Under review.

Interpretable recommender system with heterogeneous information: A geometric deep learning perspective

Yan Leng, Rodrigo Ruiz, Xiaowen Dong, Alex Pentland

Recommender systems (RS) are ubiquitous in the digital space. This paper develops a deep learning-based approach to address three practical challenges in RS: complex structures of high-dimensional data, noise in relational information, and the black-box nature of machine learning algorithms. Our method—Multi-Graph Graph Attention Network (MG-GAT)—learns latent user and business representations by aggregating a diverse set of information from neighbors of each user (business) on a neighbor importance graph. MG-GAT out-performs state-of-the-art deep learning models in the recommendation task using two large-scale datasets collected from Yelp and four other standard datasets in RS. The improved performance highlights MG-GAT's advantage in incorporating multi-modal features in a principled manner. The features importance, neighbor importance graph and latent representations reveal business insights on predictive features and explainable characteristics of business and users. Moreover, the learned neighbor importance graph can be used in a variety of management applications, such as targeting customers, promoting new businesses, and designing information acquisition strategies. Our paper presents a quintessential big data application of deep learning models in management while providing interpretability essential for real-world decision-making.

Key words: Recommender systems; Heterogeneous information; Matrix completion; Geometric deep learning; Representation learning; Interpretable machine learning

1. Introduction

Recommender systems (RS henceforth) are ubiquitous in the digital space*. The Big Data deluge, together with the rapid development of information technology, such as communication, mobile, and networking technologies, has proliferated business opportunities for companies and marketing departments (Rust and Huang 2014, Lee and Hosanagar 2020). Digital profiles of businesses and consumers have become unprecedentedly richer—a digital platform may collect users' behavioral information, social connections, and other demographic information. How to combine such heterogeneous information is, therefore, the key to designing effective recommender systems in e-commerce and beyond.

Despite their effectiveness in a wide range of business applications, existing RS face several challenges and opportunities. First, multi-modality has been increasingly common in big data, hence requiring RS to handle the complexities and utilize the information in a principled way. Second, there exist discrepancies between observed relational information, often in the form of networks (e.g., social networks), and the actual relevant relationship for effective predictions. This could result from network measurement errors or the heterogeneity of relationships, due to the strength of social ties (Granovetter 1977) and the context-dependent nature of social roles (Biddle 1986). Third, existing RS lack interpretability, especially those that are relying on high-performing black-box algorithms (Rudin and Carlson 2019). The fourth challenge is the typical cold-start problems in RS, causing new businesses to often be unfairly biased against (Li et al. 2020, Huang et al. 2007).

To address these challenges, we adopt in this paper techniques developed in the emerging field of geometric deep learning (Bronstein et al. 2017)—particularly the graph attention networks (Veličković et al. 2018)—and propose a novel framework for recommending businesses to users given heterogeneous information. Precisely, we consider a matrix completion problem, i.e., predicting the missing entries in a partially observed user-business rating matrix. To this end, we develop a geometric deep learning model to learn latent representations (embeddings in a low-dimensional space) for both users and businesses for the recommendation. The core idea is that these representations are obtained by aggregating a diverse set of information from the neighbors of each user or business in the respective network, where the predictive powers of the neighbors are weighted according to their relevance to the target user or business. In other words, neighbor importance is assigned to one's neighbors, which helps filter out noisy information and make the resulting model more effective and interpretable. We demonstrate the effectiveness of the proposed approach To show the generalization of the predictive performance, we further evaluate the performance of our method on four other standard data sets for recommendation tasks, including MovieLens, Douban, Flixster, and YahooMusic. We show that the proposed approach outperforms most state-of-the-art deep learning benchmarks, highlighting the advantage of incorporating heterogeneous information sources with relative importance. Furthermore, analysis and applications of the neighbor importance and latent embeddings demonstrate that our method can effectively extract interpretable patterns of the user-business interactions and characteristics of businesses.

Our paper makes five contributions to recommender systems. **First, our framework integrates heterogeneous information of different nature in a principled way.** We merge spatial, temporal, relational (network), and other types of data and selectively utilize the most relevant information for the recommendation task. Unlike methods relying on global feature smoothness (Leng et al. 2020a), we impose a local smoothness structure to better utilize the network information. Different from SVD++ (Koren 2008) and multi-view learning (Xu et al. 2013), we utilize auxiliary information to extract informative network connections (using graph attention networks) and model explicitly the interactions between different sources of information, e.g., auxiliary information and rating matrix. Our method is capable of handling both non-relational (features) and relational (network) information effectively by integrating these building blocks into a geometric deep learning model. In addition, the improved prediction performances on the Yelp and four other standard data sets for recommendation tasks (MovieLens, Douban, YahooMusic, and Flixster) suggest the effectiveness of the methods, as well as the meaningfulness of the learned graph and latent representations.

Second, our method is capable of unveiling informative relational information for both users and businesses. Our method is motivated by the weak tie theory (Granovetter 1977) and the role theory (Biddle 1986). Based on the auxiliary information and the ratings, we assign importance scores to the neighbors by computing the weights associated with network connections. In other words, rather than assuming all user-user (or business-business) relationships to reflect the actual social networks and all connections to contribute equally to the predictions, we learn the neighbor importance scores according to the relevance of the connections to the target predictions. This process is especially helpful in the presence of noisy or uninformative network connections, which are common in real-world applications. The performance improvement through heterogeneity in the neighbor importance graph shows that existing social network data in various online platforms may contain redundant relational information; hence it may not be the most effective if the observed information is used directly.

Third, the proposed algorithm accommodates an inductive learning framework to address the typical cold-start problem. Our method easily generalizes to a completely unseen data set. This is achieved via the learning of the importance of auxiliary information (business or user features) in generating the neighbor importance graph. Once computed, these feature relevance can be used to find “neighbors” for a new business or user in the existing data set, or to select important relationships in a completely new business or user network. This highlights the generalization capability of the proposed framework and extends the flexibility of RS significantly.

Fourth, our paper contributes to the growing interpretable machine learning literature for real-world decision-making in RS. Our framework enables interpretability from three aspects. First, neighbor importance enables the predictions to be more interpretable. Specifically, the framework reveals a selective subset of neighbors to the focal user or business, contributing to RS’s improved performance. Second, our method provides insights into which business features are more predictive, which accommodates feature selection in high-dimensional settings. For example, we found that location proximity, similarity in operation hours, parking facility, whether a business accepts credit cards, ambience and WiFi facility have high feature importance on the Yelp platform. Third, the learned latent representations capture the underlying characteristics of users (businesses) and reveal informative user (business) segmentation, which is useful in explaining the behavior of the RS. For example, the group structures for users reveal important features (e.g., parking facility and suitability for dinner) in distinguishing a high business rating from a low one. Additionally, business segmentation presents clear separation patterns in both business categories and rating distributions.

Finally, the feature importance and the neighbor importance graph generated by the proposed framework is immediately useful in various business contexts. This is possible as the neighbor importance graph extracted by the graph attention mechanism reveals the most predictive neighbors for each business and user. Therefore, our method yields unique ordering on businesses (users) for each user (business), contributing to personalized search rankings and targeting strategies. Moreover, the neighbor importance graph provides managerial insights on information acquisition by the Yelp platforms, enabled by investigating the centrally-positioned users/businesses. Beyond these applications, these underlying similarities have been shown to affect user behavior (Cheng et al. 2020) and product demand (Kumar and Hosanagar 2019).

In summary, our approach can be considered as a quintessential example of utilizing big data and state-of-the-art machine learning techniques for management science, especially for analyzing online digital platforms that contain rich, heterogeneous, high-dimensional, and in particular relational (network) information about users or businesses. Our method effectively handles noises and measurement errors in the data and provide managerial insights that lead to more informed decision-making in a variety of managerial applications.

The remainder of the paper is structured as follows. We review relevant literature in Section 2. Section 3 formulates the problem and introduces the proposed framework. We report the experimental results on Yelp, together with four other data sets in Section 4. We discuss the interpretability of the framework, as well as the managerial insights in Section 5. We demonstrate the managerial applications of our method in Section 6. Section 7 concludes.

2. Literature review

This paper grounds in three strands of works: recommender systems, geometric deep learning, and interpretable machine learning.

2.1. Recommender systems

Personalized recommendations have become commonplace due to the widespread adoption of RS. Major companies, including Amazon, Facebook, Netflix, and Yelp, provide users with recommendations on various products, including friends, movies, songs, and restaurants (Jannach et al. 2016, Bobadilla et al. 2013). There are two main approaches to building recommender systems: collaborative filtering (Van Roy and Yan 2010, Huang et al. 2007, Lee and Hosanagar 2020) and content-based filtering (Ansari et al. 2018, Panniello et al. 2016). Collaborative filtering recommends products based on the interests of users with similar ratings. In contrast, content-based filtering recommends products similar to other products in which the user has expressed interest.

