive biases of these task-models generate better link predictions compared to graph-based heuristics and can thus densify the sparse/skewed neighborhoods in the knowledge graph [96, 110]. However, different task-models may generate *contradicting or aligned predictions* depending on their respective *inductive biases*. We thus aim to develop a unified framework to simultaneously facilitate knowledge graph enrichment and task-model training with minimal restricting assumptions on each task-model. We now categorize prior research into three distinct directions and describe our innovations.

The first is knowledge graph embedding [18, 193, 219], which attempts to enrich the knowledge graph and incorporate latent structural proximities of nodes by transitively learning a heuristic or path-based patterns such as *symmetry*, *anti-symmetry*, *composition* and *analogy* (formally described in Section 7.3.1). However, these patterns do not distinguish entity or relation types and apply equivalently to all of them. This leads to contradictory and incorrect inferences violating domain knowledge. In our toy example, a user may like a book, but not necessarily the book's broad genre. A task-model specifically designed for book genre preference avoids such incorrect transitive inferences owing to its inductive design/bias. Thus, our approach adopts patterns as a first-cut solution, but subsequently *enhances the embeddings*

using *task-model feedback*.

The second direction views the knowledge graph as an oracle to guide task-models [71, 216] by leveraging the connectivity patterns in different relation types. However, the knowledge graph's view is not optimized to the architecture or training objective of each task-model. In contrast, we specialize the graph to each task-model with a *task-specific residual function added to the knowledge graph embeddings*, motivated by counterfactual domain-shift theory. This enables simultaneous bidirectional updates across distinct task-models and the knowledge graph (Section 7.5.1).

A third recent direction includes hybrid solutions that combine task augmentation and graph enrichment [21, 54] under specific architectural assumptions or external feedback. They do not extend to the broader multi-task setting, where distinct task-models involve overlapping nodes in the graph. In contrast, *we make no architectural assumptions* and, in theory, incorporate any gradient-updated task-model across different entities and relations in the knowledge graph.

We achieve the above objectives by viewing each task-model as an intervention or treatment to the respective entity sets. In Figure 7.1, when we use the *Book Recommender* to recommend books for the user U3, we obtain B4 as a recommended book. This recommendation enables us to create a new counterfactual link between the user U3 and the book B4—see Figure 7.2, as opposed to factual links which exist in the graph. Unlike factual links, counterfactual links are biased by the nature of the task-model which generates them. Hence, we pose the causal inference question [106, 145] of whether the counterfactual link (e.g., Recommendation U3 $\rightarrow$ B4) originates purely from the task-model eccentricity or if it indicates the existence of a link in the knowledge graph. This leads us to the following questions:

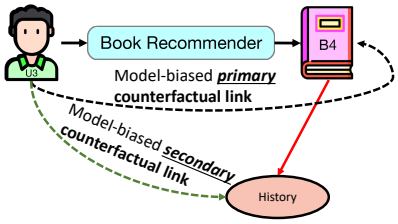
Q1: Given the task-model biased counterfactual links, can we infer the missing factual links in the knowledge graph?

Q2: Conversely, given the factual links in the graph, what are the likely counterfactual links predicted by a specific task-model?

While the answer to **Q1** enables us to enrich the knowledge graph, **Q2** improves the task-model by providing a task-specific view of the factual links. We learn these forward and reverse transformations with task-specific residual functions to enable bidirectional knowledge transfer between each task-model and the knowledge graph embeddings. In summary, our contributions are:

Merging Multi-Task Learning and Knowledge Graph Enrichment/Embedding: We propose a holistic view of knowledge graphs and multi-task learning that permits the bidirectional transfer of knowledge between domain-specific knowledge graphs and task-

Figure 7.2: We use the *Book Recommender* model to infer counterfactual edges (shown using dotted lines) to enrich the KG. Primary counterfactual links are inferred directly from the model. Secondary counterfactual links connect the one-hop neighbors of the primary links.



models. This holistic view overcomes the limiting assumptions of past work that restrict the direction or type of knowledge transfer. While enabling bidirectional knowledge transfer, we also avoid assumptions about the nature of the specific task-models, architectures, or objectives.

Generalizability: The proposed framework is highly generalizable; we make no assumptions about the data-domain or the task-models. As a result, we can integrate diverse tasks and model architectures with the same underlying knowledge graph embeddings. In our experiments, we exhibit this capability with two distinct models, a recommendation model connecting users and items and an item-content model that predicts likely words in item descriptions. We show counterfactual updates from the word-prediction model can significantly improve the recommendation model for sparse users (*item-word links are leveraged to improve user-item links*).

Modeling Multi-Task Embedding Updates via Residuals: We identify the connection between multi-task knowledge graph updates and covariate domain-shift theory [80], which permits us to model different task-specific distributions with the same underlying knowledge graph via residual function learning in a very inexpensive manner.

Strong Experimental Results: We demonstrate strong experimental results with knowledge graphs constructed from two large distinct datasets, the *Google Local Reviews Dataset* [1][57, 151] and the *Yelp Challenge Dataset* [2]. We show how to leverage two very different task-models, word2vec [139] and a context-aware recommender [98] to densify and improve the knowledge graph, and also simultaneously perform model-to-model cross-training (i.e., use the first model to update the graph, which can then improve the second model). On the whole, we show strong results on graph completion (5% relative to the state-of-the-art embedding baselines) and show significant potential for knowledge transfer.

[1] `http://cseweb.ucsd.edu/~jmcauley/datasets.html`
[2] https://www.yelp.com/dataset/challenge

We now summarize related work, formalize our problem, describe our solution, and evaluate the proposed framework.

7.2 PROBLEM DEFINITION

In this section, we present the distinct components associated with our knowledge graph and the associated task-models, similar to the examples in Figure 7.2.

Knowledge Graph Notations: We consider a heterogeneous directed knowledge graph with multiple types of entities (i.e., types of nodes) and relations. Let us represent entity types as:

$$\mathbf{E}_1 \text{ (e.g., users)}, \mathbf{E}_2 \text{ (e.g., items)} \cdots \mathbf{E}_{|\mathcal{E}|} \tag{7.1}$$

where $\mathcal{E} = \{\mathbf{E}_1, \mathbf{E}_2 \cdots \mathbf{E}_{|\mathcal{E}|}\}$ is the set of all entity types. The set of all nodes in the graph is $\cup \mathbf{E}_i$. Let $\mathcal{R} = \{\mathbf{R}_1, \mathbf{R}_2 \cdots \mathbf{R}_{|\mathcal{R}|}\}$ denote the set of relations where each relation $\mathbf{R}_r : \mathbf{E}_1(r) \rightarrow \mathbf{E}_2(r)$ is a collection of links between head and tail entity sets $\mathbf{E}_1(r), \mathbf{E}_2(r) \in \mathcal{E}$ (In Figure 7.1, the relation $\mathbf{R}_{prefers} : \mathbf{E}_{users} \rightarrow \mathbf{E}_{books}$). Note that multiple relations can exist between entity sets (e.g., users *prefer* books, users *dislike* books).

Factual Links: Factual links exist apriori in the knowledge graph, in contrast to counterfactual links suggested by task-models. We denote each factual link as (e_1, r, e_2) where $e_1 \in \mathbf{E}_1(r)$, $e_2 \in \mathbf{E}_2(r)$ are the head and tail entities, and r is their relation. $\vec{\mathbf{e}}_1, \vec{\mathbf{e}}_2$ denote the respective d-dimensional entity embeddings. Each relation r is described by head and tail projectors $(\vec{\mathbf{p}}_1(r), \vec{\mathbf{p}}_2(r))$, which have the same dimensionality as the entity embeddings.

Task-Model Notations: For simplicity, we only consider discrete bimodal one-to-one prediction tasks j between entity sets $\mathbf{E}_1(j), \mathbf{E}_2(j) \in \mathcal{E}$ in our analysis. However, regression tasks and multivariate tasks can be discretized or factored to fit a similar abstraction. We consider task-models $\mathcal{M}(j)$ (for task j) to connect input entity $e_1 \in \mathbf{E}_1(j)$ to a predicted output entity $e_2 \in \mathbf{E}_2(j)$, thus inducing counterfactual links $(e_1(j), e_2(j))$ between the entity sets $\mathbf{E}_1(j)$, $\mathbf{E}_2(j) \in \mathcal{E}$, depending on its prediction task (e.g., the book recommendation model in Figure 7.2 induces counterfactual user to book links). Note that the task-model $\mathcal{M}(j)$ leverages the factual links between the same entity sets as training data.

Counterfactual Links: Task model $\mathcal{M}(j) : \mathbf{E}_1(j) \rightarrow \mathbf{E}_2(j)$ induces primary counterfactual links of the form $(e_1(j), e_2(j))$ by predicting $e_2(j) \in \mathbf{E}_2(j)$ as the output for the input entity $e_1(j) \in \mathbf{E}_1(j)$. Also note that unlike the factual links (e_1, r, e_2), we do not have a relation label for the counterfactual link ($e_1(j)$, $e_2(j)$). Further, as described in Figure 7.2, the task-model also induces secondary counterfactual links of the form $(e_1(j), e')$ or $(e'', e_2(j))$

Table 7.1: Notation Description Table.

Symbol	Description
$\mathbf{E}_i$	Entity set (i.e., specific type of entities/nodes)
$\mathcal{E}$	Set of all entity sets
$e_i, \vec{\mathbf{e}}_i$	Entity $e_i \in \mathbf{E}_i$ and its embedding vector
$\mathbf{R}_r : \mathbf{E}_1(r) \to \mathbf{E}_2(r)$	Relation between entity sets $\mathbf{E}_1(r)$ and $\mathbf{E}_2(r)$
$(\vec{\mathbf{p}}_1(r), \vec{\mathbf{p}}_2(r))$	Head and tail projectors for relation $\mathbf{R}_r$
(e_1, r, e_2)	Factual link between head entity $e_1 \in \mathbf{E}_1(r)$ and tail entity $e_2 \in \mathbf{E}_2(r)$
j	Prediction-task j linking entity set $\mathbf{E}_1(j)$ to $\mathbf{E}_2(j) \in \mathcal{E}$
$\mathcal{M}(j)$	Task-model for prediction-task j
$e_1(j)$	Input entity $e_1(j) \in \mathbf{E}_1(j)$ to task-model $\mathcal{M}(j)$
$e_2(j)$	Output $e_2(j) \in \mathbf{E}_2(j)$ from $\mathcal{M}(j)$ for input $e_1(j)$
$(e_1(j), e_2(j))$	Primary counterfactual link predicted by $\mathcal{M}(j)$

by connecting $e_1(j)$ and $e_2(j)$ to each other's one-hop neighbors.

We summarize the above notations in Table 7.1.

7.3 KNOWLEDGE GRAPH EMBEDDINGS

This section describes a scalable and expressive approach to embed the factual links in the knowledge graph. Then, we bi-directionally integrate task-models over the learned embedding space by fitting counterfactual residual functions.

7.3.1 Factual Link Embedding Model

Knowledge graph embedding techniques encode heuristic connectivity pattens in their embedding objectives [193], such as *symmetry/antisymmetry*, *composition*, and *inversion*, which can be stacked to encode higher-order linking patterns. We also note that *analogy* can be encoded in the embedding space, as in distributional word embeddings:

- **Symmetry:** $(e_1, r_a, e_2) \implies (e_2, r_a, e_1)$
- **Anti-Symmetry:** $(e_1, r_a, e_2) \implies \textit{not}\ (e_2, r_a, e_1)$
- **Inversion:** $(e_1, r_a, e_2) \implies (e_2, r_b, e_1)$
- **Composition:** $(e_1, r_a, e_2)\ \textit{and}\ (e_2, r_b, e_3) \implies (e_1, r_c, e_3)$

- **Analogy:** (e_1, r_a, e_2) *and* $(e_3, r_a, e_4) \implies (e_1, r_b, e_3)$ / (e_2, r_c, e_4)

While these patterns enable first-cut link selection, they do not distinguish the different entity and relation types, leading to incorrect inferences. Prior embedding methods do not provide mechanisms to address the *spurious inference challenge*.

Our fundamental hypothesis is that leveraging task-models designed for a specific prediction task can help filter the encoded patterns. Further, in a heterogenous knowledge graph, the degree of sparsity may not be evenly spread across the different node and relation modalities. Thus, cross-modal transfer is particularly important in any enrichment or completion effort, i.e.,

- How do we leverage (e_1, r_a, e_2) for link predictions of the form (e_1, r', e'), (e_2, r', e'), (e'', r'', e_1), (e'', r'', e_2)?

Note that the answer to the above form of cross-modal learning is specific to the relation types r_a, r', r'' as well the entity nodes e_1 and e_2, and thus can be answered effectively by leveraging task-models $\mathcal{M}(j)$ involving either these entities or relations.

In addition to these properties, paralellizable embedding learning is critical for knowledge graph applications, owing to their scale. DistMult [228] is widely applied due to its simple block-optimizable form [102]. The basic DistMult model follows a bilinear function with a learned diagonal projector matrix ($\mathbf{P}_r$) representing the relation type r. Thus the likelihood of an edge (e_1, r, e_2) is given by:

$$\mathcal{L}(\vec{\mathbf{e}}_1, r, \vec{\mathbf{e}}_2) = \vec{\mathbf{e}}_1^T \mathbf{P}_r \vec{\mathbf{e}}_2 \tag{7.2}$$

However, due to the symmetric nature of the above transformation, anti-symmetry and inversion are hard to encode in this form [193]. On the other hand, other methods that do not have a symmetric objective wrt. the head and tail entities (e.g., Rotate [193]) pose block optimization constraints [102].

To overcome these limitations, we apply an inexpensive modification to Equation (7.2). We break the symmetry in Equation (7.2) by describing a head and tail dual-projector form for each relation. Note that this form only involves a few additional parameters, namely twice as many parameters for the relation embeddings. However, in most knowledge graphs, the number of relation-types are several orders of magnitude fewer than the number of nodes so that this parameter overhead is negligible. We define the likelihood of an edge (e_1, r, e_2) as (where *sim* is the cosine-similarity):

$$\mathcal{L}(\vec{\mathbf{e}}_1, r, \vec{\mathbf{e}}_2) = sim\Big(\vec{\mathbf{e}}_1 \otimes \vec{\mathbf{p}}_1(r),\ \vec{\mathbf{e}}_2 \otimes \vec{\mathbf{p}}_2(r)\Big) \tag{7.3}$$

The above modification enables composition, inversion, and anti-symmetry:

- **Anti-Symmetry:** Consider relations r_a to be anti-symmetric, so that, $(e_1, r_a, e_2) \implies$ *not* (e_2, r_a, e_1) We can encode this in our likelihood term with orthagonal projectors for the head and tail, i.e., $\vec{\mathbf{p}}_1(r) \perp \vec{\mathbf{p}}_2(r)$ so that we take the orthagonal projections of the head and tail entity when the direction of the relation is reversed.