With the availability of extensive data collected for both users and businesses, the design of recommender systems has increasingly taken into account extra information, such as friends, locations, user-generated contents, if available, to improve the quality of recommendation. Bao et al. (2015) presents a comprehensive overview of the recent progress in recommendation services in location-based social networks. The analyses combine the rating information with various data sources such as user profiles, user location trajectories (Ghose et al. 2015), social networks (Dewan et al. 2017), and geo-tagged social media activities (Ye et al. 2011, Cheng et al. 2012). Besides, some studies utilize click-through rate, click-stream data, and the products' rankings in the sponsored search to model user preferences (Agarwal et al. 2011, Chen and Yao 2017). Moreover, some studies utilize user-generated content to understand user preferences and make recommendations (Timoshenko and Hauser 2019). For example, Ansari et al. (2018) integrates product reviews, user-generated tags, and firm-provided covariate on MovieLens into a probabilistic topic model framework, which can infer the set of latent product features (topics) that not only summarizes the semantic content but is also predictive of user preferences.

Very recently, the success of neural networks makes its application to RS an active research field. Dziugaite and Roy (2015), He et al. (2017) present a straightforward extension to the traditional matrix factorization approach using multilayer perceptron. Rao et al. (2015a), Monti et al. (2017a) integrate networks with the rating information and propose to solve graph-regularized matrix completion with the deep learning frameworks. Some studies treat rating information as a user-business bi-partite graph and use graph autoencoder (van den Berg et al. 2017), higher-order connectivity in the networks (Wang et al. 2019), or in conjunction with user-user and business-business graph (Fan et al. 2019) to make recommendations. Except for the transductive methods mentioned above, some studies develop inductive methods for RS, which can be applied to unseen data sets. Hartford et al. (2018) uses exchangeable matrix layers to perform inductive matrix completion without using content information. Another very recent work, (Zhang and Chen 2020), uses one-hop subgraphs around user-item pairs to perform inductive matrix completion. The main differences between these approaches and our work are: (1) the attention-based mechanism adopted in our approach leads to several important advantages, both in performance and in the business implications; (2) the integration of heterogeneous information in a principled and selective fashion.

2.2. Geometric deep learning

The recent development in deep learning techniques (LeCun et al. 2015) has mostly advanced the state-of-the-art in a variety of machine learning tasks. Classical deep learning approaches are most successful on data with an underlying Euclidean or grid-like structure with a built-in notion of invariance. Real-world data, however, often come with a non-Euclidean structure such as networks and graphs. For example, the rating matrix in RS can be viewed as data associated with an underlying user-user or product-product similarity graph. It is not straightforward to generalize classical deep learning techniques to cope with such data, mostly due to the lack of well-defined operations such as convolutions on networks and graphs. To cope with these challenges, geometric deep learning (Bronstein et al. 2017) is a branch of emerging deep learning techniques that make use of novel concepts and ideas brought about by graph signal processing (Shuman et al. 2013), a fast-growing field by itself, to generalize classical deep learning approaches to data associated with networks and graphs.

Notable examples of early development in geometric deep learning include Bruna et al. (2014), Defferrard et al. (2016), Kipf and Welling (2017), where the authors have defined the convolution operation on graphs indirectly via the graph spectral domain by making use of a generalized notion of spectral filtering. In addition, Monti et al. (2017a) proposes a spatial-domain convolution on graphs using local patch operators represented as Gaussian mixture models. For more comprehensive reviews of geometric deep learning and graph neural network models, the reader is referred to a number of recent surveys (Hamilton et al. 2017, Battaglia et al. 2018, Wu et al. 2020). Out of the many successful applications, geometric deep learning techniques have been applied to the problem of matrix factorization with state-of-the-art performances in recommendation tasks (Monti et al. 2017b, van den Berg et al. 2017). This line of research inspired the present paper; however, two notable differences are that we use external auxiliary and network information, and we utilize an attention mechanism to enforce meaningful local smoothness constraints on the solutions.

Attention-based models are inspired by human perception (Mnih et al. 2014): instead of processing everything at once, we process the task-relevant information by selectively focusing our attention. Since their inception, attention-based models have been successfully applied to several different deep learning tasks. Attention has been shown to help identify the most task-relevant parts of an input, ignore the noise, and interpret results (Vaswani et al. 2017). Recently, attention mechanisms have been generalized for graph-structured data, which opens the possibility of designing various graph attention networks (Veličković et al. 2018, Lee et al. 2019). These graph-based attention models also enable the combination of data from multiple views (Shang et al. 2018): in addition to improving model performance with more task-relevant data, using data from multiple views improves model interpretability by allowing us to learn from the learned attention weights which views are the most task-relevant.

2.3. Interpretable machine learning

Despite the success achieved by sophisticated machine learning models, especially the deep neural networks (DNNs), in a wide variety of domains, their deployment in real-world scenarios and especially safety-critical applications remains hampered by the difficulty in understanding how the models make decisions and predictions. This has lead to an increasing interest in recent years in what is often called explainable artificial intelligence (XAI) or interpretable machine learning (IML) (Guidotti et al. 2018, Gilpin et al. 2018). What these nascent fields aim to achieve is to provide explanations or interpretations of how a machine learning model makes predictions to better aid decision-making.

The definition for an explanation or interpretation may vary and is likely to be domain and application-specific. In the management science community, interpretability has been realized in the context of text mining. For example, the work in (Lee et al. 2018) has proposed a method that automatically extracts interpretable concepts from text data and quantifies their importance to the business outcome of interest (e.g., conversion). Frameworks like this may help managers and businesses to make more informed decisions. Despite its importance, there is a lack of studies that address the interpretability issue in designing recommender systems. Such a limitation has important implications in terms of both user trust/experience and business decision-making.

Our study represents another direction towards deploying interpretable machine learning in business applications, particularly in recommender systems. Indeed, modeling the structure of the data with a graph could be a way of introducing prior knowledge that biases the system towards relational learning (Battaglia et al. 2018), which makes learning architectures such as geometric deep learning models inherently more interpretable than typical black-box models (e.g., the DNNs). More importantly, the attention-based mechanism adopted in our model provides additional interpretability via the learning of the relative importance of related users and businesses in making rating predictions. Such relative importance not only explains how the predictions are made but also yields business insights by themselves. We discuss these aspects in more detail in Section 5.

3. Model

In this section, we first use a simple example to motivate our framework design, which grounds in the weak tie theory (Granovetter 1977) and the role theory (Biddle 1986), as well as the multi-layer networks in physics (Kivelä et al. 2014). We then formulate the problem and introduce a new framework, called Multi-Graph Graph Attention Network (MG-GAT).

3.1. Motivating example and intuition

Our motivating example comes from a renewed interest in recent years in multi-layer networks (Kivelä et al. 2014), where the same set of nodes may share multiple types of relationships in different network layers. The heterogeneity in network connections ground in the weak tie theory (Granovetter 1977)[†] and the role theory (Biddle 1986)[‡] in sociology. Models of multi-layer networks have been adopted by sociologists to study different types of ties among individuals (Wasserman and Faust 1994, Scott 2012).

In existing RS and marketing applications, these different ties are usually collapsed into a single observed network (Ma et al. 2014, Goel and Goldstein 2014, Culotta and Cutler 2016, Sismeiro and Mahmood 2018); more specifically, they assume that the relationship observed corresponds to the actual underlying network. However, the actual interactions and the observed data may differ substantially due to errors in network measurements (Newman 2018) and the existence of multiple types of relationships (Granovetter 1977, Rishika and Ramaprasad 2019, Choi et al. 2020). Results may be unreliable if we assume all network connections are equivalent, but in fact, they are not (Aswani et al. 2018). Therefore, there is a need to develop methods robust to measurement errors and adaptable to the heterogeneity nature of networks. We proceed with a simple example to motivate our framework, as illustrated in Figure 1. We consider two types of networks: a professional network and a social network. A digital platform (e.g., Yelp or Tripadvisor) aims to infer consumers' preferences on restaurants to make personalized recommendations. Suppose that individuals form professional relationships based on their educational background and form a social relationship based on their hobbies. In this example, let us assume that only the social network provides relevant information on user preferences in restaurants. Typically, digital platforms cannot distinguish these two types of relationships. However, suppose we observe hobbies, this information can be useful in disentangling social relationships from professional relationships, which in turn will help with the prediction problem. This suggests that even if we cannot observe social and professional networks separately, we can leverage relevant auxiliary information and their predictive power over consumer behaviors to filter out the uninformative connections (e.g., professional relationships in this example) in the network effectively.

The example above provides the intuition on how to get informative network connections. Next, we focus on utilizing these informative connections to extract the latent user and business features. We continue with the illustration of users in Figure 2. The skeleton of our framework can be segmented into four steps. The motivating example above corresponds to the first and the second step, represented in Figure 2a and Figure 2b. The output from Figure 2b—the inferred connections with the importance score of each neighbor—is fed into Figure 2c to aggregate the features of

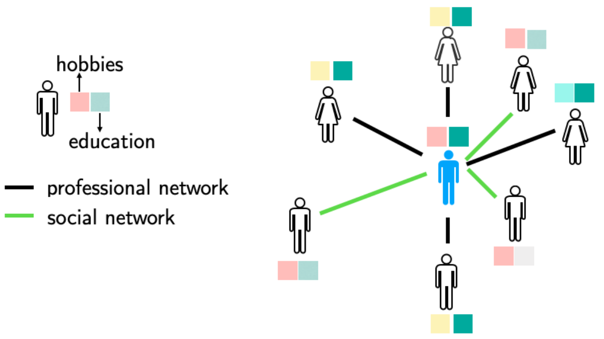


Figure 1 A motivating example. Each individual is characterized by hobbies and educational background. The focal user and neighbors are colored in blue and white. The black and green link correspond to professional and social relationships, respectively.