- **Inversion:** Consider relations r_a, r_b to be inversions of each other, so that, $(e_1, r_a, e_2) \implies (e_2, r_b, e_1)$ We can encode this in our likelihood term by switching the head and tail projectors, i.e., $\vec{\mathbf{p}}_1(r_a) = \vec{\mathbf{p}}_2(r_b)$ and $\vec{\mathbf{p}}_2(r_a) = \vec{\mathbf{p}}_1(r_b)$. It is easy to verify that this would result in $\mathcal{L}(\vec{\mathbf{e}}_1, r_a, \vec{\mathbf{e}}_2) = \mathcal{L}(\vec{\mathbf{e}}_2, r_b, \vec{\mathbf{e}}_1)$ which results in the desired inversion.

- **Composition:** Consider relations r_c to be composed of r_a and r_b, so that,

$$(e_1, r_a, e_2) \; \textit{and} \; (e_2, r_b, e_3) \implies (e_1, r_c, e_3) \tag{7.4}$$

  i.e., r_c is a sequential composition of two other relations. We can encode sequential compositions in our likelihood terms with the following simple switch, i.e., $\vec{\mathbf{p}}_1(r_c) = \vec{\mathbf{p}}_1(r_a)$ and $\vec{\mathbf{p}}_2(r_c) = \vec{\mathbf{p}}_2(r_a)$. This would transitively align the composed relation with the head and tail entities e_1 and e_3.

Finally, we introduce a identity-matrix scaling factor to retain proportion s of the embedding dimensions in the projected versions:

$$\mathcal{L}(\,\vec{\mathbf{e}}_1, r, \vec{\mathbf{e}}_2\,) = sim\Big(\,\vec{\mathbf{e}}_1 \otimes (\vec{\mathbf{p}}_1(r) + s\mathbb{I})\,, \;\; \vec{\mathbf{e}}_2 \otimes (\vec{\mathbf{p}}_2(r) + s\mathbb{I})\,\Big) \tag{7.5}$$

While the notion of head and tail projectors is also present in the TransD [73] model, our similarity function, which is just a dot product, enables block-sampling and optimization advantages. As a result, our model is scalable with the block optimizations proposed by Lerer et al. [102] and sufficiently expressive to integrate diverse task-models bidirectionally.

7.4 MULTI-TASK AUGEMENTATION VIA COUNTERFACTUAL LINKS

Consider a prediction task j to link input entities $e_1(j) \in \mathbf{E}_1(j)$ to $e_2(j) \in \mathbf{E}_2(j)$. As an example, the book recommendation model (i.e., model $\mathcal{M}(j)$ in Figure 7.1) links users to books. Each prediction of model $\mathcal{M}(j)$ creates a counterfactual link across the two entity sets $\mathbf{E}_1(j)$, $\mathbf{E}_2(j) \in \mathcal{E}$.

Note that the specific prediction task j modeled by $\mathcal{M}(j)$ may vary even between the same pair of entity sets $\mathbf{E}_1(j), \mathbf{E}_2(j) \in \mathcal{E}$. For instance, in our toy example in Figure 7.1, we could train a preferred book recommendation model to connect users to the books they like and a dislike-prediction model to connect users to books they disklike. These two models produce different task-biased counterfactual links between users and books since they have different objective functions. In this manner, each $\mathcal{M}(j)$ generates a different counterfactual link distribution across entity sets, depending on its inductive bias and task objective.

7.4.1 Viewing Task-Models as Interventions

Our key insight is to consider each task-model as an intervention on a specific subset of nodes in the knowledge graph, analogous to a medical treatment applied to a patient [80]. Note that the intervention depends on both the task (or objective) of the trained model and the model architecture, i.e., its inductive bias. Our key objective in the rest of this section is to develop a consistent pathway to densify the knowledge graph with the task-model biased counterfactual links, and conversely, enhance the model performance using the factual links in the knowledge graph.

We learn to encode task-model biases as *counterfactual residual functions of the node embeddings*, motivated by covariate domain-shift theory [80, 133] to correct for the distributional biases introduced by each task-model.

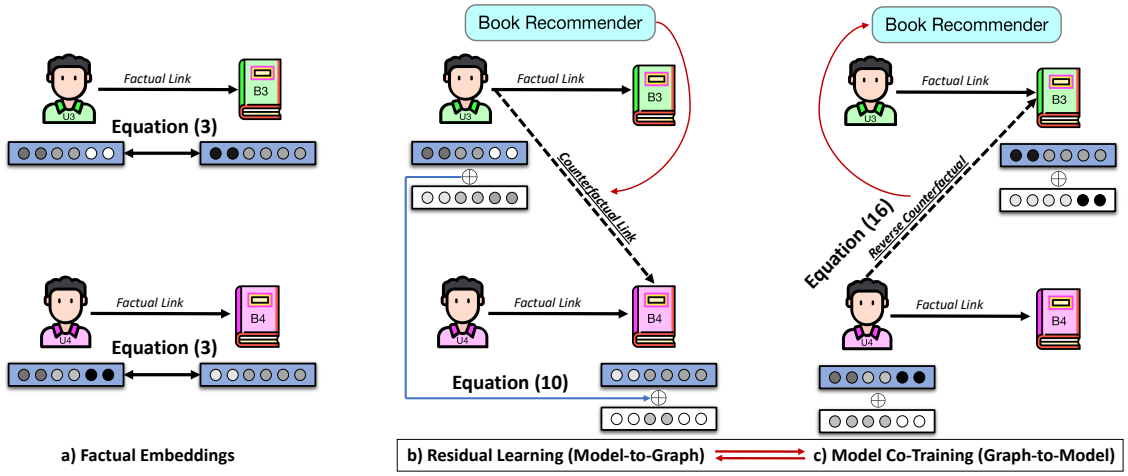
7.4.2 Model-Biased Counterfactual Links

We refer to the links directly predicted by the task models as the primary counterfactual links (as illustrated in Figure 7.2).

If $\mathcal{M}(j)$ predicts output $e_2(j) \in \mathbf{E}_2(j)$ for the input $e_1(j) \in \mathbf{E}_1(j)$, then $(\, e_1(j),\, e_2(j)\,)$ is the *primary counterfactual link*. Note that we do not know the relation label for counterfactual links. Also consider the one-hop neighbors $\mathcal{N}1\,(\, e_1(j)\,)$ and $\mathcal{N}1\,(\, e_2(j)\,)$. These neighbors nodes may belong to entity sets different from $\mathbf{E}_1(j)$ and $\mathbf{E}_2(j)$. We create cross-modal links of the form $(e_1(j), e')$, where $e' \in \mathcal{N}1\,(\, e_2(j)\,)$ and conversely, $(e'', e_2(j))$, where $e'' \in \mathcal{N}1\,(\, e_1(j)\,)$. We refer to these links as the *secondary counterfactual links*.

Thus, each task-model $\mathcal{M}(j)$ induces primary and secondary counterfactual links for each focus entity $e_1(j) \in \mathbf{E}_1(j)$, which we can leverage to update the embeddings $\vec{\mathbf{e}}_1(j)$ or learn the linking biases introduced by $\mathcal{M}(j)$.

Figure 7.3: (a) We learn the base entity embeddings via Equation (7.5) over factual links, (b) we then generate counterfactual links with the *Book Recommender* model to train the residual functions with Equation (7.13), (c) and improve the *Book Recommender* by reversing the learned residuals from step (b) in Equation (7.19). Note that steps (b), (c) can be iteratively optimized.



7.4.3 Counterfactual Link Likelihood

In this subsection, we describe how to estimate the likelihoods of counterfactual links as a function of the entity embeddings, so that we can backpropagate gradient updates. Let us consider the factual links of a focus entity $\mathbf{e}_1 \in \mathcal{E}_1$. Under our base embedding model in Equation (7.5), the likelihood of a factual link (e_1, r, e_2) is given by:

$$\mathcal{L}(\vec{\mathbf{e}}_1, r, \vec{\mathbf{e}}_2) = sim\Big(\vec{\mathbf{e}}_1 \otimes (\vec{\mathbf{p}}_1(r) + s\mathbb{I}) \,,\;\; \vec{\mathbf{e}}_2 \otimes (\vec{\mathbf{p}}_2(r) + s\mathbb{I}) \Big) \tag{7.6}$$

where the embedding vectors $\vec{\mathbf{e}}_1$, $\vec{\mathbf{e}}_2$ receive gradient updates to maximize the above likelihood term.

Unlike the factual links, a counterfactual link $(\, e_1(j),\, e_2(j)\,)$ generated by task-model $\mathcal{M}(j)$ does not have a relation type assigned to it. We consider three heuristic likelihood functions for the counterfactual links:

Relation-Agnostic (RA) Counterfactual Likelihood is computed as follows:

$$\mathcal{LRA}_{CF}\,(\,\vec{\mathbf{e}}_1(j),\, \vec{\mathbf{e}}_2(j)\,) = \boldsymbol{\sigma}\Big(\; sim(\;\vec{\mathbf{e}}_1(j),\; \vec{\mathbf{e}}_2(j)\;)\;\Big) \tag{7.7}$$

where σ denotes a suitable likelihood function, such as the sigmoid function.

The intuition of $\mathcal{LRA}_{CF}$ is to maximize the dimensions along which the two entity embeddings match. This strategy effectively increases the likelihood of any valid relation-type

between the entity pair depending on the projection component, except if the different relations across the entities are anti-correlated.

Preferred-Relation (PR) Counterfactual Likelihood is computed as follows:

$$\mathcal{LPR}_{CF}\left(\vec{\mathbf{e}}_1(j),\, \vec{\mathbf{e}}_2(j)\right) = \underset{r:\mathbf{E}_1(j)\rightarrow\mathbf{E}_2(j)}{\text{argmax}}\ \boldsymbol{\sigma}\Bigg(sim\Big(\vec{\mathbf{e}}_1(j) \otimes \vec{\mathbf{p}}_1(r),\ \vec{\mathbf{e}}_2(j) \otimes \vec{\mathbf{p}}_2(r) \Big) \Bigg) \tag{7.8}$$

$\mathcal{LPR}_{CF}$ only considers the most likely relation for any pair of entities in the likelihood estimation. This formulation is more reliable than $\mathcal{LRA}_{CF}$ for entity sets that have anti-correlated relations between them (e.g.,*user likes/dislikes book* in Figure 7.1).

Relation-Sum (RS) Counterfactual Likelihood is computed as follows:

$$\mathcal{LRS}_{CF}\left(\vec{\mathbf{e}}_1(j),\, \vec{\mathbf{e}}_2(j)\right) = \sum_{r:\mathbf{E}_1(j)\rightarrow\mathbf{E}_2(j)}\ \boldsymbol{\sigma}\Bigg(sim\Big(\vec{\mathbf{e}}_1(j) \otimes \vec{\mathbf{p}}_1(r),\ \vec{\mathbf{e}}_2(j) \otimes \vec{\mathbf{p}}_2(r) \Big) \Bigg) \tag{7.9}$$

$\mathcal{LRS}_{CF}$ amortizes the gradients across all relations between the pair of entity sets $\mathbf{E}_1(j)$, $\mathbf{E}_2(j)$.

However, more fundamentally, all the above likelihoods directly backpropagate gradient updates from the counterfactual links to the node embeddings. These updates may be incompatible with the factual links if the task-models learn different aspects of the underlying entities. In such a case, the counterfactual likelihoods in Equation (7.7), Equation (7.8) or Equation (7.9) must account for the biases introduced by each task-model. We view these biases as distributional shifts on the node embeddings obtained from the factual links via Equation (7.5). This view is grounded in the notion of individualized treatment effect [80], wherein we assess the distributional shift of the task-model individually on each entity.

Let us consider the entity embeddings of a pair of entity sets $\mathbf{E}_1, \mathbf{E}_2$ to be drawn from a joint factual distribution P_F to optimize the factual link likelihoods $\mathcal{L}(\vec{\mathbf{e}}_1, r, \vec{\mathbf{e}}_2)\ \forall e_1 \in \mathbf{E}_1, e_2 \in \mathbf{E}_2$:

$$(\vec{\mathbf{e}}_1,\, \vec{\mathbf{e}}_2) \sim P_F(\mathbf{E}_1,\, \mathbf{E}_2) \tag{7.10}$$

Conversely, the node embeddings that satisy the counterfactual links of task-model $\mathcal{M}(j)$ induce a different joint distribution in the embedding spaces for entities $\vec{\mathbf{e}}_1(j)$, $\vec{\mathbf{e}}_2(j)$, depending on the objectives and inductive biases of model $\mathcal{M}(j)$:

$$(\vec{\mathbf{e}}_1(j), \vec{\mathbf{e}}_2(j)) \sim P_{CF}(\mathbf{E}_1(j), \mathbf{E}_2(j)) \tag{7.11}$$

We can marginalize the above distributions to obtain marginals $P_F(\mathbf{E})$ and $P_{CF}(j, \mathbf{E})$ for each entity set $\mathbf{E}$ under task-model $\mathcal{M}(j)$. This leads to the distribution difference,

$$\boldsymbol{\Delta}(j, \mathbf{E}) = \mathbf{KL}(P_F(\mathbf{E}),\ P_{CF}(j, \mathbf{E})), \tag{7.12}$$

We encode these distributional differences, $\boldsymbol{\Delta}(j, \mathbf{E})$ for each $\mathbf{E} \in \mathcal{E}$ under each $\mathcal{M}(j)$ via residual shifts [56]. This enables bidirectional knowledge-transfer between the node embeddings and the respective task-models via forward and reverse residual shifts.

7.4.4 Counterfactual Residual Functions

To generate consistent embedding updates, the counterfactual link likelihoods must account for the distributional differences induced by the factual and counterfactual links, i.e., $P_F(\mathbf{E})$ vs. $P_{CF}(j, \mathbf{E})$ for each $\mathbf{E} \in \mathcal{E}$.