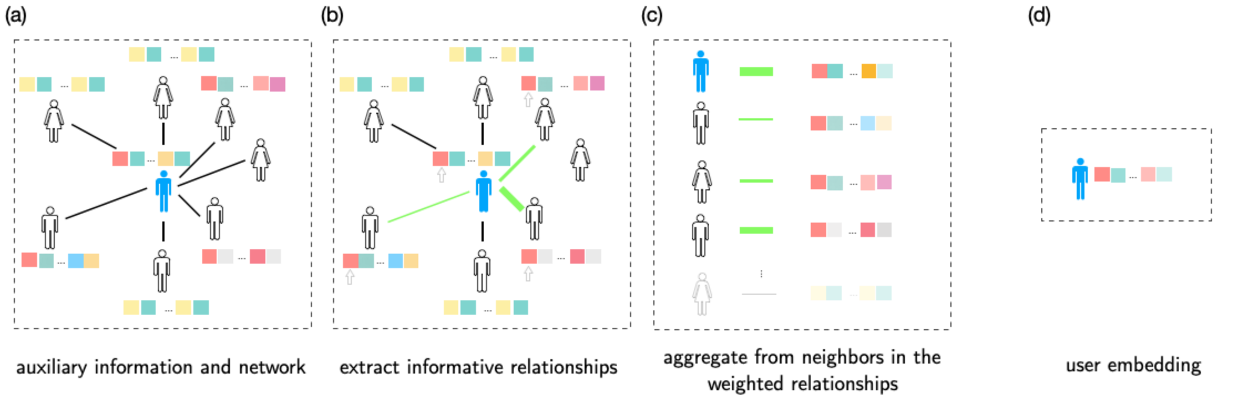

(a)
(b)
(c)
(d)
auxiliary information and network
extract informative relationships
aggregate from neighbors in the weighted relationships
user embedding

Figure 2 Skeleton of the process. We assume that the predictive feature are known beforehand (as pointed out with the arrow in (b)). The focal individual and neighbors are colored in blue and white. Our framework can be summarized in four steps. The first and second step extract the predictive neighbors. The third to fourth step aggregate node embeddings.

neighbors for the focal user in Figure 2d. Finally, we combine user embeddings with business embeddings (obtained in a similar way) to make rating predictions.

This simple illustration assumes that we know (1) the social network is informative, and (2) hobbies is a known predictive feature that can help extract informative connections. While the first can be realistically assumed in practice, the second is often not known beforehand. To cope with this challenge, we formalize how to learn predictive features and the corresponding connections using a new deep learning framework.

3.2. Problem formulation

We consider a partially observed user-business rating matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$, where n is the number of individuals, m is the number of businesses, and the ij-th entry X_{ij} is the rating of user i on business j, where $X_{ij} \in \{1, 2, 3, 4, 5\}$. We consider auxiliary information on users and businesses, which are denoted as $\mathbf{S}^{(u)} \in \mathbb{R}^{n\times s_u}$ and $\mathbf{S}^{(b)} \in \mathbb{R}^{m\times s_b}$, respectively, where s_u and s_b represent the number of the auxiliary features for users and businesses.

In addition to auxiliary information, relational information in the form of networks may further benefit the prediction task. For example, it would be reasonable to assume that businesses of similar categories or users who are friends of each other tend to have similar characteristics that are pertinent to the ratings. Motivated by this, we further utilize a user-user and a business-business network to capture the relationships among users and businesses. The friendship network collected on digital platforms can be used as the user network (e.g., the friendship network on Yelp or following relationship on Twitter), whose adjacency matrix is denoted as $\mathbf{G}_u$, and the corresponding combinatorial graph Laplacian matrix as $\mathbf{L}_u$. Similarly, the business network is denoted as $\mathbf{G}_b$ and the graph Laplacian matrix is $\mathbf{L}_b$ (see Section 4.1 for the exact definition).

Our objective is to infer user preferences on the businesses they have not yet rated, i.e., to complete the empty entries in $\mathbf{X}$ given the observed entries as well as the complementary information provided by $\mathbf{G}_u$, $\mathbf{G}_b$, $\mathbf{S}^{(u)}$, and $\mathbf{S}^{(b)}$. We cast this problem as a matrix completion problem given additional relational (network) and non-relational information (auxiliary information on nodes). One of the variants of the matrix completion problem is to find a low-rank matrix $\hat{\mathbf{X}}$ that matches the original matrix $\mathbf{X}$ conditioned on the observed entries. The notations used in this study are summarized in Table 1[§]. In practice, for robustness against noise as well as computational efficiency, the problem is often formulated as matrix factorization with the loss function $\mathcal{L}$:

$$\mathcal{L} = ||\mathbf{\Omega}_{\text{training}} \circ (\mathbf{X} - \mathbf{U}^T\mathbf{B})||_F^2 + \mathcal{R}(\mathbf{U}) + \mathcal{R}(\mathbf{B}), \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{n\times k}$ and $\mathbf{B} \in \mathbb{R}^{m\times k}$ are two latent representations to learn; k is the dimension of the latent representations; $\mathbf{\Omega}_{\text{training}}$ is the indicator matrix with one in the observed entries of $\mathbf{X}$ in the training set and zero otherwise; $\circ$ denotes the Hadamard product; and $||\cdot||_F$ denotes the Frobenius norm. The regularization terms $\mathcal{R}(\mathbf{U})$ and $\mathcal{R}(\mathbf{B})$ enforce additional constraints on the structure of $\mathbf{U}$ and $\mathbf{B}$. In our context, we interpret $\mathbf{U}$ as a latent representation that captures users' preferences on businesses, and $\mathbf{B}$ as a latent representation encodes the characteristics of businesses.

In the problem of Eq. (1), it is common to consider a regularization term $\mathcal{R}(\cdot)$ that enforces certain structural constraint on $\mathbf{U}$ and $\mathbf{B}$. Common forms of $\mathcal{R}(\cdot)$ include ℓ_2 norm (Frobenius norm for matrices), ℓ_1 norm, and smoothness with respect to some underlying network structure, which corresponds to a graph Laplacian based regularization (Cai et al. 2010). The basic idea of graph-based regularization is to make the latent representations of two users (businesses) close to each other if there exists a connection between them in the user (business) network (Li and Yeung 2009). In our context, for example, the smoothness of the latent user representation $\mathbf{U}$ can be promoted

Table 1 Key notations

Notations	Definitions and Descriptions
$\mathbf{X}$	User-business rating matrix
$\hat{\mathbf{X}}$	Predicted user-business rating matrix
$\mathbf{U}, \mathbf{B}$	Inferred latent user and business representations
$\mathbf{S}^{(u)}, \mathbf{S}^{(b)}$	Auxiliary information about users and businesses
$\mathbf{G}_u, \mathbf{G}_b$	User friendship network and business similarity network
$\mathbf{L}_u, \mathbf{L}_b$	Graph Laplacian of the user friendship network and business similarity network
$\alpha^u_{k\to i}, \alpha^b_{l\to j}$	Neighbor importance for user and business
$\mathbf{a}_{u,\text{self}}, \mathbf{a}_{b,\text{self}}$	Feature weights of focal nodes in node importance
$\mathbf{a}_{u,\text{nb}}, \mathbf{a}_{b,\text{nb}}$	Feature weights of neighbor nodes in node importance

with $\mathcal{R}(\mathbf{U}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} ||\mathbf{U}_{i\cdot} - \mathbf{U}_{j\cdot}||_2^2 \mathbf{G}_{u,ij}$. This is equivalent to setting $\mathcal{R}(\mathbf{U}) = \text{Tr}(\mathbf{U}^T \mathbf{L}_u \mathbf{U})$, where $\mathbf{L}_u$ is the graph Laplacian matrix of a user friendship network, and $\text{Tr}(\cdot)$ denotes the trace operator. This regularization also applies to businesses.

3.3. Multi-Graph Graph Attention Network

The smoothness constraint described above is a global one in the sense that it enforces the representations for every neighbor pair to be close across the entire network. This is a reasonable and widely adopted assumption in the signal processing and machine learning literature (Dong et al. 2016, Kalofolias et al. 2014, Leng et al. 2020a). However, as motivated above, the observed friendship network connections are not necessarily all meaningful or of equal importance in practical situations Practically, a local smoothness may be more appropriate, i.e., only a subset of friends would predict a given user's preference on a given behavior (Leng et al. 2020c).

In this paper, we propose to promote such local smoothness of $\mathbf{U}$ and $\mathbf{B}$ using the graph attention network (GAT) (Veličković et al. 2018), which allows for the modeling of heterogeneous relationships. As illustrated in Figure 2, the GAT places higher weights on neighbors who provide task-relevant information and lower weights on those who do not. In other words, neighbors are not weighted equally but instead by how they contribute to the recommendation (i.e., matrix completion) task. This leads to two key benefits of the proposed framework: (1) removing noisy connections and weighing relevant neighbors differently in the network; (2) revealing how information is aggregated via the weights on neighbors, to render the framework more interpretable.

We now explain the framework, named Multi-Graph Graph Attention Network (MG-GAT), in more detail. We describe the graph attention mechanism, which enforces local smoothness of latent representations in our framework. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ as the node set and $\mathcal{E}$ as the edge set, we first define three concepts: "node embedding", "edge embedding", and "graph attention".

Definition 1. **Node embedding** (Cai et al. 2018). A node embedding is a function $g_n : v_i \to \mathbb{R}^k$, which maps each node $v_i \in \mathcal{V}$ to a k-dimensional vector.

Definition 2. **Edge embedding** (Cai et al. 2018). An edge embedding is a function $g_e : e_{ik} \to \mathbb{R}^{k'}$, which maps each edge $e_{ij} \in \mathcal{E}$ to a k'-dimensional vector.

A user embedding matrix is denoted as $\mathbf{H}_u^T = [\mathbf{H}_{u,1}, \mathbf{H}_{u,2}, ..., \mathbf{H}_{u,n}]$ with $\mathbf{H}_{u,i} \in \mathbb{R}^{d_0^u}$, where n represent the number of users, and d_u represents the latent dimension. A business embedding matrix is denoted as $\mathbf{H}_b^T = [\mathbf{H}_{b,1}, \mathbf{H}_{b,2}, ..., \mathbf{H}_{b,m}]$ with $\mathbf{H}_{b,i} \in \mathbb{R}^{d_0^b}$, where m represent the number of businesses, and d_b represents the latent dimension.

We first impose a linear transformation (with a dense layer) on the auxiliary information to maintain the interpretability on these features.

$$\begin{aligned} \text{user: } \mathbf{H}_{u,i}^{(1)} &= \mathbf{W}_u^{(1)} \mathbf{S}_i^{(u)}, \\ \text{business: } \mathbf{H}_{b,j}^{(1)} &= \mathbf{W}_b^{(1)} \mathbf{S}_j^{(b)}, \end{aligned} \tag{2}$$

where $\mathbf{W}_u^{(1)} \in \mathbb{R}^{d_0^u \times s_u}, \mathbf{W}_b^{(1)} \in \mathbb{R}^{d_0^b \times s_b}$ are the learnable coefficients applied to every node; d_0^u and d_0^b are the dimension of node embedding for user and business after the first transformation.

Next, we discuss the graph attention mechanism used in our study. We first define graph attention and neighbor importance.

Definition 3. **Graph attention** (Lee et al. 2019) and **neighbor importance**. Graph attention is defined as a function, $\mathcal{A} : \{v_i\} \times N_i \to \alpha_{\cdot \to i} \in [0, 1]$, which assigns a weight to each node in a neighborhood N_i of a given node v_i. This weight is named as neighbor importance. From node v_i's perspective, neighbor importance ($\alpha_{\cdot \to i}$) determine how much attention to pay to a particular neighbor of v_i. Notice that $\alpha_{\cdot \to i} \in \mathbb{R}^{|N_i|}$ and $\sum_{k \in N_i} \alpha_{k \to i} = 1$. The neighbor importance between each user in $\mathbf{G}_u$ and business pair in $\mathbf{G}_b$ define the **neighbor importance graphs**.

A single GAT layer takes as input a user network $\mathbf{G}_u$ or business network $\mathbf{G}_b$ in which nodes represent users/businesses, as well as user/business embeddings. The output is a one-dimensional edge embedding. From the perspective of v_i, the attention mechanism first takes as input the current node embedding for v_i and its neighbors in $\mathbf{G}_u$ (or $\mathbf{G}_b$), and then compute an edge coefficient for each edge between v_i and a neighbor v_k.

Specifically, we perform graph attention on each user via a shared attention $\mathcal{A}(\cdot) : \mathbb{R}^{d_0^u} \times \mathbb{R}^{d_0^u} \to \mathbb{R}$. $\mathbf{a}_u = [\mathbf{a}_{u,\text{self}} || \mathbf{a}_{u,\text{nb}}] \in \mathbb{R}^{2d_0^u}$ is a coefficient vector for users and $\mathbf{a}_b = [\mathbf{a}_{b,\text{self}} || \mathbf{a}_{b,\text{nb}}] \in \mathbb{R}^{2d_0^b}$ is a coefficient vector for businesses. To ensure scalability and efficiency, this step is only performed on nodes that are neighbors on the input graph $\mathbf{G}_u$ and $\mathbf{G}_b$. The neighbor set of node i is denoted as N_i^u on the user graph and that of node j as N_j^b on the business graph. In the case where efficiency is not a concern, the attention mechanism can be performed on a fully connected graph, e.g., the attention mechanism is performed on every node pair separately. To make coefficients easily comparable across different nodes, we normalize the weights across all neighbors of v_i using a softmax function. The neighbor importance can be computed as summarizing the process described above.

$$\begin{aligned} \text{user: } v \rightarrow e : \alpha^u_{k\rightarrow i} &= \text{softmax}_k\Big(\text{LeakyReLU}\big(\mathbf{a}^T_u\,[\mathbf{H}^{(1)}_{u,i}||\mathbf{H}^{(1)}_{u,k}]\big)\Big), \\ \text{business: } v \rightarrow e : \alpha^b_{l\rightarrow j} &= \text{softmax}_l\Big(\text{LeakyReLU}\big(\mathbf{a}^T_b\,[\mathbf{H}^{(1)}_{b,j}||\mathbf{H}^{(1)}_{b,l}]\big)\Big), \end{aligned} \tag{3}$$

where $\cdot||\cdot$ represents concatenation; LeakyReLU is the Leaky Rectified Linear Unit activation function, which is adopted by Veličković et al. (2018) as the activation in the attention mechanism. $\alpha^u_{k\rightarrow i}$ and $\alpha^b_{l\rightarrow j}$ determine how much attention to be paid to a particular neighbor k (l) when updating information on i (j) for the user (business).

Next, we define feature weights, which is another key concept for the interpretability of our method.

Definition 4. **Feature weights**. Feature weights are defined as the coefficients of the auxiliary information in computing the neighbor importance. Feature weights for the focal node are $\mathbf{a}^T_{\text{u,self}}\mathbf{W}^{(1)}_u$ $(\mathbf{a}^T_{b,\text{self}}\mathbf{W}^{(1)}_b)$ for users (businesses), and feature weights for the neighboring nodes are $\mathbf{a}^T_{b,\text{nb}}\mathbf{W}^{(1)}_b$ $(\mathbf{a}^T_{b,\text{nb}}\mathbf{W}^{(1)}_b)$ for users (businesses).

The next step aggregates neighbors' embedding for the focal node, which is a mapping from edge embedding to node embedding using the obtained neighbor importance $\{\alpha^u_{k\rightarrow i}\}$ and $\{\alpha^b_{l\rightarrow j}\}$:

$$\begin{aligned} \text{GAT aggregation for users: } e \rightarrow v : \mathbf{H}^{(2)}_{u,i} &= \text{actv}_1\Big(\sum_{k\in N^u_i}\big(\alpha^u_{k\rightarrow i}\mathbf{H}^{(1)}_{u,k}\big) + \mathbf{b}^{(1)}_u\Big), \\ \text{GAT aggregation for businesses: } e \rightarrow v : \mathbf{H}^{(2)}_{b,j} &= \text{actv}_1\Big(\sum_{l\in N^b_j}\big(\alpha^b_{l\rightarrow j}\mathbf{H}^{(1)}_{b,l}\big) + \mathbf{b}^{(1)}_b\Big) \end{aligned} \tag{4}$$

where $\mathbf{b}^{(1)}_u \in \mathbb{R}^{d^u_0}$ and $\mathbf{b}^{(1)}_b \in \mathbb{R}^{d^b_0}$ are the bias term; N^u_i and N^b_j are the neighbors for node i on the user and node j on the business graph, respectively; $\text{actv}_1(\cdot)$ is an activation function for nonlinearity¶.

Next, we feed $\mathbf{H}^{(2)}_{u,i}$ and $\mathbf{H}^{(2)}_{b,j}$ into separate dense layers, specifically:

$$\begin{aligned} \text{user: } v \rightarrow v : \mathbf{H}^{(3)}_{u,i} &= \text{actv}_2\big(\mathbf{W}^{(2)}_u\mathbf{H}^{(2)}_{u,i} + \mathbf{W}^{(3)}_u\mathbf{S}^{(u)}_i + \mathbf{b}^{(2)}_u\big), \\ \text{business: } v \rightarrow v : \mathbf{H}^{(3)}_{b,j} &= \text{actv}_2\big(\mathbf{W}^{(2)}_b\mathbf{H}^{(2)}_{b,j} + \mathbf{W}^{(3)}_b\mathbf{S}^{(b)}_j + \mathbf{b}^{(2)}_b\big), \end{aligned} \tag{5}$$

where $\mathbf{W}^{(2)}_u \in \mathbb{R}^{k_f\times d^u_0}, \mathbf{W}^{(2)}_b \in \mathbb{R}^{k_f\times d^b_0}$ and $\mathbf{b}^{(2)}_u, \mathbf{b}^{(2)}_b \in \mathbb{R}^{k_f}$ are the learnable weights; $\mathbf{W}^{(3)}_u \in \mathbb{R}^{k_f\times s_u}$ and $\mathbf{W}^{(3)}_b \in \mathbb{R}^{k_f\times s_b}$; k_f is the pre-specified latent dimension; $\text{actv}_2(\cdot)$ is an activation function for nonlinearity

To obtain the final nodal embedding for user ($\mathbf{U}$) and business ($\mathbf{B}$), we perform the following:

$$\begin{aligned} \mathbf{U}_i &= \mathbf{H}^{(3)}_{u,i} + \mathbf{H}^{(4)}_{u,i}, \\ \mathbf{B}_j &= \mathbf{H}^{(3)}_{b,j} + \mathbf{H}^{(4)}_{b,j}, \end{aligned} \tag{6}$$

where $\mathbf{H}_u^{(4)} \in \mathbb{R}^{k_f}$ and $\mathbf{H}_b^{(4)} \in \mathbb{R}^{k_f}$ are additional embeddings learned independently from the attention mechanism. This adds further flexibility and expressivity to the final user and business embeddings.