The counterfactual and factual embedding distributions are an instance of a covariate-shift in the embedding space, a special case of domain adaptation [30, 80]. The covariate-shift is however, specific to each task-model $\mathcal{M}(j)$ and entity set $\mathbf{E}$. We propose to learn this shift using residual functions $\boldsymbol{\delta}[j, \mathbf{E}_1(j)]$ to shift the entity embeddings as follows:

$$\begin{aligned} \vec{\mathbf{es}}_1(j) &= \vec{\mathbf{e}}_1(j) + \boldsymbol{\delta}[j, \mathbf{E}_1(j)]\ (\vec{\mathbf{e}}_1(j)) \\ \vec{\mathbf{es}}_2(j) &= \vec{\mathbf{e}}_2(j) + \boldsymbol{\delta}[j, \mathbf{E}_2(j)]\ (\vec{\mathbf{e}}_2(j)) \end{aligned} \tag{7.13}$$

where each residual function $\delta[j, \mathbf{E}]$ is given by,

$$\boldsymbol{\delta}[j, \mathbf{E}_1(j)]\ (\vec{\mathbf{e}}_1) = tanh(\ \boldsymbol{W}[j, \mathbf{E}_1(j)]\,(\vec{\mathbf{e}}_1) + \mathbf{b}[j, \mathbf{E}_1(j)]\ ) \tag{7.14}$$

We hypothesize each task-model to produce a scaled shift on the distributional characteristics of entity embeddings, which we encode with the above scaling matrices $\mathbf{W}[j, \mathbf{E}_1(j)]$ with *tanh* receptive curves.

We then optimize the counterfactual likelihoods Equation (7.7), Equation (7.8), Equation (7.9) for the residual shifted entity embeddings $\vec{\mathbf{es}}$,

$$\mathcal{LPR}_{CF}(\ \vec{\mathbf{es}}_1(j), \vec{\mathbf{es}}_2(j)\ )\ \text{ in place of }\ \mathcal{LPR}_{CF}(\ \vec{\mathbf{e}}_1(j), \vec{\mathbf{e}}_2(j)\ ) \tag{7.15}$$

In this manner, the entity embeddings $\vec{\mathbf{e}}_1(j)$, $\vec{\mathbf{e}}_2(j)$ are only updated with the filtered gradients from the respective residual functions, $\boldsymbol{\delta}[j, \mathbf{E}_1(j)]$ and $\boldsymbol{\delta}[j, \mathbf{E}_2(j)]$.

Table 7.2: Residual Function Notations.

Symbol	Description
j, $\mathcal{M}(j)$	Prediction task j and model $\mathcal{M}(j)$
$\mathbf{E}_1(j)$, $\mathbf{E}_2(j)$	Input & output entity sets of $\mathcal{M}(j)$
$\vec{\mathbf{e}}_1(j)$, $\vec{\mathbf{e}}_1(j)$	Embeddings of an input entity $e_1(j) \in \mathbf{E}_1(j)$ with output $e_2(j) \in \mathbf{E}_2(j)$ from $\mathcal{M}(j)$
$\boldsymbol{\delta}[j, \mathbf{E}]$	Residual shift function for $\mathbf{E}$ under $\mathcal{M}(j)$
$\boldsymbol{\delta}[j, \mathbf{E}_1(j)]$	Residual shift function for the inputs of $\mathcal{M}(j)$
$\boldsymbol{\delta}[j, \mathbf{E}_2(j)]$	Residual shift function for the outputs of $\mathcal{M}(j)$
$\vec{\mathbf{es}}_1(j)$, $\vec{\mathbf{es}}_1(j)$	Residual shifted embeddings of $e_1(j)$, $e_2(j)$ $\vec{\mathbf{es}}_1(j) = \vec{\mathbf{e}}_1(j) + \boldsymbol{\delta}[j, \mathbf{E}_1(j)]\ (\vec{\mathbf{e}}_1(j))$ $\vec{\mathbf{es}}_2(j) = \vec{\mathbf{e}}_2(j) + \boldsymbol{\delta}[j, \mathbf{E}_2(j)]\ (\vec{\mathbf{e}}_2(j))$

7.5 TRAINING METHOD

In this section, we describe the overall training method to learn the residual functions in Table 7.2, and the algorithms for simultaneous graph embedding updates and model training (i.e., co-training), and the transfer of knowledge across task-models via serial updates (cross-training).

7.5.1 Learning the Counterfactual Residuals

We randomly sample a subset of focus entites from the inputs of $\mathcal{M}(j)$, i.e., $\mathbf{S}_1 \subseteq \mathbf{E}_1(j)$, and generate primary and secondary counterfactual links for each input $e_1(j) \in \mathbf{S}_1$ with task-model $\mathcal{M}(j)$ as described in Section 7.4. Let us denote this set of counterfactual links as $(e_1(j), e_{CF}) \in \mathbf{CF}(j, \mathbf{S}_1)$. Similarly, we denote the set of factual links associated with each $e_1(j) \in \mathbf{S}_1$ as $(e_1(j), r, e_F) \in \mathbf{F}(\mathbf{S}_1)$.

To learn the residual functions $\delta[j, \mathbf{E}_1(j)]$ across $\mathbf{CF}(j, \mathbf{S}_1)$ and $\mathbf{F}(\mathbf{S}_1)$, we optimize the following two objective functions alternatingly (stochasticity emerges from random selection of the subset, $\mathbf{S}_1 \subseteq \mathbf{E}_1(j)$):

$$\mathcal{L}_F = \sum_{(e_1(j), r, e_F) \in \mathbf{F}(\mathbf{S}_1)} \mathcal{L}(\vec{\mathbf{e}}_1, r, \vec{\mathbf{e}}_2) \tag{7.16}$$

$$\mathcal{L}_{CF}(j) = \sum_{(e_1(j), e_{CF}) \in \mathbf{CF}(j, \mathbf{S}_1)} \mathcal{LPR}_{CF}(\vec{\mathbf{es}}_1(j), \vec{\mathbf{es}}_{CF}) \tag{7.17}$$

with the above notations following from Equation (7.5), Equation (7.13) and Equation (7.15).

Note that $\mathbf{E}_{CF}$ (where $e_{CF} \in \mathbf{E}_{CF}$) can be any entity set, and is not limited to just $\mathbf{E}_2(j)$, since we also use the one-hop neighbors of $e_2(j)$ to form secondary counterfactual links. Optimizing Equation (7.16) and Equation (7.17) alternatingly results in simultaneous updates to both, the entity embeddings $\vec{\mathbf{e}}_1(j)$, $\vec{\mathbf{e}}_{CF}$, and the parameters of the residual functions, $\boldsymbol{\delta}[j, \mathbf{E}_1(j)]$ and $\boldsymbol{\delta}[j, \mathbf{E}_2(j)]$.

7.5.2 Graph and Model Co-Training

Section 7.5.1 focused on knowledge transfer from the task-model to the graph by learning the residual transformations across the factual and counterfactual links. We now describe our approach to train entity embeddings and task-model parameters bidirectionally for white-box task-models with a continuous differentiable objective function.

Note that each residual function is applied additively to the entity embeddings, as described in Equation (7.17). However, in Equation (7.17), the task-model is held fixed, i.e., we only perform the backpropagation updates to the entity embeddings. The direction of information flow is from the task-model to the embeddings. Conversely, if we wish to update the task-model $\mathcal{M}(j)$, we need the gradients to flow from the embeddings to the model. To achieve this directionality, we can apply the same residual transformations to the embeddings of factual links in the graph (instead of the counterfactual links); and then add them as a soft-alignment criterion to the task-model optimization objective.

We again sample focus entities $\mathbf{S}_1 \subseteq \mathbf{E}_1(j)$, and their factual links $\mathbf{F}(\mathbf{S}_1)$ as described in Section 7.5.1. For each link $(e_1(j), r, e_F) \in \mathbf{F}(\mathbf{S}_1)$, we estimate the likelihood on the shifted versions as follows:

$$\mathcal{SA}(e_1(j), e_F) = \mathcal{L}(\vec{\mathbf{es}}_1(j), r, \vec{\mathbf{es}}_{CF}) \tag{7.18}$$

Where $\mathcal{L}$ is the factual likelihood in Equation (7.5), applied to the residual shifted embeddings of $e_1(j)$ and e_F. We can now regularize the objective function $\mathcal{O}(j)$ of $\mathcal{M}(j)$ with the above terms:

$$\tilde{\mathcal{O}(j)} = \mathcal{O}(j) + \boldsymbol{\lambda}(j)\Big(\sum_{\mathbf{F}(\mathbf{S}_1)} \mathcal{SA}\left(e_1(j), e_F\right) - \mathcal{M}(j)\left(e_1(j), e_F\right)\Big) \tag{7.19}$$

Here, we overload the $\mathcal{M}(j)\left(e_1(j), e_F\right)$ term to indicate how $\mathcal{M}(j)$ measures the proximities of its input and output entities. The second term matches the model proximities to those suggested by Equation (7.18). The parameter $\lambda(j)$ determines the strength of the regularization. We can simulataneously update the model parameters, as well as the entities and residual functions by alternatingly optimizing all three objectives, Equation (7.16),

Equation (7.17), and Equation (7.19).

7.5.3 Model to Model Cross-Training

Let us consider the following directionality of model-to-model cross-training: say $\mathcal{M}(j_1) \rightarrow \mathcal{M}(j_2)$. For cross-training these two models, we need the condition $\{\mathbf{E}_1(j_1), \mathbf{E}_2(j_1)\} \cap \{\mathbf{E}_1(j_2), \mathbf{E}_2(j_2)\} \neq \Phi$, i.e., at least one of the entity sets whose node emebddings are updated by the counterfactual likelihoods in Equation (7.17) is present across both, $\mathcal{M}(j_1)$ and $\mathcal{M}(j_2)$.

We explain the model to model cross-training with the sample scenario where $\mathbf{E}_2(j_1) = \mathbf{E}_1(j_2)$, i.e., the output entity set of the first task-model is the input entity set of the second task-model.

- Learn the first cut entity embeddings $\vec{\mathbf{e}} \; \forall \; \mathbf{E} \in \mathcal{E}$ by optimizing Equation (7.5) over the factual links.
- Select the first model $\mathcal{M}(j_1)$, and learn the residual functions $\delta[j_1, \mathbf{E}_1(j_1)]$ and $\delta[j_1, \mathbf{E}_2(j_1)]$ by alternating optimization of Equation (7.16) and Equation (7.17), while holding the entity embeddings constant.
- Now update the entity embeddings for the entity sets $\mathbf{E}_1(j_1)]$ and $\mathbf{E}_2(j_1)]$ with the optimization described in Equation (7.17), while holding the residual functions constant.
- Finally, with updated node embeddings of the entity set $\mathbf{E}_2(j_1)] = \mathbf{E}_1(j_2)]$ (since the output set of $\mathcal{M}(j_1)$ is the input set for $\mathcal{M}(j_2)$), and perform the graph-to-model updates described in Section 7.5.2 to train $\mathcal{M}(j_2)$.

We observe that our overall framework is not theoretically exchangeable since the order $\mathcal{M}(j_1) \rightarrow \mathcal{M}(j_2)$ influences the final results. This is a fundamental limitation of the sequential course of knowledge transfer in our framework. This limitation also applies to the order in which models are co-trained and updated with the knowledge graph.

7.6 EXPERIMENTAL RESULTS

In this section, we present our experimental analyses on diverse multi-domain datasets and validate our framework.

First, we show that counterfactual enrichment with effective task-models can significantly improve node embedding quality with sparse connections by evaluating the updated embeddings on the held-out link completion task. Next, we show that co-training a context-aware neural recommendation model with the knowledge graph leads to simultaneous embedding updates and better model performance for nodes with lower degrees. However, we notice a minor degradation in the performance for high-degree nodes. Additionally, we exhibit that we can significantly improve the above context-aware neural recommendation model by leveraging a distributed word embedding model using the illustrated cross-training method. Finally, we do a scalability analysis against publicly available baseline implementations and conclude with limitations and discussion.

7.6.1 Data Description and Experiment Setup

Google Local Reviews Dataset [57, 151]: Users rate businesses on a 0-5 scale with temporal, spatial, and textual context features in each review. We filter this dataset with a criterion of at least ten users per business and five businesses per user recursively and eliminate all reviews with less than a 3-star rating. The resulting dataset has 38,614 users and 26,922 businesses, and the following contextual node types - Review Words, Business Name Words, Categories of the Business, Price, Location nodes - states, cities, and Temporal - time (binned into 6-hour chunks), month, day.

We create our knowledge graph by connecting all users to the businesses they rated, the name and review words of the businesses to each business, the review words, categories of visits, and business names to the users who rated them, the priceyness, locations, and times to businesses and users. On each of these links, we associated a 1-4 level depending on the strength of the associations (measured statistically on a per-user and per-business basis). These levels constitute our relation types.

Yelp Challenge Dataset: Users rate businesses on a 0-5 scale with temporal, spatial, and textual context features for each review. We filter this dataset with a criterion of at least 30 users per business and ten businesses per user recursively and eliminate all reviews with less than a 3-star rating. The resulting dataset has 25,3695 users and 69,738 businesses. We obtain the following contextual nodes - Review Words, Business Attributes, Location nodes - states, cities, lat-long (binned using a KD-tree), Temporal - time (binned by 6-hour chunks), month, day.

We create our knowledge graph by connecting all users to the restaurants they rated, the review words and attributes of the restaurants to each restaurant, the location nodes, the associated time nodes, and likewise for the users as well. On each of these links, we

Table 7.3: Google Local Graph Statistics

Entity Type	**Count**
Users	38,614
Businesses	26,922
Business Name Words	2,000
Review Words	5,000
Business Categories	650
Priciness	4
Time	23
Location	312
Total Nodes	**73,525**
Total Links	**7,325,614**

Table 7.4: Yelp Graph Statistics

Entity Type	**Count**
Users	20,750
Restaurants	75,871
Review Words	2,000
Business Attributes	200
Time	23
Location	1,062
Total Nodes	**99,906**
Total Links	**10,102,877**

associated a 1-4 level depending on the strength of the associations (measured statistically on a per-user and per-business basis). These levels constitute our relation types.

Baselines: We choose a broad array of diverse knowledge graph embedding baselines as a representative set to evaluate the edge completion task: TransE [18], DistMult [228], ComplEx [201], Rotate [193], RotH [24] and GAAT [214]. We used the OpenKE implementations[3] in Tensorflow/PyTorch with default parameter settings, wherever applicable.

7.6.2 Task-Models

For both datasets, we used a pair of task models with the same input entity-set (users) and different output entity sets (business category and businesses, respectively).