The prediction of i's rating on business j can be formalized as,

$$\hat{X}_{ij} = \mathrm{norm}\big(\mathbf{U}_i\mathbf{B}_j^T + b_i^{(u)} + b_j^{(b)} + b_x\big), \tag{7}$$

where $\mathrm{norm}(x) = (r_{\max} - r_{\min}) \cdot \mathrm{sigmoid}(x) + r_{\min}$. The maximum and the minimum of the ratings are denoted as $r_{\max}$ and $r_{\min}$. The user-specific, business-specific, and global bias term are denoted as $b_i^{(u)}$, $b_j^{(b)}$, $b_x \in \mathbb{R}$.

3.4. Model training

We minimize the mean squared error between the predicted and the ground-truth ratings in the training set. We are now ready to present the loss function ($\mathcal{L}$) for the matrix completion task:

$$\mathcal{L} = ||\mathbf{\Omega}_{\text{training}} \circ (\mathbf{X} - \hat{\mathbf{X}})||_2^2 + \theta_1 L_{\text{reg}} \tag{8}$$

where θ_1 is a hyperparameter controlling the strength of the graph regularization term. Specifically, the graph regularization term can be written as,

$$L_{\text{reg}} = \mathrm{Tr}(\mathbf{H}_u^{(4)T}\tilde{\mathbf{L}}_u\mathbf{H}_u^{(4)}) + \mathrm{Tr}(\mathbf{H}_b^{(4)T}\tilde{\mathbf{L}}_b\mathbf{H}_b^{(4)}), \tag{9}$$

where $\tilde{\mathbf{L}}_u = \mathbf{L}_u + \theta_2\mathbf{I}$ and $\tilde{\mathbf{L}}_b = \mathbf{L}_b + \theta_2\mathbf{I}$ are the regularized Laplacian (Zhou et al. 2012, Smola and Kondor 2003) with $\mathbf{I}$ being the identity matrix and θ_2 as a hyperparameter. As a standard training strategy in the deep learning literature, we also impose ℓ_2 regularization on all the learnable parameters $(\mathbf{W}_u^{(1)}, \mathbf{W}_u^{(2)}, \mathbf{W}_u^{(3)}, \mathbf{W}_b^{(1)}, \mathbf{W}_b^{(2)}, \mathbf{W}_b^{(3)}, \mathbf{b}_u^{(1)}, \mathbf{b}_u^{(2)}, \mathbf{b}_b^{(1)}, \mathbf{b}_b^{(2)}, \mathbf{a}_u, \mathbf{a}_b)$. We omit this in the loss function for simplicity. Note that $\mathbf{H}_u^{(3)}$ and $\mathbf{H}_b^{(3)}$ are the components of $\mathbf{U}$ and $\mathbf{B}$ that are not regularized with the graph regularization term, since the graph attention framework in Eq. (4) enables local smoothness. The global smoothness is imposed on the other part of the final embedding (i.e., $\mathbf{H}_u^{(4)}$ and $\mathbf{H}_b^{(4)}$), which is enforced by the regularized Laplacian quadratic from (i.e., $\mathrm{Tr}(\mathbf{H}_u^{(4)}\tilde{\mathbf{L}}_u\mathbf{H}_u^{(4)})$). This promotes the behavior of the algorithm that similar users or businesses are mapped to close-by positions in the latent spaces.

To solve the problem of Eq. (8), the framework we proposed can be summarized in Figure 3. We name the proposed framework as Multi-Graph Graph Attention Network (MG-GAT). The MG-GAT architecture consists of three layers in sequence: a linear dense layer, a GAT layer, and a dense layer. The first layer takes the networks and the auxiliary information as input and outputs the node embedding linearly (Eq. (2)). The node embeddings are fed into the GAT layer

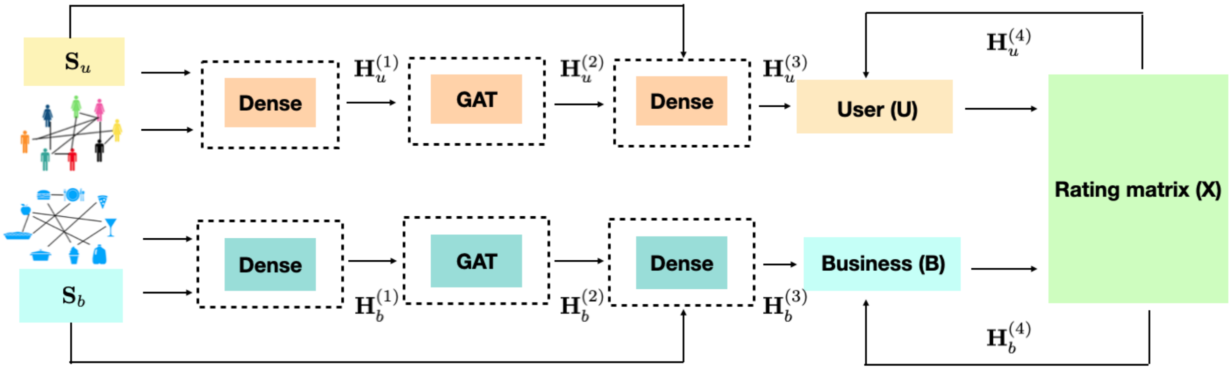


Figure 3 The proposed geometric deep learning architecture for learning latent user embeddings (U) and business embeddings (B), via iterations over four layers: a linear dense layer, a GAT layer, a dense layer, and a final aggregation layer.

to compute the neighbor importance, which is used to aggregate neighbor embeddings (Eq. (3)–(4)). The output embeddings from the GAT layer and a linear transformation on the auxiliary information are then aggregated for a nonlinear transformation (Eq. (5)) Lastly, we aggregate the output from the previous layer and additional embeddings learned from the rating matrix for the final embeddings (Eq. (6)). The final embeddings from users and businesses are used to compute the rating matrix (Eq. (7)) and the loss (Eq. (8)). We use Adam stochastic optimization to train the model and learn the parameters (Kingma and Ba 2014). We present the steps for updating $\mathbf{U}, \mathbf{B}$, and $\hat{\mathbf{X}}$ in Algorithm 1.

3.5. Comparisons with existing methods in the literature

Our method is in the vein of graph regularized non-negative matrix factorization and shares similarities with regression and graph convolutional network. We now comment on the similarities and major differences.

Graph regularized non-negative matrix factorization Our framework extends graph-regularized matrix factorization, and similarly maps users and businesses to a latent space (Cai et al. 2010). However, the major difference is that our framework integrates auxiliary information on nodes (i.e., users and business) and the connections between nodes in a principled fashion. We use auxiliary information to remove noisy links and then use this to perform localized aggregation on user and business embeddings. This process enables us to perform feature and link selection to optimize predictive power.

Regression The feature selection process involved in the GAT (Eq. (3)) bears conceptual similarity to linear regression; in particular, the former can be viewed as a nonlinear regression where the “dependent variable” is the training loss of the neural network architecture. Nevertheless, the GAT architecture enables several unique advantages: (1) it can capture any nonlinear relationship