We train the distributional word2vec word-embedding model [139] on the set of review

[3]http://139.129.163.161//

text words, business names, and all the business attributes text over all the reviews in the dataset. We use the basic version (non-transfer) of the context-aware recommender proposed in Krishnan et al. [98] with the non-textual categorical links of the users and businesses (as above) forming the context of each review. To predict business category/attribute words for each user, we take an average of their review word set embeddings, and map the average to the closest business category words as learned by the model. Note that to train the word2vec model, we use the review text as a context for the business attribute text.

Parameters: In both the above datasets, for the context-aware recommendation model [98], we use the author recommended parameters with 200-dimensional embeddings, while we use the gensim[4] implementation of word2vec with a maximum 10-length window. The additional parameters of our model, such as the discrepancy scaling in Equation (7.17) were tuned with an exponential grid-search approach (e^{-5} to e^{0}). The knowledge graph and counterfactual residuals were also trained with 200-dimensional embeddings, and implemented in Tensorflow, and run on a Tesla K80 GPU.

Metrics for Link Prediction: In both the datasets, we attempt to predict held-out links using the embeddings learned by our models, as well as the embedding baselines. For each held-out link of the form (e_1, r, e_2), we create several negative samples of the form $(e_1, r, \tilde{e}_2)$ and $(\tilde{e}_1, r, e_2)$, i.e., with the same relation type and head and tail entity types, however a randomly sampled entity for either the head or tail. We then rank the entire list of negative samples against the true link (e_1, r, e_2) under each embedding model and measure the **Recall@K** metric for the respective ranked lists. Specifically, we measure the **Recall@5**, **Recall@10** for two types of held-out links - *User* $\rightarrow$ *Business* and *User* $\rightarrow$ *Category-word* (Attribute in case of yelp), for a 100-length ranked list.

7.6.3 Primary Results - Link Prediction

We evaluate the above two knowledge graphs on the link completion task. We randomly tag 20% of the user nodes as held-out nodes. We then held out two types of links for these users - we held out half of their user-business links and half of their user-business attribute/category word links. These two link types directly correspond to the two task models we used: The word2vec model predicts user-business category word links while the context-aware recommender predicts the user-business links.

For our model, we present two variants - MUTATE-F, which only uses the factual nodes, and MUTATE-CF, which uses counterfactual enrichment for the held-out user set. Specifically, we use the top-5 words predicted by the word2vec model and the top-5 businesses

[4]https://pypi.org/project/gensim/

Table 7.5: Overall Link Prediction Results. Bold-font denotes statistically significant gains over all baselines at the 0.05 significance-level under *paired t-tests*, while * denotes the second-best performer.

Link Type	**User to Business**		**User to Category**	
Metric	**R @ 5**	**R @ 10**	**R @ 5**	**R @ 10**
TransE [18]	0.43	0.60	0.52	0.68
RotatE [193]	0.59*	0.72	0.64	0.80
RotH [24]	0.58	0.76*	0.65*	0.79
DistMult [228]	0.56	0.70	0.63	0.77
CompleX [201]	0.57	0.70	0.61	0.76
GAAT [214]	0.59*	0.74	0.63	0.82*
MutatE-F	0.58	0.73	0.64	0.79
MutatE-CF	**0.62**	**0.80**	**0.68**	0.84

predicted by the recommender to form counterfactual user-business and user-word links. We also trained all the baseline embedding models on the same knowledge graphs and attempted to predict the same set of held-out links using their trained embeddings.

Key Observations from Table 7.5: The relative order of performance of the baselines is as expected, DistMult [228] performs moderately owing to the inverse nature of some relation-types in our graphs across user-context-business paths. In contrast, our base model can overcome this challenge and perform comparably to the other baselines.

We also observe that our MUTATE-CF model strongly outperforms all the competing models on user-word link prediction and user-business link prediction tasks. The two external task models, namely word2vec and the context-aware recommender, can better predict the missing links and enrich the graph compared to the heuristic or path-based link completion approach in the other baselines. It is easy to see how we can leverage the inductive biases of the specific models. While the word2vec model can interpret the review text's distributional properties, the context-aware recommender leverages the multiplicative predictors from the context features. Also, note that these two models use the same data as the Knowledge Graphs and do not depend on any external sources.

7.6.4 Co-Training Model with Graph

In this section, we describe our co-training approach for the recommender model with the knowledge graph. Specifically, we make predictions from these models for users and use these counterfactual links to update the knowledge graph embeddings, as described in Equation (7.16). Simultaneously, we make predictions from the updated embeddings for the

Table 7.6: Co-training Performance Gains against the Information-flow Parameter λ^j from Equation (7.17)

λ^j	e^{-5}	e^{-4}	e^{-3}	e^{-2}	e^{-1}
Word2Vec	-5.6%	-1.3%	**+8.1%**	-4.9%	-18.6%
Context Recommender	+2.8%	-1.03%	**+5.4%**	-8.6%	-28.9%

users and use these to augment the loss function of the recommender model as described in Equation (7.18). In this manner, we attempt to improve the model performance over just training the model in isolation.

Although we did not achieve a dramatic performance difference, we observe that over-regularizing the model or under-regularizing the model is suboptimal. In other words, the co-training proceeds best when we set the regularizer λ^j to an optimal balance.

The numbers in Table 7.6 indicate the best performance improvements we were able to achieve for the recommender model under different settings of λ^j. A higher value of λ^j meant that the recommender was more constrained by the knowledge graph, while a lower value meant that more information flows from the model to the graph. Thus, we need an ideal trade-off between the forward and reverse information flow.

7.6.5 Cross-Training across Tasks

In this section, we describe our cross-training approach for the recommender model by leveraging the word2vec model. We first train the word2vec model on the base data, then use it to update the knowledge graph embeddings using the model to graph knowledge transfer method described in Section 7.5.3. We then use the reverse direction to regularize the recommender model as in Equation (7.19), i.e., knowledge now flows from the updated graph to the recommender model. Thus, the overall direction of knowledge flow is as follows:

$$\mathcal{M}^{word2vec} \rightarrow \textit{Knowledge Graph} \rightarrow \mathcal{M}^{context\text{-}aware\text{-}recommender} \tag{7.20}$$

Since the review text is informative of both the user embeddings and the business embeddings owing to their shared link structure, we were able to achieve noticeable performance gains for the recommender model (see Table 7.6) after we leveraged the sequence of update-steps described in Section 7.5.3.

However, we note that the reverse model-to-model transfer direction, namely from the context-aware recommender to the word2vec model, does not result in any noticeable performance gain (Table 7.8), indicating the importance of chosing a more informative model

Figure 7.4: Cross-training performance gains for the context-recommender with word2vec with respect to the parameter λ^j set to varying values as in Equation (7.17). Information flow directions are:
(a) $\mathcal{M}^{word2vec} \rightarrow KG \rightarrow \mathcal{M}^{context\text{-}aware\text{-}recommender}$ and
(b) $\mathcal{M}^{context\text{-}aware\text{-}recommender} \rightarrow KG \rightarrow \mathcal{M}^{word2vec}$.

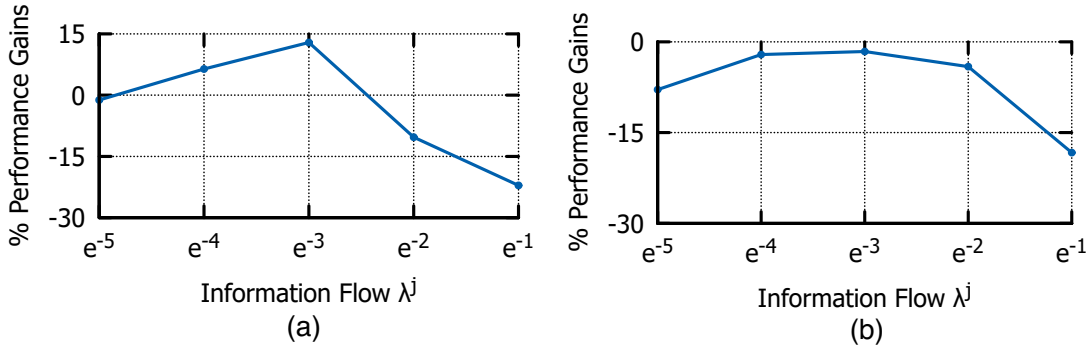


Table 7.7: Cross-Training performance gains for the context-recommender with word2vec, where the information flow direction is $\mathcal{M}^{word2vec} \rightarrow$ *Knowledge Graph* $\rightarrow \mathcal{M}^{context\text{-}aware\text{-}recommender}$, parameter λ^j is again set to varying values as in Equation (7.17), percentages relative to isolated performance.

λ^j	e^{-5}	e^{-4}	e^{-3}	e^{-2}	e^{-1}
Context Recommender	-1.2%	**+6.4%**	**+12.9%**	-10.3%	-22.1%

to enrich the knowledge graph before attempting transfer.

7.6.6 Sparsity Analysis

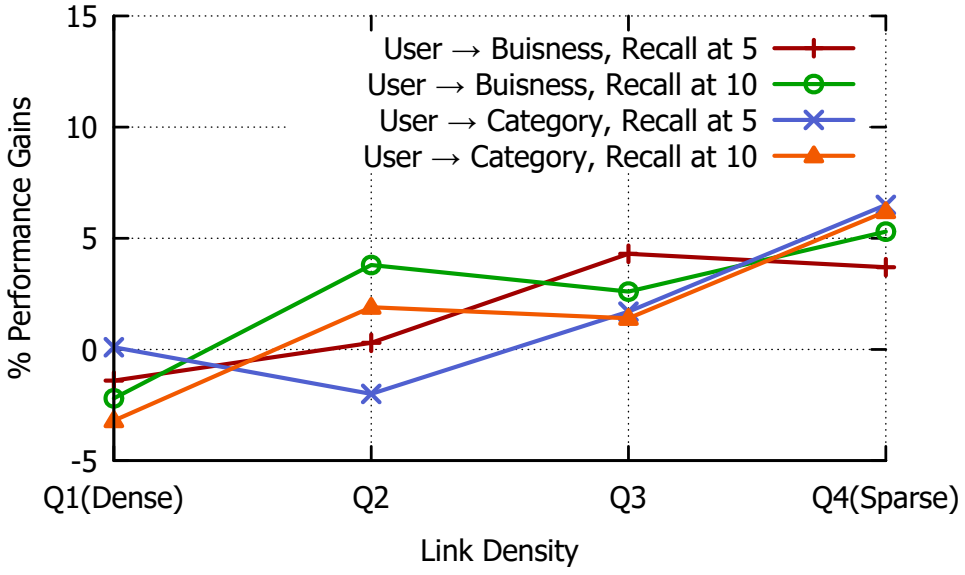
In this subsection, we attempt to study the impact of counterfactual updates on sparse and non-sparse nodes. Specifically, for both the tasks, user-word link prediction and user-business link prediction, we study the relative gains obtained by counterfactual updates, i.e., the difference in MUTATE and MUTATE-F performance in the different sparsity sets. $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\mathbf{Q}_3$ and $\mathbf{Q}_4$ denote the four sparsity quartiles for each respective user node. We then measure the average performance difference between MUTATE and MUTATE-F for each quartile in Figure 7.5.

As expected, we obtain the most robust gains for sparse users, i.e., users in quartiles Q3/Q4, since they lack the word associations to help us learn better node embeddings. Thus, the distributional knowledge encoded in the word2vec model can significantly bridge this gap in the knowledge graph and enrich the corresponding node embeddings.

Table 7.8: Cross-training performance gains for the word2vec model, where the info flow direction is $\mathcal{M}^{context\text{-}aware\text{-}recommender} \rightarrow Knowledge\ Graph \rightarrow \mathcal{M}^{word2vec}$, parameter λ^j is again set to varying values as in Equation (7.17), percentages relative to isolated performance.

λ^j	e^{-5}	e^{-4}	e^{-3}	e^{-2}	e^{-1}
Word2vec	-7.9%	**-2.1%**	**-1.6%**	-4.1%	-18.3%

Figure 7.5: The gains of MUTATE-CF relative to MUTATE-F on the two types of link prediction. In each case, we measure the performance gains across 4 quartiles of users, arranged by the density of that specific type of link for the user.



7.6.7 Limitations and Discussion

Our work's two primary weaknesses are the non-exchangeability of the order in cross-training and the assumption of homoscedastic embeddings within each entity set. In other words, we assume that a single residual function, conditioned on the node embeddings of each node, can fully account for the distributional differences introduced by the task-models.

A few alternatives exist to capture heteroscedastic node embeddings, such as Gaussian mixture embedding spaces [22]. However, they are hard to implement efficiently within a knowledge graph neural network optimization framework, owing to the expensive optimization structure and strong constraints on the learned embedding spaces, thus limiting generalizability across diverse task-models.

Further, since we do not bound the task-models' nature, we do not have a tight bound to describe the discrepancy distance function in Equation (7.17). We plan to study the trade-offs between generalizability and theoretical guarantees on the residual functions or overall

Table 7.9: We measure the gains of MUTATE-CF relative to MUTATE-F on the two types of link prediction, and in each case, we measure the performance gains across 4 quartiles of users, arranged by the density of that specific type of link for the user.

Link-Type	**R@K**	**Q1 (Dense)**	**Q2**	**Q3**	**Q4**
User → Business	R@5	-1.4%	+0.3%	+4.3%	+3.7%
	R@10	-2.2%	+3.8%	+2.6%	+5.3%
User → Category	R@5	+0.1%	-2.0%	+1.7%	+6.5%
	R@10	-3.2%	+1.9%	+1.4%	+6.2%

exchangeability in future work.

7.7 RELATED WORK

Knowledge graphs are essential resources for many AI tasks today. While one branch of research considers the knowledge graph as an oracle and develops machine learning models that leverage existing connectivity patterns to improve task outcomes, they often suffer from incompleteness.

A variety of representation-based/embedding methods - tensor factorization based and neural network-based - have been developed that attempts to enrich the knowledge graph and incorporate latent structural proximities of nodes by transitively learning a range of simple heuristic patterns among the nodes [18, 74, 75, 116, 148, 149, 193, 219]. These patterns are unable to distinguish the different relation types and are applied in an equivalent manner to all of them. Thus, it can lead to contradictory and incorrect inferences, which in turn, may violate the domain knowledge. Additionally, some of these methods are also not suited to handle unbalanced heterogeneous graphs.

Several recent efforts have attempted to leverage the knowledge graph structure for recommendation [3, 194, 216]. The methods are either path-based that feed the high-order information to the predictive model or regularization-based that leverages the network structure to regularize the recommender model learning.

However, the above methods are typically not optimized for the specific recommendation objective since they rely on the same static view of the underlying knowledge graph. Conversely, the inductive task-models cannot be directly leveraged to densify or improve the knowledge graph either. Other tasks such as search personalization [147] and question-answering [71] also suffer from similar drawbacks. To overcome these limitations, we propose a holistic solution that subsumes multi-task learning and knowledge graph enrichment via counterfactual residual functions.

7.8 CONCLUSION AND FUTURE WORK

This chapter proposes a holistic view of knowledge graph representations and multi-task learning that permits the multi-directional transfer of knowledge between domain-specific knowledge graphs and task-models. The proposed framework is highly generalizable and can integrate diverse tasks and model architectures through a common set of underlying knowledge embeddings. The proposed strategy overcomes both the fundamental limitations of prior work; It permits multiple views of the underlying node representations via task-specific residual functions while also enabling co-training across each gradient-updated task-model underlying graph, independent of the model architecture.

Our framework effectively models different task-specific distributions with the same underlying knowledge graph via counterfactual residual learning. The fundamental reason for our gains is simple: No single embedding representation can capture the task-specific distributions across diverse tasks unless all tasks are perfectly correlated to each other. As a result, we enable task-specific expressivity in an architecture agnostic manner and overcome the above fundamental challenge. In the future, we intend to study the trade-offs between generalizability and theoretical guarantees on the residual functions and overall exchangeability.

In the next chapter, we provide a bird's eye overview of all the previous chapters, the essential conclusions that we draw from our work, the most promising avenues for future work to extend our work and address the broader questions that we attempt to answer in the previous chapters of this thesis.

CHAPTER 8: CONCLUSIONS AND FUTURE WORK

8.1 INTRODUCTION

Our generalizable, multimodal, multi-source representation-learning, and regularization frameworks were focused on addressing behavioral data skew and sparsity across recommendation and personalized user inferencing models in this thesis. Although human behavior exhibits activity skew and sparsity across various applications and contexts (e.g., economic markets, crowdsourcing), our objective of addressing these challenges through empirical data-driven methodologies requires targeted problem settings to make precise observations and insightful contributions.

We choose the recommendation problem due to its broad application across diverse platforms incorporating diverse user and item participation modalities, each exhibiting unique characteristics in how the skew and sparsity issues manifest and the modeling criteria. Further, the recommendation problem also incorporates either direct subproblems or auxiliary problems, predicting interactions among entities of different types associated with either users or items in different contexts. This lends the recommender models heterogeneity and broader coverage across machine learning domains and tasks. Further, its importance to applications, including e-commerce, social media, and advertising, maximizes our work's impact.

Neural recommender models represent state-of-the-art performance and the ability to accommodate diverse modeling hypotheses. To enable neural recommender models to handle data skew, we focus on organizing their latent representations to account for the data characteristics, rather than externally altering the data distribution to best fit the operating regime of neural models, e.g., classical approaches to address class imbalance involve under/over-sampling strategies to create balanced training samples. We note that static approaches employ heuristics that are not designed to optimize specific neural network architectures and training algorithms. Our data-driven solutions overcome these mismatches with static criteria.

Finally, our multimodal framework is agnostic to the representation models' architectural choices corresponding to each data modality. In this thesis, we choose to remain architecture-agnostic to maximize our proposed framework's applicability across a diverse set of neural recommenders. However, a feasible alternative to tackle data challenges is that of architectural enhancement, whereby specific neural modules could be developed and / or pruned to improve model performance and robustness [121].

8.2 RESEARCH SUMMARY AND TAKEAWAYS

8.2.1 Sparsity and Skew-Aware Representations

In the previous chapters of this thesis, we developed and demonstrated clustering and representational organization strategies to the respective latent representation spaces toward learning sparsity and skew-aware entity representations with neural recommendation and modality-specific representation models in tandem to learn entity representations. Our strategies are predicated on the below methodological contributions of this thesis:

Data-Driven Clusters with Soft Guidance: The representation clustering strategy introduced in Chapter 3 differs from prior models in two key respects. First, we identify and eschew implicit hypotheses about the user data's distributional characteristics along any dimension, irrespective of the graphical/probabilistic model for the specific application or platform.

Note that our clustering strategies incorporating probabilistic models trivially extend to neural generative models as well [111]. The guidance provided to the learner to update its parameters is data-driven, in the form of mutually coupled iterative profile-learning and entity grouping. In this manner, the model design is not restricted to any specific distributional assumptions, nor any specific parametric assumptions or architectural specifics.

Generalizing Aggregate Co-Occurrences: While aggregate co-occurrence information is often a valuable signal to understand and represent sparse entities [129], it does not distinguish the specific semantics of each co-occurrence. In Chapter 4, we develop a self-supervised learning framework to better distinguish the aggregate co-occurrences by simultaneously understanding user preferences and item representation strategies. Further, we introduce architecture-agnostic learning by decoupling the item representation model from the user preference representations. The contextually conditioned item co-occurrences act as soft regularizers unifying the two sets of representations.

Context-Conditioned Clustering: In Chapter 4 and Chapter 5, we develop context-conditioned alignment strategies, and apply them in two very distinct applications. While in Chapter 4, the context-conditioning serves to distinguish and characterize the co-occurrences of different entities.

In Chapter 5, we employ context-conditioning to differentially weight and unify the distinct data-modalities for a given entity. The objective of conditioned representations in Chapter 4 is to enable association learning across different entities. In contrast, Chapter 5 learns to aggregate data-modalities for a given entity towards a recommendation or inference task. It is feasible to combine the two strategies as well, learning associations within each data-

modality in tandem with cross-modal combinations.

8.2.2 Handling Multimodal Inference and Recommendation

The multimodality of user data on online platforms is a key consideration across several chapters in this thesis.

Model Architecture and Data-Modality Agnostic Abstractions: Our clustering and knowledge transfer strategies are grounded on generalizable abstractions. Expressly, our abstractions do not limit the kinds of data-modalities or model architectures and transformation functions that can be applied towards learning user or item representations.

We demonstrate multiple such abstractions: In Chapter 3, the skew-aware grouping mechanism adapts to fit the generative profile model chosen to describe user data. In Chapter 5, the proposed adaptive noise contrastive estimation strategy relies not on any specific architectural constraints or dependencies. In the cross-domain scenario, we adopt a module-level abstraction, where the modules interface via entity representations. As a result, each module may be independently modified without impacting the transferability of the invariant structures. Finally, in Chapter 7, we enable multi-task residuals to adapt to the task distributions agnostic to the specific models (and their inductive biases) that generate the distributions.

Aggregate Representation Strategy: We contextualize the aggregation of multimodal representations associated with each user or entity. This is a fundamental contribution over prior work, owing to heterogeneous participation across independent data generation processes. The resulting attribution enables us to select the appropriate data-modality (or modalities) in a weighted manner to explain each training sample. Thus, the gradient updates are selectively used to update the respective representations.

Noise Contrastive Estimation: In Chapter 4 and Chapter 5, we demonstrate self-supervised learning across data-modalities with a noise contrastive estimation strategy, where we select or generate the best negative samples or cross-modal samples to cluster their entity representations simultaneously. The quality of the negative samples is reliant on both the current model state and the training data. This accounts for both the distributional heterogeneities introduced by uneven entity participation across the data modalities and the aggregation of the respective representations towards a joint objective.

8.2.3 Cross-Domain and Multi-Task Knowledge Sharing

In this thesis, we propose two pathways to cross-domain and multi-task recommendation and inference via knowledge sharing.

Cross-Domain Contextual Invariants: Our key intuition is to infer combinatorial behavioral invariants from users' interaction histories in a dense-source domain. We subsequently transfer and adapt these learned invariants to improve inference in sparse-target domains. Clustering users who interact under covariant combinations of contextual predicates in different domains lets us better incorporate their behavioral similarities and analogously infer item clusters and the overall user-cluster to item-cluster mappings in the sparse domain.

Task-Specific Residual Adaptation: Residual functions are added to the base representations of entities, thus enabling implicit parameter sharing. The shared aspect derives from the underlying knowledge graph, while the task-specific aspects are encoded in the respective residual transformations.

Hard parameter-sharing across domains or tasks severely restricts the expressivity of the joint model. To overcome these challenges, in Chapter 6, we alter the *input distribution* to the shared modules to account for variance across target domains, rather than learning an alternate set of parameters from scratch. Analogously, our inexpensive residual learning strategy in Chapter 7 accounts for varying task distributions in their respective entity representation spaces.

8.3 DATA CHALLENGES BEYOND SPARSITY AND SKEW

8.3.1 Diversity and Fairness

Part of the reason for recommender systems' success is their ability to recommend relevant items to which the user has no direct or indirect ties by identifying latent characteristics of the item and aligning them to the preferences of the users. Thus, the phenomenon of reduced diversity and low inventory coverage with neural recommender models [95] significantly impacts their efficacy and user retention [50].

While prior work in diversification tries to solve the overfitting problem and improve personalization [100], each user perceives diversity differently, not only in an aggregate sense but also from the contextual view of each interaction. Thus, the precise set of items that constitute diversity (in terms of utility) should be defined on a per-interaction basis. The context pooling strategy that we use to group users and items in Chapter 6 may be leveraged towards defining contextual utility.

However, consumer utility is not the only consideration of online recommendation platforms. Notions of recommendation diversity and fairness are intertwined. Still, the outcomes are not adversarial, unlike the multimodal attribute problem addressed in Chapter 4.

Broadly, fairness efforts can be classified along a few axes.

- **Stakeholder Fairness:** The validation methods employed in the preceding chapters of this thesis primarily focus on the user satisfaction objective. However, real-world recommendation applications involve multiple stakeholders apart from the user, typically providers and side stakeholders [2] (stakeholders facilitate the provision of the items and services to the users). Thus, the precise recommendations provided to users must balance consumer satisfaction with provider and side stakeholder constraints and preferences (referred to as *C, P* and *S-fairness* respectively). Irrespective of the precise entities of focus, the ability to handle interaction data challenges empowers recommender systems to address their requirements holistically.

- **Group Fairness:** Notions of group fairness are not limited to any specific set of entities and apply to all stakeholders and items / content and services. Group fairness can be measured via parity of outcomes (recommender outcome fairness) and equal treatment (e.g., group representations in the recommender model training process). Typically, group fairness efforts are hindered by the unavailability of high-quality training data for subsets of the user and item populations. In such scenarios, the knowledge extraction, transfer, and adaptation approaches presented in this thesis are directly applicable to supplement data augmentation and domain expert interventions.

Demographic Parity: Demographic parity is succinctly represented as follows,

$$P(\mathbf{Y}_j = 1|\mathbf{A} = 0, \mathbf{X}) = P(\mathbf{Y}_j = 1|\mathbf{A} = 0, \mathbf{X}) \tag{8.1}$$

Where Y_j denotes the likelihood of recommending item-j given the protected attribute(s) $\mathbf{A}$ and the non-protected attributes $\mathbf{X}$ associated with the user, such as their item consumption histories.

Demographic parity is closely linked to observation bias notions. The missing ratings are not randomly distributed when marginalized over the protected attribute(s) since users cannot rate the content they are not recommended.

The Importance of Modular Learning Frameworks: The modularity and generalizability concerns highlighted in the previous chapters have implications for fairness and diversification efforts. For instance, the ability to simultaneously address individual recommendation quality (measured via RMSE, MAE etc. [224]) and group treatment metrics (measured via group metrics [16]) requires us to subsume user representations and improve the attribution methods that link outcomes to the user representations and underlying user data.

Further, the precise objectives depend on the application scenario and desired outcomes. For instance, the value unfairness metric [232] may not be suited to recommendation problems with differential costs associated with the overestimation and underestimation unfairness metrics [232]. Thus, the ability to decompose the overall model and apply the respective objectives irrespective of the precise data modalities and model architectures is an important consideration.

8.3.2 Scalability Aspects

In practice, recommender systems are highly complex systems incorporating several interconnected modules ingesting and processing the input data (ETL pipelines [204]), as well as the user feedback and the associated update loop [78]. While optimizations to the data pipeline and deployment / feedback loops are largely outside this thesis's scope, our approaches' overall modularity enables the reuse of both recommender models and the associated optimization and deployment aspects.

Scaling Knowledge Extraction and Model Training: The following axes are useful to speed up the training process for architecturally complex neural recommendation models:

- *Accounting for Sample Informativeness:* Informativeness of training samples is typically measured via heuristic metrics, such as information entropy and gradient variance-reduction [127]. In contrast, we propose measuring and updating the sample informativeness metrics as a function of the recommender model's current state. Critic models can be chosen from an appropriate class of neural architectures and may be conditioned by auxiliary data [197] or architectural considerations [22]. The key advantages of critic-based approaches to model training are visible in multimodal (Chapter 5) and multi-objective learning (Section 8.3.1), where a single measure, objective, or metric cannot adequately address recommendation applications.

- *Variance Reduction:* Variance reduction heuristics attempt to maintain continuous and steady convergence while trading off the magnitude of the updates to achieve the best-amortized training times [31, 81]. A specific instantiation of this approach is the online batch-selection [128] strategy, where the metrics are not computed just once but rather factor the updates from prior batches to pick the best subsequent batches of training points in the training process. The batch selection meta-problem can be iteratively solved alongside the recommender model, analogous to the critic-models trained in Chapter 4 and Chapter 5.

- *Curriculum / Multi-stage Learning:* While curriculum learning is typically applied to multi-task learning problems [52], the broad concepts apply to recommendation systems as well, especially with the multi-stage training approach. Multi-stage learning can be applied by first training the recommender system on a subset of the user and item data (i.e., core subsets), followed by adaptation or local parameter search methods (e.g., simulated annealing) to fit the non-core entities and their representations [98]. Since memory usage and computation costs often scale non-linearly [102], multi-stage learning's marginal advantages are significant. They may even result in improved task performance.

Multi-task / Multi-domain Transfer Learning: We specifically refer to transfer learning methods in the context of neural models, where the presence of data or training invariants is critical to model reuse [9]. Methods that rely entirely on layer activations [126] are not interpretable in the context of deep collaborative recommenders since they are not bound to specific facets or patterns in the training data. In these application scenarios, invariants may be induced either by restructuring data representations as in Chapter 7 or orienting the latent representations to induced invariants as in Chapter 6.

Beyond Transfer Learning - Knowledge Distillation: Unlike transfer learning methods, knowledge distillation is primarily employed to reduce the resource footprint associated with a sophisticated over-parametrized neural network model for a specific task instantiation / use-case. The distilled model (or the *student* model), which is deployed to the resource-constrained use-case, is typically trained only to mimic specific aspects of the *parent* model and does not learn from the raw data. This enables us to train multiple student models with slightly different performance objectives and avoids the overheads of retraining an expensive parent model towards each objective independently [28].