Algorithm 1 Multi-Graph Graph Attention Network (MG-GAT)

```
input $\mathbf{X}$, $\mathbf{S}_u$, $\mathbf{S}_b$, $\mathbf{G}_u$, $\mathbf{G}_b$, $k_f, \theta_1, \theta_2$, $t_T$
for $t = 0 : t_T$ do
    Update user embedding
    for $i = 1 : n$ do
        $\mathbf{H}_{u,i}^{(1)(t)} = \mathbf{W}_u^{(1)(t)} \mathbf{S}_i^{(u)}$
        for $k \in N_i^u$ do
            $\alpha_{k \to i}^{u(t)} = \text{softmax}_k \Big( \text{LeakyReLU} \big( \mathbf{a}_u^{(t)T} \big[ \mathbf{H}_{u,i}^{(1)(t)} || \mathbf{H}_{u,k}^{(1)(t)} \big] \big) \Big)$
        $\mathbf{H}_{u,i}^{(2)(t)} = \text{actv}_1 \Big( \sum_{k \in N_i^u} \big( \alpha_{k \to i}^{u} \mathbf{H}_{u,k}^{(1)(t)} \big) + \mathbf{b}_u^{(1)(t)} \Big)$
        $\mathbf{H}_{u,i}^{(3)(t)} = \text{actv}_2 \big( \mathbf{W}_u^{(2)(t)} \mathbf{H}_{u,i}^{(2)(t)} + \mathbf{W}_u^{(3)(t)} \mathbf{S}_i^{(u)} + \mathbf{b}_u^{(2)(t)} \big)$
        $\mathbf{U}_i^{(t)} = \mathbf{H}_{u,i}^{(3)(t)} + \mathbf{H}_{u,i}^{(4)(t)}$
    Update business embedding
    for $j = 1 : m$ do
        $\mathbf{H}_{b,j}^{(1)(t)} = \mathbf{W}_b^{(1)(t)} \mathbf{S}_j^{(b)(t)}$
        for $l \in N_j^b$ do
            $\alpha_{l \to j}^{b(t)} = \text{softmax}_l \Big( \text{LeakyReLU} \big( \mathbf{a}_b^{(t)T} \big[ \mathbf{H}_{b,j}^{(1)(t)} || \mathbf{H}_{b,l}^{(1)(t)} \big] \big) \Big)$
        $\mathbf{H}_{b,j}^{(2)(t)} = \text{actv}_1 \Big( \sum_{l \in N_j^b} \big( \alpha_{l \to j}^{b(t)} \mathbf{H}_{b,j}^{(1)(t)} \big) + \mathbf{b}_b^{(1)(t)} \Big)$
        $\mathbf{H}_{b,j}^{(3)(t)} = \text{actv}_2 \big( \mathbf{W}_b^{(2)(t)} \mathbf{H}_{b,j}^{(2)(t)} + \mathbf{W}_b^{(3)(t)} \mathbf{S}_j^{(u)} + \mathbf{b}_b^{(2)(t)} \big)$
        $\mathbf{B}_j^{(t)} = \mathbf{H}_{b,j}^{(3)(t)} + \mathbf{H}_{b,j}^{(4)(t)}$
    $\hat{X}_{ij}^{(t)} = \text{norm} \big( \mathbf{U}_i^{(t)} \mathbf{B}_j^{(t)T} + b_i^{(u)(t)} + b_j^{(b)(t)} + b_x^{(t)} \big)$
output $\mathbf{U}^{(t_T)}, \mathbf{B}^{(t_T)}$, $\hat{\mathbf{X}}^{(t_T)} = \mathbf{U}^{(t_T)} \mathbf{B}^{(t_T)T}$
```

between the business/user features and the rating matrix; (2) it produces neighbor importance and latent business/user embeddings that may be further analyzed for additional business insights (Section 5); (3) it enables novel business applications, as demonstrated in Section 6.

Graph convolutional network GAT is in the realm of geometric deep learning. Hence, we compare our method with graph convolutional network (GCN), one of the most popular geometric deep learning models. GCN is developed from a spectral graph filtering perspective where the layer-wise propagation matrix comes essentially from a degree-one polynomial of the graph Laplacian matrix. This makes the smoothing or regularization in GCN a global operation (due to the global nature of the graph Fourier transform (Shuman et al. 2013)). In comparison, GAT can be interpreted more as a local smoothing framework, offering more flexibility.

In GCN, the weight matrix used for layer-wise propagation is $\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{G}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$, where $\tilde{\mathbf{G}} = \mathbf{G} + \mathbf{I}$ and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{G}}_{ij}$. The adjacency matrix of the graph, $\mathbf{W}$, is fixed and non-adaptive. In the case of an unweighted graph (where all edges have unitary weight, which is the case we consider for both the business and user networks), this leads to an equal weighting of all neighbors for each node. In comparison, via an attention mechanism, GAT allows for adapting the weights of the neighbors to the data hence a more flexible and complex neighborhood structure. This leads to several advantages in terms of both performance and interpretability.

Besides, GCN is an inherently transductive framework, while GAT can be both transductive and inductive. Specifically, GCN learns the nodal representations on an observed graph; hence the learned weights in the framework cannot be generalized to a completely unseen graph with new nodes. In comparison, GAT is not restricted to the observed (training) graph. It learns the weights for different nodal auxiliary features; hence the learned weights can be applied to an unseen graph. This is an important feature of our framework, which has significant managerial implications. We demonstrate this benefit in the experimental section by applying the framework to analyzing businesses unseen during the training process.

4. Empirical results

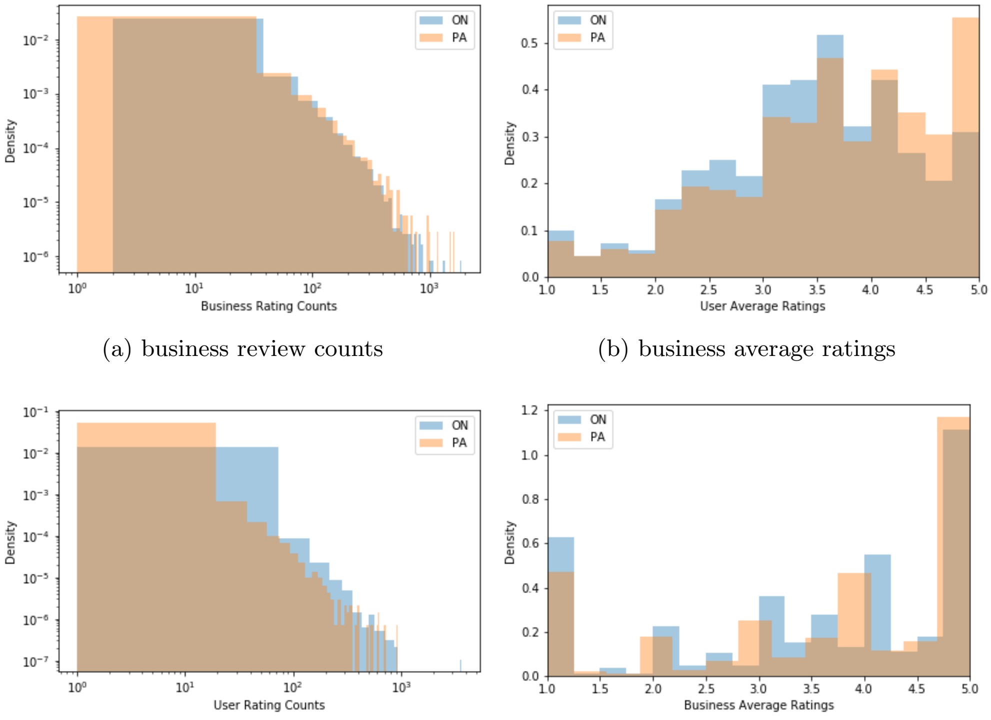
This section introduces the main data sets, Yelp, used in this paper for empirical analysis. We also describe the experimental setup and the performance evaluations, comparing with other state-of-the-art deep learning methods both on Yelp and on four other standard data sets commonly used for recommendation tasks(i.e., MovieLens, Flixster, Douban, and YahooMusic). We conclude this section to test the contributions of different components of our framework.

4.1. Data descriptions

We utilize the data set provided by Yelp[‖], an online review platform where users may rate and post reviews on businesses (e.g., restaurants, bars, spas). The data was collected from 2009 to 2018. We focus the analysis on Ontario (ON) in Canada and Pennsylvania (PA) in the United States. The two states have 135173/76865 users and 32393/10966 businesses, respectively. The density of nonzeros in the matrix is 0.0161%/0.0309%. We summarize the statistics of the ratings for the businesses in Table 2. We show the distributions of the number of reviews of each business and user in Figure 4a and 4c, respectively, both of which follow a power-law distribution. The distributions of the average ratings of businesses and users are shown in Figure 4b and 4d.

Table 2 **Summary statistics of the data.**

	Ontario	**Pennsylvania**
Rating count	706,998	260,350
User count	135,173	76,865
Business count	32,393	10,966
Average rating (std.)	3.556 (1.334)	3.728 (1.384)
Ratings per user (std.)	5.230 (21.007)	3.387 (12.140)
Ratings per business (std.)	21.826 (47.221)	23.742 (56.371)



(a) business review counts

(b) business average ratings

(c) user review counts

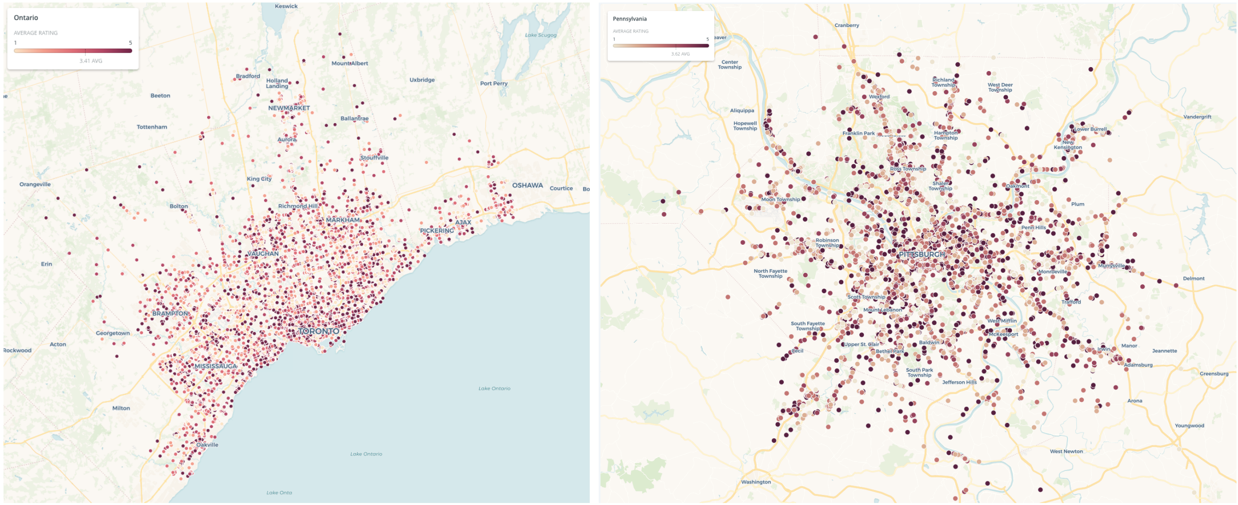
(d) user average ratings

Figure 4 Distributions of review counts and average ratings for businesses and users.