8.3.3 Streaming Data Applications

This thesis's preceding chapters primarily focus on conventional recommendation models that utilize all historical user-item interactions (interactions referring to the direct or indirect user actions on items) to learn representations of users, items, and other entities. This approach implicitly assigns equal importance to all of the historical interactions towards inferring current preferences. On the other hand, incremental temporal models aim to address the non-uniformity of past interactions towards inference tasks [215].

Session Formulation vs. Sequence Formulation: The session formulation assigns user intent to blocks of user interactions, separated by either explicit markers such as plat-

form logs or by implicit criteria such as the duration of user inactivity between two subsequent interactions [108]. Session formulations typically incorporate notions of short and long-term user interests [6]. The short-term interests are handled with sequential dependency considerations. In contrast, the long-term interests form priors over the recommendations made to the users. Note that short-term interests also lack notions of evolution, while long-term representations are often updated across sessions.

In contrast, sequence-aware recommender systems are limited to sequential representations of user interactions [82], without explicit session demarcations. As a result, these models do not explicitly distinguish the short and long-term interests of users.

Extending our Work to Temporal Recommender Models: We analyze each chapter's extensions and application to the temporal recommendation problem.

- *Chapter 3 - Joint mitigation of skew and sparsity:* Note that the precise computation of the profile likelihood for users in Equation (3.2) does not influence the grouping mechanism but rather modifies the criteria for user grouping. We identify two broad approaches to extend our work; the first replaces the profile model in Equation (3.2) with a temporal model, which results in users being grouped by their evolution trajectories. Alternately, we can modify the seating arrangement on a session-segmented basis, analogous to session models [215] so that the seating arrangement is permitted to evolve.

- *Chapter 4 - Item representations:* Although the framework in Chapter 4 does not permit for item evolution; the model can be applied to each temporally sliced segment of the user data (i.e., snapshot) or sessions, independently utilizing only the item co-occurrences in the respective frames to guide the item association structure.

- *Chapter 5 - Adversarial attribution:* The overall framework is directly applicable to the temporal recommendation problem by adversarially regularizing a session-aware recommender model in place of a static neural collaborative model. The framework also permits the user of a temporal social model, e.g., for evolving social networks [178].

- *Chapter 6 and Chapter 7 - Knowledge transfer:* Contextual invariants in Chapter 6 and the knowledge graph invariants in Chapter 7 do not trivially extend to temporal settings. Still, they may be modified by either learning the invariants on temporally sliced segments of the user data or permitting the invariants to evolve as a function of the timestamps, i.e., the invariants are no longer static but rather learned functions of the interaction timestamps.

8.4 LONG-TAIL PROBLEMS BEYOND RECOMMENDER SYSTEMS

This section discusses prevalent long-tail challenges across other machine learning domains and both the similarities and dissimilarities in contrast to the challenges handled in this thesis. We also discuss the applicability of the presented work and its underlying conceptual frameworks to these alternate problem settings.

Majority-Minority Classification Problems: Imbalanced classification problems appear in several machine learning domains including computer vision [69], real-world object detection [123], language processing tasks such as sentence entailment and relationship classification [185], and medical applications such as disease-diagnosis [184]. Some of the common solution approaches include data augmentation [233], sample reweighting [186], metric learning [117], hard-negative mining [37] and meta-learning [202]. We previously discussed some of the challenges and shortcomings associated with each class of techniques in Chapter 2. In this section, we focus on how our approaches can be extended to these application scenarios.

Open-Set and Closed-Set Formulations: In several real-world problems in domains such as computer vision and language processing, machine learning models are required to classify among a few common and many rare categories [175]. These models are needed to generalize the concept of a single category from only a few known instances and simultaneously to acknowledge novelty upon an example of a previous unseen category or class [123]. The open-end distributed data does not bind the set of classes associated with the data points. Instead, it establishes a continuous spectrum of the head, tail, and open or previously unknown classes [123].

Open-set applications require handling imbalanced classification and few-shot learning to handle tail classes in the closed (or known) part of the class spectrum and recognize open-set instances with one integrated algorithm. Note the connections of such a formulation to the iterative group-discovery algorithm developed in Chapter 3. Static classification and sparsity-mitigation approaches focus on one aspect and deliver poorly over the entire class spectrum. Both tail robustness, i.e., the ability to accurately identify instances of tail classes, and open-set identification, i.e., the model's sensitivity to previously unknown classes, are simultaneously handled.

While accurate tail identification may enable moving tail classes higher up the spectrum, open-set identification enables introducing new data classes to the long-tail. Modeling solutions must ideally perform both, share and transfer knowledge from the head classes to the tail classes, and learn sharper boundaries for the tail classes to separate the open classes.

Out-of-sample (OOS) Distributions: Successful training and deployment of ma-

chine learning models often require us to distinguish between data that is anomalous or significantly different from data accessed in training. This is particularly important for deep neural network architectures, which might incorrectly classify out-of-distribution (OOD) inputs into training classes with high confidence. This has significant implications when these predictions inform real-world decisions such as bacteria identification based on genomic sequences [165]. Bacteria detection informs diagnosis and treatment recommendations and helps identify new pathogens. Real-world data is ever-evolving. It will inevitably include genomes (or equivalently data samples) from previously unknown classes (OOD inputs).

Out of sample distributions may be viewed in two ways: Each architectural component $\mathcal{M}$ of the classifier model encodes input features $\mathbf{x}_{\mathcal{M}}$ to generate the output representation $\mathbf{y}_{\mathcal{M}}$. Note that $\mathbf{y}_{\mathcal{M}}$ denotes the distribution over the class labels for the aggregate model. However, intermediate representations may not directly correlate to the class labels.

$$p(\mathbf{y}_{\mathcal{M}}, \mathbf{x}_{\mathcal{M}}) = p(\mathbf{y}_{\mathcal{M}}|\mathbf{x}_{\mathcal{M}}) \times p(\mathbf{x}_{\mathcal{M}}) \tag{8.2}$$

where the parameters of $\mathcal{M}$ determine the conditional $p(\mathbf{y}_{\mathcal{M}}|\mathbf{x}_{\mathcal{M}})$ (i.e., *the task distribution*) and $p(\mathbf{x}_{\mathcal{M}})$ describes the inputs to component ($\mathcal{M}$).

Out-of-sample Challenges: There are two ways the above pre-trained component may fail at test-time:

- **Out-of-distribution Task:** The new task presented to the model does not obey the conditional $p(\mathbf{y}_{\mathcal{M}}|\mathbf{x}_{\mathcal{M}})$ encoded in the model parameters.
- **Out-of-distribution Inputs:** The input feature distribution $p(\mathbf{x}_{\mathcal{M}})$ has changed, so that we need to remap each dimension to maintain the feature distributions observed at train-time by the learned model.

In practice, we need a combination of task and input / feature adaptation strategies to address application scenarios and generalize pre-trained classification models.

Defining Learning Objectives with Sparse Training Data - Disentanglement: Models often fit spurious correlations between class labels and input features, owing to the limited number of samples in the training data where the spurious correlation may suffice to achieve high accuracies. This problem is especially pronounced in multimodal problems such as Visual Question Answering (VQA), where models have been shown to rely on superficial correlations between question and answer words, i.e., aggregate language priors (marginal distribution) instead of the joint distribution of the language and image representations [163].

However, this challenge is not limited to multimodal problems. Spurious feature correlations exist in unimodal problems, such as image classification as well [174]. These failures

typically result from geometric and statistical skews, i.e., the shapes of the feature distribution curves and how well they are separated by the aggregate class labels [189]. This phenomenon necessitates either extensive data augmentation to null-out all spurious correlations in the empirical risk minimization (i.e., overall gradient computation on the data) or domain-specific model design as in Ramakrishnan et al. [163].

8.4.1 Extending our Work to the Above Problem Settings

We now analyze each preceding chapter's potential extensions and application to the problem settings in Section 8.4.

- *Extensions to imbalanced classification problems:* The non-parametric grouping mechanism described in Chapter 3 has implications for the minority classes, specifically in incentivizing the discovery of the class members. Further, we do not require the classes to be known apriori, and instead, discover them at train-time. The inter-item association strategy in Chapter 4 can be generalized to inter-feature association structures across head and tail classes to improve long-tail class identification. Feature associations can be leveraged to introduce invariant latent dimensions associated with the long-tail classes, analogous to Chapter 6.

- *Extensions to open-set formulations:* The grouping mechanism described in Chapter 3 enables class discovery with the exploration sensitivity governed by the discount parameter. A second strategy is to identify feature invariants for the head classes, analogous to the contextual invariants described in Chapter 6. Tail classes and open classes may then be separated by their mixtures over the head class invariants, rather than directly modeling their distributions from the raw data, thus mitigating data sparsity.

- *Generalizing to out-of-sample data and task distributions:* The residual adaptation strategies introduced in Chapter 6 and Chapter 7 are directly applicable to a broad range of problems involving neural layers. Parameter overheads are significantly reduced via distributionally regularized residual learning (Section 6.5), which can help align input distributions to account for feature variations.

- *Handling disentanglement and defining learning objectives:* Disentanglement objectives are best represented in inter-modular training objectives, analogous to those in Chapter 4 and Chapter 5. The modularity and generalizability of our solutions enable the integration of such objectives to avoid learning spurious correlations.

8.4.2 Long-Tail Knowledge-Graph Representations

Knowledge graphs (KGs) are a critical tool for backend data representations to support machine learning applications. Vast amounts of specialized domain-specific information (ranging from linguistics [220], biomedicine [42] to finance [27]) can be succinctly represented as a set of interacting entities (or KG nodes) and their **attributed** relationships (or KG edges)—see Figure 7.1. Each entity is semantically enriched by the rich transitive attributed associations to their entity neighborhoods, hence finding utility towards both, bridging data sparsity for individual entities, and providing a consistent or invariant representation towards multi-domain and multi-task settings (e.g., the KG embedding representations leveraged in Chapter 7).

Entity Representations: Entity representation models succinctly capture the entity neighborhoods and transitive attributed association structures with low-dimensional embedding representations of each node in the KG. The following are three popular classes of approaches to represent KG nodes:

- *Graph Convolutional Representations:* Graph convolution models adapt the convolution operation on regular grids (such as image pixels) to graph-structured data $\mathbf{G} = (V, E_{\mathbf{G}})$, learning low-dimensional vertex representations. Let N denote the number of vertices, and $\mathbf{X} \in \mathbb{R}^{N*d}$ the d-dimensional features of the vertices. The graph convolution operation for vertex $v \in V$ with features $\mathbf{X}_v \in \mathbb{R}^N$, and a learned filter g_θ in the fourier domain can be efficiently approximated with first-order terms [89] as follows,

  $$g_\theta * \mathbf{X}_v = \theta_0 \mathbf{X}_v + \theta_1 \left(\mathbf{L} - \mathbf{I}_N\right) \mathbf{X}_v \tag{8.3}$$

  with the normalized graph Laplacian, $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{A}$ denotes the adjacency matrix of graph $\mathbf{G}$ with N vertices and $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ is the corresponding diagonal degree matrix. The filter parameters, θ_0 and θ_1 are shared across all vertices.

  The above representation concepts can be extended along many dimensions to account for the knowledge graph's expressivity. For example, handling contrast and similarity relations [146], multi-relational tasks [176], positive and negative-edges [33], and motif-structures [177].

- *Link-completion heuristics:* Knowledge graph embedding techniques encode heuristic connectivity pattens in their embedding objectives [193], such as *symmetry/antisymmetry*, *composition*, and *inversion*, which can be stacked to encode higher-order linking patterns. We provide a technical summary of these heuristics in Section 7.3.1.

- *Higher-order structures:* The two primary higher-order structures associated with knowledge graphs include meta paths and motifs. Meta paths represent specific recurrent pathways in terms of the types, attributes, and precise sequences of nodes and relations between adjacent nodes in the path [38]. On the other hand, motifs are recurrent structures whose instances are identified via graph-isomorphism and leveraged to represent nodes based on the set of motifs they participate in [177].

Handling Long-Tail Entities: Long-tail data entities are those that are less commonly referenced within the knowledge graph in comparison to head entities, which act as hubs or central interlinking locations. Long-tail entities exhibit fewer connections to the remainder of the knowledge graph. Additionally, the concept of tail components is not limited to the nodes. It extends to the types of relations they exhibit (Chapter 7) and the types of structures and structural roles they participate in [177].

Bridging Sparsity via Higher-Order Structures: Role-aware models embed structurally similar nodes close in the latent space, independent of their precise network position [170]. While we can employ strict structural equivalence to embed nodes with identical local structures to the same point in the latent space [168], we can perform soft structural clustering based on the statistical measures (e.g., node degrees, motif count statistics) to transfer knowledge from dense to sparse nodes. In contrast to these methods, our prior work [177] contrastively learns attribute correlations in higher-order structures to identify correspondences between distant nodes, not just based on structure, but what each link in the structure represents based on the associated attribute values.

Connections to our Work: The concepts introduced in the preceding chapters can be applied bi-directionally to applications involving knowledge graphs, i.e., to improve the representations of nodes in the knowledge graph and to improve the quality of recommendation models using the knowledge graph representations.

- *Mining harder negative samples:* Training knowledge graph representations is best achieved via contrastive objectives [193] to distinguish positive associations (i.e., links in the graph) from missing ones (i.e., negatives). To generate suitable negative examples, a common method is to remove the correct tail entity and randomly sample it from a uniform distribution [228]. However, because the sampled entity may be semantically unrelated to the head entity and the relation, the quality of randomly generated negative examples is often poor. While existing resources such as ontologies may help generate better negatives, these resources are unavailable for specialized application domains. We may leverage the self-supervision method proposed by us in Chapter 4 to

filter among a candidate set of negative samples and pick the most informative samples at each gradient iteration, depending on the current node representations.

- *Improved node and link attribution:* In contrast to the data-sparsity problem, we may face a knowledge-sparsity where we do not know the precise reason for a specific link or interaction and hence struggle to attribute the cause to the entity neighborhood correctly. In such a scenario, we may leverage the adversarial attribution framework developed in Chapter 5 to learn a data-driven attribution function across the various higher-order structures and structural roles satisfied by the target entity.

- *Invariant identification:* In Chapter 7, we propose a task-independent base representation model for the knowledge graph, which is then distributionally altered in a task-specific manner via residual functions. While the base representation forms the task-invariant in our applications, we can modify the objectives to mine invariants. In other words, identify or reweight the precise higher-order structures, node neighborhoods, and relation types associated with nodes that best inform a specific task-model. These invariants may be identified from the set of dense nodes in the graph and then applied to the sparse nodes to benefit their respective task performance.

8.5 IMPROVEMENTS TO THE FRAMEWORK

A few interesting future directions include updating representation with streaming data and incorporating knowledge priors on expected behavior patterns (e.g., if we knew what combinations of context are more likely to dictate interactions) to benefit the learned context transformation space.

We also expect the following advances to significantly enhance the performance and applicability of the methods presented in this thesis - the development of adaptive samplers to produce informative *fake-pairs* to regularize the interest space with diverse objectives (such as those presented in Section 8.3) and speed up model convergence, enhanced contextual weighting with a fine-grained combination of the context projections, and finally, the development of efficient and expressive discriminator architectures for domain-specific multimodal applications. We expect a closer analysis of the specific interactions between the generator and discriminator architectures [187] to provide insights to improve the efficacy of our adversarial frameworks and avoid convergence to degenerate solutions.

8.5.1 Temporal Cluster Evolution

The frameworks designed in Chapter 4, Chapter 5, Chapter 6 and Chapter 7 focus a snapshot of the data, i.e., the models are applied to the training data without any notions of the temporal evolution of the associated users and other interacting entities. The grouping mechanism in Chapter 3 incorporates temporal profiling models, although the temporal parameters do not evolve but rather fit the entire data trajectory. Developing incremental models for streaming data could enable an application to real-time online platforms.

Our frameworks can be extended to streaming data via evolving representations. This solution assumes multiple snapshots of evolving data. Our models can then be sequentially updated in the following manner. We first learn parameters associated with the first data snapshot. The parameter estimates are then applied as the initialization to the second data snapshot and so on.

8.5.2 Incorporation of Domain Knowledge

Incorporating knowledge priors on expected behavior patterns can help guide our representations and avoid drift in the self-supervised frameworks in Chapter 4 and Chapter 5. Specifically, these constraints can be applied in the form of group priors in Chapter 3, item feature representations, and pre-defined regularizers in Chapter 4 as an additional objective, restricted or pre-defined context combinations as behavioral invariants in Chapter 6, and sequential task updates in Chapter 7.

While the above suggestions provide simple extensions to incorporate domain knowledge in our framework, a more fundamental approach would be to update the respective abstractions to incorporate specific realizations of domain knowledge, such as entity taxonomies.

8.5.3 Characterizing Effective Latent Space Representations

Our frameworks benefit from more explicit characterizations of users and other entities' desirable latent space representation properties. Specifically, the following criteria merit further investigation either as posthoc criteria for model evaluation or direct incorporation in the objective functions and training methodologies.

Ideally, we want to sample new data from the model representation space, potentially contributing to data augmentation strategies [138]. This requires representation in the latent space to be disentangled with respect to the data features. Each factor of variation in the data space can be mapped to specific dimensions in the latent space. We also want small and

meaningless noise in data not to be encoded in the latent space. Finally, we want smooth transitions in the latent space. This implies that the pairwise distances between data points are correlated between the raw feature space and the respective latent spaces learned by our frameworks.

8.6 OPEN PROBLEMS AND FUTURE WORK

We discuss four important open problems concerning neural networks and discuss how they apply to our work.

Explainability: This is one of the main concerns the deep-learning community currently faces. Owing to the complexities of the decision boundaries learned by neural models, it is hard to attribute decisions and predictions to human-interpretable concepts consistently.

This challenge extends to our frameworks as well. Example-based grounding strategies can help mitigate some of these concerns and offer insights into our modeling approaches' clusters and knowledge representations. Specifically, our frameworks can be extended to pick representative examples associated with each entity cluster, behavioral invariant, recommendation/inference task, or recommendation domain to offer empirical explanations of the learned representations.

Robust Neural Models: This is one other main concern the community is trying to address. Adding small amounts of targeted noise to the data point and feeding it to a trained neural model at inference leads to incorrect or inverted results. This vulnerability is referred to as an adversarial example. Although the problem of robust learning prevails in other machine learning areas, a direct extension to multimodal recommendation merits further investigation since the noise may be injected in any of the data modalities, and the resulting interaction effects are computationally complex to simulate or anticipate.

Developing a Theoretical Understanding of our Models' Decision Boundaries: We note that our proposed strategies' effectiveness depends on the choice of base recommender; gains will vary with the base neural architecture type (autoencoder; GCN). Our proposal aims to analyze regularization strategies empirically; We leave a rigorous theoretical characterization of our regularization technique's covariation with base neural recommender architectures for future work.

Data Augmentation and Model Construction: Data augmentation methods generate synthetic examples to increase the diversity of training examples, and data fine-tuning techniques adjust the input examples to fit a given model's decision boundary. In this proposal, we choose remaining architecture-agnostic rather than proposing specific architectural enhancements.

We acknowledge that examining the intricacies of neural recommenders and proposing architectural innovations is an alternative viable research direction to address the challenges imposed by data skew. We leave the exploration of such strategies as future work.

REFERENCES

[1] Himan Abdollahpouri and Robin Burke. Multi-stakeholder recommendation and its connection to multi-sided fairness. *arXiv preprint arXiv:1907.13158*, 2019.

[2] Himan Abdollahpouri and Robin Burke. Multi-stakeholder recommendation and its connection to multi-sided fairness. *arXiv preprint arXiv:1907.13158*, 2019.

[3] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9), 2018.

[4] David J Aldous, Illdar A Ibragimov, and Jean Jacod. *Ecole d'Ete de Probabilites de Saint-Flour XIII, 1983*, volume 1117. Springer, 2006.

[5] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

[6] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. Neural news recommendation with long-and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 336–345, 2019.

[7] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Engaging with massive online courses. In *Proceedings of the 23rd international conference on World wide web*, pages 687–698. ACM, 2014.

[8] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[9] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Factors of transferability for a generic convnet representation. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1790–1802, 2015.

[10] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *International Conference on Computational Learning Theory*, pages 35–50. Springer, 2007.

[11] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 301–304. ACM, 2011.

[12] Albert-Laszlo Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

[13] Jonathan Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 28(1):7–39, 1997.

[14] Alex Beutel, Kenton Murray, Christos Faloutsos, and Alexander J Smola. Cobafi: collaborative bayesian filtering. In *Proceedings of the 23rd international conference on World wide web*, pages 97–108. ACM, 2014.

[15] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 46–54. ACM, 2018.

[16] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2212–2220, 2019.

[17] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[18] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[19] Hongyun Cai, Vincent W Zheng, Fanwei Zhu, Kevin Chen-Chuan Chang, and Zi Huang. From community detection to community profiling. *Proceedings of the VLDB Endowment*, 10(7):817–828, 2017.

[20] Liwei Cai and William Yang Wang. Kbgan: Adversarial learning for knowledge graph embeddings. *arXiv preprint arXiv:1711.04071*, 2017.

[21] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The World Wide Web Conference (WWW)*, 2019.

[22] Francesco Paolo Casale, Adrian Dalca, Luca Saglietti, Jennifer Listgarten, and Nicolo Fusi. Gaussian process prior variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 10369–10380, 2018.

[23] Krishna Chaitanya, Neerav Karani, Christian F Baumgartner, Anton Becker, Olivio Donati, and Ender Konukoglu. Semi-supervised and task-driven data augmentation. In *International conference on information processing in medical imaging*, pages 29–41. Springer, 2019.

[24] Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-dimensional hyperbolic knowledge graph embeddings. *arXiv preprint arXiv:2005.00545*, 2020.

[25] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*, pages 1002–1012, 2017.

[26] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning for recommendation. *arXiv preprint arXiv:1802.07876*, 2018.

[27] Dawei Cheng, Fangzhou Yang, Xiaoyang Wang, Ying Zhang, and Liqing Zhang. Knowledge graph-based event embedding framework for financial quantitative investments. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, 2020.

[28] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4794–4802, 2019.

[29] Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*, 2015.

[30] Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26, 2006.

[31] Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *arXiv preprint arXiv:1812.04529*, 2018.

[32] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[33] Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 929–934. IEEE, 2018.

[34] Tyler Derr, Yao Ma, and Jiliang Tang. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 929–934. IEEE, 2018.

[35] Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. Finding bursty topics from microblogs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 536–544. Association for Computational Linguistics, 2012.

[36] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[37] Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1851–1860, 2017.

[38] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144, 2017.

[39] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. Time-sensitive recommendation from recurrent user activities. In *NIPS*, volume 15, pages 3492–3500, 2015.

[40] Zhengxiao Du, Xiaowei Wang, Hongxia Yang, Jingren Zhou, and Jie Tang. Sequential scenario-specific meta learner for online recommendation. *arXiv preprint arXiv:1906.00391*, 2019.

[41] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*, pages 278–288. International World Wide Web Conferences Steering Committee, 2015.

[42] Patrick Ernst, Amy Siu, and Gerhard Weikum. Knowlife: a versatile approach for constructing a large knowledge graph for biomedical sciences. *BMC Bioinformatics*, 16(1), 2015.

[43] Shanshan Feng, Gao Cong, Arijit Khan, Xiucheng Li, Yong Liu, and Yeow Meng Chee. Inf2vec: Latent representation model for social influence embedding. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 941–952. IEEE, 2018.

[44] Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. Initializing bayesian hyperparameter optimization via meta-learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[45] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[46] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth JF Jones. Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 795–798, 2015.

[47] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. Cross-domain recommendation without sharing user-relevant data. In *The World Wide Web Conference*, pages 491–502. ACM, 2019.

[48] Sheng Gao, Hao Luo, Da Chen, Shantao Li, Patrick Gallinari, and Jun Guo. Cross-domain recommendation via cluster-level latent factor model. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 161–176. Springer, 2013.

[49] Chase Geigle, Himel Dev, Hari Sundaram, and ChengXiang Zhai. A generative model for discovering action-based roles and community role compositions on community question answering platforms. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 181–192, 2019.

[50] Sharad Goel, Andrei Broder, Evgeniy Gabrilovich, and Bo Pang. Anatomy of the long tail: ordinary people with extraordinary tastes. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 201–210, 2010.

[51] Lovedeep Gondara. Medical image denoising using convolutional denoising autoencoders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 241–246. IEEE, 2016.

[52] Pinghua Gong, Jieping Ye, and Changshui Zhang. Multi-stage multi-task feature learning. *arXiv preprint arXiv:1210.5806*, 2012.

[53] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[54] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. Knowledge graph embedding with iterative guidance from soft rules. *arXiv preprint arXiv:1711.11231*, 2017.

[55] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE, 2008.

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[57] Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 161–169. ACM, 2017.

[58] Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 355–364. ACM, 2017.

[59] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 549–558. ACM, 2016.

[60] 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*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[61] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *arXiv preprint arXiv:1906.12340*, 2019.

[62] Tad Hogg and Gabor Szabo. Diversity of user activity and content quality in online communities. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 3, 2009.

[63] Danfeng Hong, Naoto Yokoya, Nan Ge, Jocelyn Chanussot, and Xiao Xiang Zhu. Learnable manifold alignment (lema): A semi-supervised cross-modality learning framework for land cover and land use classification. *ISPRS journal of photogrammetry and remote sensing*, 147:193–205, 2019.

[64] Seyed Abbas Hosseini, Ali Khodadadi, Keivan Alizadeh, Ali Arabzadeh, Mehrdad Farajtabar, Hongyuan Zha, and Hamid R Rabiee. Recurrent poisson factorization for temporal recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 32(1):121–134, 2018.

[65] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*, pages 193–201. International World Wide Web Conferences Steering Committee, 2017.

[66] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR, 2020.

[67] Guangneng Hu, Yu Zhang, and Qiang Yang. Transfer meets hybrid: A synthetic approach for cross-domain collaborative filtering with text. In *The World Wide Web Conference*, pages 2822–2829. ACM, 2019.

[68] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. Ieee, 2008.

[69] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

[70] Jian Huang, Ya Li, Jianhua Tao, Zheng Lian, Mingyue Niu, and Minghao Yang. Multimodal continuous emotion recognition with data augmentation using recurrent neural networks. In *Proceedings of the 2018 on Audio/Visual Emotion Challenge and Workshop*, pages 57–64, 2018.

[71] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. Knowledge graph embedding based question answering. In *International Conference on Web Search and Data Mining (WSDM)*, 2019.

[72] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM, 2010.

[73] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 2015.

[74] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In Dale Schuurmans and Michael P. Wellman, editors, *International Conference on Artificial Intelligence (AAAI)*, 2016.

[75] Yantao Jia, Yuanzhuo Wang, Hailun Lin, Xiaolong Jin, and Xueqi Cheng. Locally adaptive translation for knowledge graph embedding. In *International Conference on Artificial Intelligence (AAAI)*, 2016.

[76] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. Social contextual recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 45–54. ACM, 2012.

[77] Meng Jiang, Peng Cui, Fei Wang, Xinran Xu, Wenwu Zhu, and Shiqiang Yang. Fema: flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1186–1195. ACM, 2014.

[78] Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli. Degenerate feedback loops in recommender systems. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 383–390, 2019.

[79] Shuhui Jiang, Zhengming Ding, and Yun Fu. Deep low-rank sparse collective factorization for cross-domain recommendation. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 163–171. ACM, 2017.

[80] Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International Conference on Machine Learning (ICML)*, 2016.

[81] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[82] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE, 2018.

[83] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86. ACM, 2010.

[84] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018.

[85] Takeshi Kawabata and Masafumi Tamoto. Back-off method for n-gram smoothing based on binomial posteriori distribution. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 192–195. IEEE, 1996.

[86] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325*, 2016.

[87] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[88] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[89] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[90] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[91] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*, 2015.

[92] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[93] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[94] Adit Krishnan, Ashish Sharma, and Hari Sundaram. Improving latent user models in online social media. *arXiv preprint arXiv:1711.11124*, 2017.

[95] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. An adversarial approach to improve long-tail performance in neural collaborative filtering. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1491–1494. ACM, 2018.

[96] Adit Krishnan, Ashish Sharma, and Hari Sundaram. Insights from the long-tail: Learning latent representations of online user behavior in the presence of skew and sparsity. In *To appear in Proceedings of the 2018 ACM on Conference on Information and Knowledge Management*. ACM, 2018.

[97] Adit Krishnan, Hari Cheruvu, Cheng Tao, and Hari Sundaram. A modular adversarial approach to social recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1753–1762, 2019.