Business Yelp collects business information via both self-uploaded information from business owners and surveys from users. The rich and high-dimensional information of different nature makes it an ideal data set for testing the effectiveness of the proposed method. Roughly speaking, there are three types of information about the businesses, i.e., basic information (attributes, categories, and operation hours), location information, and check-in information (temporal popularity).

The basic information collected by Yelp consists of business attributes (features related to amenities), business categories, and operation hours. Yelp collects different information in Canada and the US, with eighty-four and ninety-three attributes for ON and PA, separately. Business attributes cover information such as the provision of parking space, WiFi hotspot, and takeout service. Since most attributes are categorical variables, we adopt one-hot (i.e., one-of-K) encoding indicating whether the business possesses a particular business attribute. There are 953 and 946 business categories in ON and PA, e.g., Mexican, burgers, gastropubs. Each business may belong to multiple business categories. We similarly adopt the one-hot encoding on business categories. The operation hours contain information about when businesses open and close.

The location information, in latitude and longitude, allows us to locate the businesses on the map, as shown in Figure 5. Due to the limits in human mobility, spatially-proximate businesses tend to attract more similar customers than farther ones (Kaya et al. 2018, Leng et al. 2018).



(a) Ontario

(b) Pennsylvania

Figure 5 Spatial distributions of businesses in (1) Ontario and (2) Pennsylvania. The color code represents the average ratings of businesses.

The check-in information from users allows us to analyze the temporal patterns of the popularity of the businesses. We aggregate the check-ins into 144 hourly bins of a week (24 h × 7 days) to obtain one check-in vector for each business.

Due to the lack of external relational information, we build a business network $\mathbf{G}_b$ in which an edge between businesses i and j is constructed if they belong to the same business category. The underlying assumption is that characteristics for businesses within one category are similar. As a sparse graph is needed for memory and computational efficiency in the case of large-scale graphs, we construct a k-nearest neighbor graph ($\mathbf{G}_b$) One rule-of-thumb is to set $k \sim \log(m)$, where m is the number of businesses (Von Luxburg 2007). In our case, k is set to be ten.

User User information can be categorized into two types, i.e., the basic metadata and friendships on Yelp. The basic metadata on each user includes whether they are elite in a certain year, the number of "useful", "funny", and "cool" reviews, number of fans, number of compliments on reviews as being "hot", "cute", "plain", "cool", "funny", or "good writer", and number of compliments on the user's profile, lists, notes, photos, and other information. This auxiliary information leads to nineteen attributes for each user in ON and thirty-three features for each user in PA.

Regarding the social network data, users can "friend" each other on Yelp. The data provides a list of Yelp users as friends of each given user. With this information, we can build a friendship network, where a connection on the network indicates that the two users are friends.

Implicit features Explicit feedback describes user rating behaviors (in ordinal or continuous variables), and implicit feedback captures users' interactions with businesses (in binary variables). In addition to the features mentioned above, we extract the implicit features from the rating matrix

Table 3 Statistics about the training, validation and test set

	Time period	Metrics	Ontario	Pennsylvania
Training	2009 - 2016	Average user rating	3.468 (1.373)	3.641 (1.403)
		Average business rating	3.419 (0.995)	3.604 (1.049)
Validation	2017	Average user rating	3.463 (1.479)	3.665 (1.501)
		Average business rating	3.401 (1.266)	3.614 (1.285)
Test	2018	Average user rating	3.469 (1.502)	3.667 (1.535)
		Average business rating	3.395 (1.314)	3.616 (1.373)

by treating the continuous ratings as binary features. This feature is inspired by the Netflix Prize, which concludes that harnessing implicit feedback into rating predictions was highly predictive and outperformed vanilla matrix factorization (Rendle et al. 2019). Using implicit features from implicit feedback relies on assuming that users have stronger preferences on businesses they have visited and rated than those they have not. We perform singular value decomposition on the binarized rating matrix (i.e., the entries are converted to one if rated, and zero otherwise) to obtain the implicit features. We denote the binarized rating matrix as $\mathbf{X}_{\text{bina}} = \boldsymbol{\Omega}_{\text{training}} \circ \mathbf{X} = \mathbf{U}^{(0)}\Sigma\mathbf{B}^{(0)}$, where $\boldsymbol{\Omega}_{\text{training}}$ is a indicator matrix indexing all entries in the training set; $\mathbf{U}^{(0)} \in \mathbb{R}^{n\times k_i}, \Sigma \in \mathbb{R}^{k_i\times k_i}, \mathbf{B}^{(0)} = \mathbb{R}^{k_i\times m}$. k_i is tuned as a hyperparameter. The implicit features on users and businesses are then computed as $\mathbf{S}_{u,\text{imp}} = \mathbf{U}^{(0)}\Sigma^{\frac{1}{2}}$ and $\mathbf{S}_{b,\text{imp}} = \mathbf{B}^{(0)T}\Sigma^{\frac{1}{2}}$.

All the features mentioned above are summarized into the auxiliary information matrix, $\mathbf{S}_b$ for business, and $\mathbf{S}_u$ for users, respectively.

4.2. Experimental setting

We split the rating data into training, validation, and test sets according to time of the ratings, i.e., the ratings between the year 2009 and 2016 are used as the training set, the ratings in the year 2017 are used as the validation set, and ratings in 2018 are used as the test set. We present the statistics about the train, test, and split of the data sets in Table 3. We use Hyperopt, a distributed Bayesian optimization implemented as a Python package (Bergstra et al. 2013) to search for the hyperparameters**.

We use the average Root Mean Squared Error (RMSE) as the performance metric:

$$\text{RMSE} = \sqrt{\frac{||\boldsymbol{\Omega}_{\text{test}} \circ (\mathbf{X} - \hat{\mathbf{X}})||_2^2}{||\boldsymbol{\Omega}_{\text{test}}||_1}} = \sqrt{\frac{||\boldsymbol{\Omega}_{\text{test}} \circ (\mathbf{X} - \mathbf{U}\mathbf{B}^T)||_2^2}{||\boldsymbol{\Omega}_{\text{test}}||_1}}, \tag{10}$$

where $\boldsymbol{\Omega}_{\text{test}}$ is the indicator matrix with one for the entries in the test set and zero otherwise, and $||\cdot||_1$ represents entry-wise ℓ_1 norm.

In addition to RMSE, we also use three other ranking-based metrics, Spearman's rank-order Correlation (Spearman's correlation), Bayesian personalized ranking (BPR) (Rendle et al. 2012), and Fraction of Concordant Pairs (FCP) (Koren and Sill 2013). These ranking-based metrics are defined on a pair level, instead of on an instance level (one rating) in RMSE. Spearman's correlation is defined by comparing the predicted and the actual ratings for all prediction pairs. BPR is a pairwise personalized ranking loss that is derived from the maximum posterior estimator. BPR is computed as,

$$\mathrm{BPR} = e^{\frac{1}{|D_{\text{test}}|}\sum_{i,j,k \in D_{\text{test}}} \ln\sigma\,(\hat{X}_{ij} - \hat{X}_{ik})},$$

where D_{test} is the set of test data and consists of pairs where $X_{ij} \geq X_{ik}$, for all u. $\sigma(\cdot)$ represents the sigmoid function. The defined BPR represents the geometric mean of the data likelihood. A higher BPR corresponds to better personalized ranking.

FCP measures the correct ranked business-pairs in recommender systems. The number of correct-ranked (concordant) business pairs by predicted ratings is $n_c^i = |\{(j,k)|\hat{X}_{ij} \leq \hat{X}_{ik} \text{ and } X_{ij} \leq X_{ik}\}|$. The number of discordant pair for user i is defined in a similar fashion, $n_d^i = |\{(j,k)|\hat{X}_{ij} < \hat{X}_{ik} \text{ and } X_{ij} \leq X_{ik}\}|$. FCP is therefore defined as,

$$\mathrm{FCP} = \frac{\sum_{i=1}^{N} n_c^i}{\sum_{i=1}^{N} n_c^i + \sum_{i=1}^{N} n_d^i}.$$

A higher FCP represents more concordant pairs.

4.3. Learning performance

We consider one non-deep learning approach SVD++ (Koren 2008), which is a strong benchmark shown to outperform all other commonly-adopted non-deep-learning methods (e.g., SVD, NMF, Slope One, k-NN and its variations, Co-Clustering) on the Movielens datasets (Hug 2020). Moreover, this method has been shown to outperform some recent state-of-the-art deep learning methods (e.g., sRGCNN (Monti et al. 2017b), GRALS (Rao et al. 2015b)), and on par with F-EAE (Hartford et al. 2018), on the MovieLens100K dataset.