[98] Adit Krishnan, Mahashweta Das, Mangesh Bendre, Hao Yang, and Hari Sundaram. Transfer learning via contextual invariants for one-to-many cross-domain recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1081–1090, 2020.

[99] Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

[100] Matevž Kunaver and Tomaž Požrl. Diversity in recommender systems–a survey. *Knowledge-based systems*, 123:154–162, 2017.

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

[102] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. Pytorch-biggraph: A large-scale graph embedding system. *arXiv preprint arXiv:1903.12287*, 2019.

[103] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650. ACM, 2010.

[104] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, 2014.

[105] Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 891–900. ACM, 2014.

[106] Ang Li and Judea Pearl. Unit selection based on counterfactual logic. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 1793–1799, 2019.

[107] Bin Li, Qiang Yang, and Xiangyang Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[108] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1419–1428, 2017.

[109] Qibing Li, Xiaolin Zheng, and Xinyue Wu. Collaborative autoencoder for recommender systems. *ArXiv E-Prints*, 2017.

[110] Wentian Li. Zipf's law everywhere. *Glottometrics*, 5:14–21, 2002.

[111] Xiaopeng Li and James She. Collaborative variational autoencoder for recommender systems. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 305–314. ACM, 2017.

[112] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

[113] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 831–840. ACM, 2014.

[114] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, pages 951–961. International World Wide Web Conferences Steering Committee, 2016.

[115] 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*, pages 689–698. International World Wide Web Conferences Steering Committee, 2018.

[116] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *International Conference on Artificial Intelligence (AAAI)*, 2015.

[117] Fan Liu, Zhiyong Cheng, Changchang Sun, Yinglong Wang, Liqiang Nie, and Mohan Kankanhalli. User diverse preference modeling by multimodal attentive metric learning. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1526–1534. ACM, 2019.

[118] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 252–260. SIAM, 2013.

[119] Lu Liu, Jie Tang, Jiawei Han, Meng Jiang, and Shiqiang Yang. Mining topic-level influence in heterogeneous networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 199–208. ACM, 2010.

[120] Yan Liu, Alexandru Niculescu-Mizil, and Wojciech Gryc. Topic-link lda: joint models of topic and author community. In *proceedings of the 26th annual international conference on machine learning*, pages 665–672. ACM, 2009.

[121] Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3296–3305, 2019.

[122] Zemin Liu, Wentao Zhang, Yuan Fang, Xinming Zhang, and Steven CH Hoi. Towards locality-aware meta-learning of tail node embeddings on networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 975–984, 2020.

[123] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019.

[124] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[125] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org, 2017.

[126] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org, 2017.

[127] Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015.

[128] Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015.

[129] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[130] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[131] Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. A tri-role topic model for domain-specific question answering. In *AAAI*, pages 224–230, 2015.

[132] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Cross-domain recommendation: An embedding and mapping approach. In *IJCAI*, pages 2464–2470, 2017.

[133] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

[134] Benjamin M Marlin. Modeling user rating profiles for collaborative filtering. In *Advances in neural information processing systems*, pages 627–634, 2004.

[135] Peter V Marsden and Noah E Friedkin. Network studies of social influence. *Sociological Methods & Research*, 22(1):127–151, 1993.

[136] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[137] Lei Mei, Pengjie Ren, Zhumin Chen, Liqiang Nie, Jun Ma, and Jian-Yun Nie. An attentive interaction network for context-aware recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 157–166. ACM, 2018.

[138] Agnieszka Mikołajczyk and Michał Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018.

[139] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[140] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*, 2017.

[141] Thomas Minka. Estimating a dirichlet distribution, 2000.

[142] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42, 2007.

[143] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[144] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[145] Stephen L Morgan and Christopher Winship. *Counterfactuals and causal inference*. Cambridge University Press, 2015.

[146] Kanika Narang, Chaoqi Yang, Adit Krishnan, Junting Wang, Hari Sundaram, and Carolyn Sutter. An induced multi-relational framework for answer selection in community question answer platforms. *arXiv preprint arXiv:1911.06957*, 2019.

[147] Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. A capsule network-based embedding model for knowledge graph completion and search personalization. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2180–2189. Association for Computational Linguistics, 2019.

[148] Maximilian Nickel and Volker Tresp. An analysis of tensor models for learning on structured data. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, volume 8189 of *Lecture Notes in Computer Science*. Springer, 2013.

[149] Maximilian Nickel and Volker Tresp. Tensor factorization for multi-relational learning. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, volume 8190 of *Lecture Notes in Computer Science*. Springer, 2013.

[150] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. In *Twenty-fourth AAAI conference on artificial intelligence*, 2010.

[151] Rajiv Pasricha and Julian McAuley. Translation-based factorization machines for sequential recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 63–71. ACM, 2018.

[152] Dilruk Perera and Roger Zimmermann. Cngan: Generative adversarial networks for cross-network user preference generation for non-overlapped users. In *The World Wide Web Conference*, pages 3144–3150. ACM, 2019.

[153] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[154] Jim Pitman and Marc Yor. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, pages 855–900, 1997.

[155] Sanjay Purushotham, Yan Liu, and C-C Jay Kuo. Collaborative topic regression with social matrix factorization for recommendation systems. *arXiv preprint arXiv:1206.4684*, 2012.

[156] Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7229–7238, 2018.

[157] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.

[158] Jiezhong Qiu, Jie Tang, Tracy Xiao Liu, Jie Gong, Chenhui Zhang, Qian Zhang, and Yufei Xue. Modeling and predicting learning behavior in moocs. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 93–102. ACM, 2016.

[159] Minghui Qiu, Feida Zhu, and Jing Jiang. It is not just what we say, but how we say them: Lda-based behavior-topic model. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 794–802. SIAM, 2013.

[160] Qiang Qu, Cen Chen, Christian S Jensen, and Anders Skovsgaard. Space-time aware behavioral topic modeling for microblog posts. *IEEE Data Eng. Bull.*, 38(2):58–67, 2015.

[161] Xiaojun Quan, Chunyu Kit, Yong Ge, and Sinno Jialin Pan. Short and sparse text topic modeling via self-aggregation. In *IJCAI*, pages 2270–2276, 2015.

[162] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[163] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. *arXiv preprint arXiv:1810.03649*, 2018.

[164] Jinfeng Rao, Hua He, and Jimmy Lin. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1913–1916, 2016.

[165] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A DePristo, Joshua V Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *arXiv preprint arXiv:1906.02845*, 2019.

[166] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009.

[167] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820, 2010.

[168] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 385–394, 2017.

[169] Brian Roark, Murat Saraclar, and Michael Collins. Discriminative n-gram language modeling. *Computer Speech & Language*, 21(2):373–392, 2007.

[170] Ryan A Rossi, Di Jin, Sungchul Kim, Nesreen K Ahmed, Danai Koutra, and John Boaz Lee. From community to role-based graph embeddings. *arXiv preprint arXiv:1908.08572*, 2019.

[171] Lawrence A Rowe and Ramesh Jain. Acm sigmm retreat report on future directions in multimedia research. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 1(1):3–13, 2005.

[172] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[173] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018.

[174] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020.

[175] Dvir Samuel, Yuval Atzmon, and Gal Chechik. From generalized zero-shot learning to long-tail with class descriptors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 286–295, 2021.

[176] Aravind Sankar, Adit Krishnan, Zongjian He, and Carl Yang. Rase: Relationship aware social embedding. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[177] Aravind Sankar, Xinyang Zhang, and Kevin Chen-Chuan Chang. Meta-gnn: metagraph neural network for semi-supervised learning in attributed heterogeneous information networks. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 137–144, 2019.

[178] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 519–527, 2020.

[179] Aravind Sankar, Xinyang Zhang, Adit Krishnan, and Jiawei Han. Inf-vae: A variational autoencoder framework to integrate homophily and influence in diffusion prediction. *arXiv preprint arXiv:2001.00132*, 2020.

[180] Issei Sato and Hiroshi Nakagawa. Topic models with power-law using pitman-yor process. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 673–682. ACM, 2010.

[181] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[182] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

[183] Cosma Rohilla Shalizi and Andrew C Thomas. Homophily and contagion are generically confounded in observational social network studies. *Sociological methods & research*, 40(2):211–239, 2011.

[184] Tony Shen, Ariel Lee, Carol Shen, and C Jimmy Lin. The long tail and rare disease research: the impact of next-generation sequencing for rare mendelian disorders. *Genetics research*, 97, 2015.

[185] Chaitanya Shivade, Preethi Raghavan, and Siddharth Patwardhan. Addressing limited data for textual entailment across domains. *arXiv preprint arXiv:1606.02638*, 2016.

[186] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *arXiv preprint arXiv:1902.07379*, 2019.

[187] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3308–3318, 2017.

[188] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[189] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019.

[190] Erik B Sudderth and Michael I Jordan. Shared segmentation of natural scenes using dependent pitman-yor processes. In *NIPS*, pages 1585–1592. Citeseer, 2008.

[191] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer, 2016.

[192] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. *CoRR*, abs/1812.02391, 2018. URL `http://arxiv.org/abs/1812.02391`.

[193] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

[194] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. Recurrent knowledge graph embedding for effective recommendation. In *International Conference on Recommender Systems (RecSys)*, 2018.

[195] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[196] Jiliang Tang, Salem Alelyani, and Huan Liu. Feature selection for classification: A review. *Data classification: Algorithms and applications*, page 37, 2014.

[197] Tianyu Tang, Shilin Zhou, Zhipeng Deng, Huanxin Zou, and Lin Lei. Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. *Sensors*, 17(2):336, 2017.

[198] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 729–739. International World Wide Web Conferences Steering Committee, 2018.

[199] Yee Whye Teh. A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 985–992. Association for Computational Linguistics, 2006.

[200] John Towns, Timothy Cockerill, Maytal Dahan, Ian Foster, Kelly Gaither, Andrew Grimshaw, Victor Hazlewood, Scott Lathrop, Dave Lifka, Gregory D Peterson, et al. Xsede: accelerating scientific discovery. *Computing in Science & Engineering*, 16(5): 62–74, 2014.

[201] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. International Conference on Machine Learning (ICML), 2016.

[202] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. A meta-learning perspective on cold-start recommendations for items. In *Advances in neural information processing systems*, pages 6904–6914, 2017.

[203] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. A meta-learning perspective on cold-start recommendations for items. In *Advances in neural information processing systems*, pages 6904–6914, 2017.

[204] Panos Vassiliadis. A survey of extract–transform–load technology. *International Journal of Data Warehousing and Mining (IJDWM)*, 5(3):1–27, 2009.

[205] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[206] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[207] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *ICLR (Poster)*, 2019.

[208] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. Content-based neighbor models for cold start in recommender systems. In *Proceedings of the Recommender Systems Challenge 2017*, page 7. ACM, 2017.

[209] Hanna M Wallach, David M Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *Advances in neural information processing systems*, pages 1973–1981, 2009.

[210] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066. PMLR, 2013.

[211] Chong Wang and David M Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

[212] 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*, pages 515–524. ACM, 2017.

[213] Menghan Wang, Xiaolin Zheng, Yang Yang, and Kun Zhang. Collaborative filtering with social exposure: A modular approach to social recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[214] Rui Wang, Bicheng Li, Shengwei Hu, Wenqian Du, and Min Zhang. Knowledge graph embedding via graph attenuated attention networks. *IEEE Access*, 8:5212–5224, 2019.

[215] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Defu Lian. A survey on session-based recommender systems. *arXiv preprint arXiv:1902.04864*, 2019.

[216] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *International Conference on Knowledge Discovery & Data Mining (SIGKDD)*, 2019.

[217] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.

[218] Yaqing Wang, Chunyan Feng, Caili Guo, Yunfei Chu, and Jenq-Neng Hwang. Solving the sparsity problem in recommendations via cross-domain item embedding based on co-clustering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 717–725. ACM, 2019.

[219] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *International Conference on Artificial Intelligence (AAAI)*, 2014.

[220] Zhigang Wang, Juanzi Li, Zhichun Wang, Shuangjie Li, Mingyang Li, Dongsheng Zhang, Yao Shi, Yongbin Liu, Peng Zhang, and Jie Tang. Xlore: A large-scale english-chinese bilingual knowledge graph. In *International Semantic Web Conference (ISWC)*, 2013.

[221] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[222] Le Wu, Peijie Sun, Richang Hong, Yanjie Fu, Xiting Wang, and Meng Wang. Socialgcn: An efficient graph convolutional network based model for social recommendation. *arXiv preprint arXiv:1811.02815*, 2018.

[223] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. *arXiv preprint arXiv:1903.10433*, 2019.

[224] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 153–162. ACM, 2016.

[225] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617*, 2017.

[226] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[227] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 114–121. ACM, 2005.

[228] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[229] Carl Yang, Xiaolin Shi, Luo Jie, and Jiawei Han. I know you'll be back: Interpretable new user clustering and churn prediction on a mobile social application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 914–922, 2018.

[230] Chunfeng Yang, Huan Yan, Donghan Yu, Yong Li, and Dah Ming Chiu. Multi-site user behavior modeling and its application in video recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 175–184. ACM, 2017.

[231] Yezhou Yang, Ching Teo, Hal Daumé III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 444–454, 2011.

[232] Sirui Yao and Bert Huang. Beyond parity: Fairness objectives for collaborative filtering. *arXiv preprint arXiv:1705.08804*, 2017.

[233] Bee Wah Yap, Khatijahhusna Abd Rani, Hezlin Aryani Abd Rahman, Simon Fong, Zuraida Khairudin, and Nik Nik Abdullah. An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. In *Proceedings of the first international conference on advanced data and information engineering (DaEng-2013)*, pages 13–22. Springer, 2014.

[234] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. Challenging the long tail recommendation. *Proceedings of the VLDB Endowment*, 5(9):896–907, 2012.

[235] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. Lcars: a location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 221–229. ACM, 2013.

[236] Jianhua Yin and Jianyong Wang. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 233–242. ACM, 2014.

[237] Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. Geographical topic discovery and comparison. In *Proceedings of the 20th international conference on World wide web*, pages 247–256. ACM, 2011.

[238] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 785–788, 2013.

[239] Tong Zhao, Julian McAuley, and Irwin King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 261–270. ACM, 2014.

[240] Zhe Zhao, Zhiyuan Cheng, Lichan Hong, and Ed H Chi. Improving user topic interest profiles by behavior factorization. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1406–1416. International World Wide Web Conferences Steering Committee, 2015.

[241] Chenyi Zhuang and Qiang Ma. Dual graph convolutional networks for graph-based semi-supervised classification. In *Proceedings of the 2018 World Wide Web Conference*, pages 499–508, 2018.