We also consider the following state-of-the-art deep learning models, including GRALS (Rao et al. 2015b), NNMF (Dziugaite and Roy 2015), F-EAE (Hartford et al. 2018), sRGCNN (Monti et al. 2017b), GC-MC (Berg et al. 2018), GraphRec (Fan et al. 2019), NGCF (Wang et al. 2019), and IGMC (Zhang and Chen 2020). Among them, GRALS is a graph regularized matrix completion method and can incorporate auxiliary information. NNMF extends the traditional NMF with multi-layer perceptions. NGCF and GraphRec leverage the user-network graph structures for matrix completions. GC-MC and sRGCNN are transductive node-level-GNN-based matrix completion algorithms. F-EAE uses exchangeable matrix layers to perform inductive matrix completion without using content information. Finally, IGMC uses one-hop subgraphs around user-item pairs to perform inductive matrix completion. We first conduct experiments on the two states in Yelp. We further conduct experiments on four additional data sets commonly used for evaluating the matrix completion task in recommender system design, to show the generalization of our method. We conclude this section by testing how various components of our method contribute to the prediction performance.

4.3.1. Performance evaluation on Yelp We summarize the performance of our method and the baselines on four metrics (i.e., RMSE, FCP, BPR, and Spearman correlation) in Figure 6. Among all methods, the performance on different metrics are mostly consistent, as shown by the rankings of all methods. This shows that a loss function based on MSE can be applied to personalized rankings in practical RS, as measured by BPR and FCP. We observe that our method outperforms all other methods on the two datasets (ON and PA).

As mentioned above, we split the train, test, and validation set according to the year of the ratings. Hence, the test set differs more from the validation and training set, comparing with random splitting. This sample split is more sensible as it is more in line with practical RS, where platforms predict future ratings. Due to this type of sample split and the more significant difference in the test and training set, machine learning methods are more prone to overfitting. Moreover, due to the dynamic nature of the dataset, new businesses and users will join the Yelp platform. Methods that do not use auxiliary information suffer from the cold-start problem and tend to perform worse on new users and businesses. We observed that the strong non-deep-learning benchmark SVD++ performs better than many deep learning benchmarks, which may be explained by potential overfitting that tends to be a more severe problem when the test set differs more. Among all the benchmarks, IGMC, GC-MC, GRALS, and F-EAE show stronger performances than others. Out of these, IGMC and F-EAE are inductive learning methods, which perform better in this setting with new business and users joining the platform. F-EAE does not perform as well as IGMC, which can be expected based on its inferior performance on MovieLens100K as mentioned earlier. GRALS and GC-MC both incorporate auxiliary information, which offers them an advantage, especially on Yelp, where there exists rich auxiliary information. Monti et al. (2017a) utilizes networks but not auxiliary information, which provides more predictive power in the Yelp case, as shown in the ablation study in Section 4.3.3. NNMF (Dziugaite and Roy 2015) is a straightforward extension on Non-negative Matrix Factorization using multi-layer perceptions. Both methods do not rely on either auxiliary or network information, which is a disadvantage compared to other methods. Both GraphRec (Fan et al. 2019) and NGCF (Wang et al. 2019) utilize network information by assuming that all neighbors provide equal predictive powers and do not rely on auxiliary information. In the Yelp example, where there exist high-dimensional covariates and new business/users in the test set, the capability to incorporate auxiliary information is especially important. In summary, the improved performance demonstrates our method's effectiveness, due to (1) the incorporation of auxiliary information about businesses and users, and (2) the introduction of the attention mechanism that can select the most relevant information in learning the latent embeddings and handling new businesses/users.

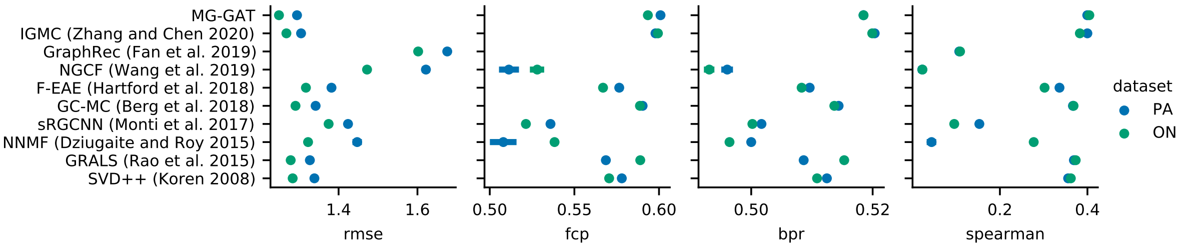


Figure 6 Performance evaluations on Yelp. A Lower RMSE, a higher FCP, a higher BPR, and a higher Spearman correlations correspond to better performance. Error bars correspond to 95% confidence interval.

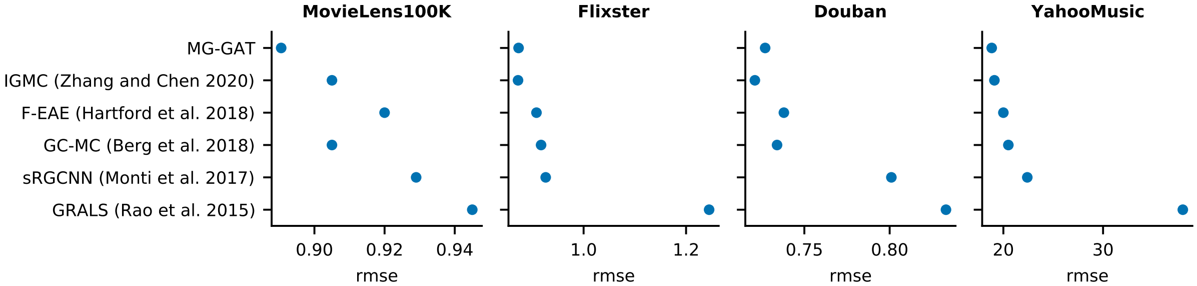


Figure 7 Performance evaluations on four standard data sets in recommendation tasks.

4.3.2. Performance evaluation on MovieLens, Flixster, Douban, and YahooMusic

To show the robustness of our method performance, we utilize four standard data sets used in the literature in recommendation tasks, including MovieLens[††], Douban[‡‡], YahooMusic and Flixster.

These four data sets are preprocessed and split by (Monti et al. 2017a) and are later adopted by many following-up machine learning papers on recommendations (Zhang and Chen 2020, Fan et al. 2019, van den Berg et al. 2017, Hartford et al. 2018). In these data sets, we only compare with stronger deep learning benchmarks in the previous section, i.e., GRALS, sRGCNN, GC-MC, F-EAE, and IGMC. The performances are shown in Figure 7. We show that our method performs better than all other benchmarks on MovieLens and YahooMusic. We perform on par with IGMC on Flixster and rank the second on Douban following IGMC best (Zhang and Chen 2020). Among the benchmarks, IGMC, F-EAE, and GC-MC perform better than the remaining two. They rank the top three among the six methods shown in some of the data sets. These analysis demonstrate the robustness and generalization of our performance.

4.3.3. Ablation study: the importance of different components of our method In this section, we test the importance of different components of our method to the predictive performance using eight variants of our method, as shown in Figure 8. Specifically, we test the contribution of GAT, quality of networks, graph regularization, and auxiliary information.

Since GAT distinguishes our framework from other methods, we test four variations on GAT. First, we test how heterogeneity of neighbor importance enabled by GAT contributes to the performance by replacing GAT with GCN. Theoretical comparisons of the two can be found in Section 3.5. We see that the performances with GCN are worse than GAT in both states. The difference between the two methods lies in whether the weights on network connections are heterogeneous or not. This result demonstrates the neighbor information graph's meaningfulness: the network connections do not contribute equally to the prediction task. The auxiliary information can be used to weigh the network connections.

Second, we test three variants without local aggregation by replacing the GAT layer on the users and (or) businesses with a dense layer. The drop in the performance in these variants demonstrates the effectiveness of MG-GAT in local information aggregation. We see that without any local aggregation, the performance of this variation is worse than GCN in PA and better in ON. This observation indicates that the network information in ON contains more noises than predictive information. Hence, equally aggregating edges on the observed network hurt the performance. Meanwhile, this result also demonstrates the effectiveness of GAT in removing noisy relational information. The performance deteriorates more in removing user GAT than business GAT in both states, indicating that the user GAT provides more predictive power.

Third, we study how our method responds to missing edges and noisy relational information. With noisy information, the performance is on par with missing edges in ON and better than that in PA. This result demonstrates the robustness of MG-GAT against noise.

Fourth, we remove the graph Laplacian term from our loss function and keep the GAT layers. The drop in the performance indicates the meaningfulness of the global smoothness assumption on the network, as observed in other studies (Cai et al. 2010, Leng et al. 2020a).

Last, we test the value of auxiliary information by dropping auxiliary information as a variation. We see that the learning performance is the worst among all variants in this case, which indicates the value of integrating heterogeneous information sources in predictions in Yelp.

The ablation study in this section confirms the value of various components (i.e., GAT, network information, graph Laplacian, and auxiliary information) in our framework.

5. Interpretation analysis and managerial insights

In addition to the improvement in prediction performance demonstrated in the previous section, an important advantage of our method is the interpretability and managerial insights enabled by the graph-based attention mechanism. To this end, we analyze feature relevance, neighbor importance graph, and the user and business embeddings.

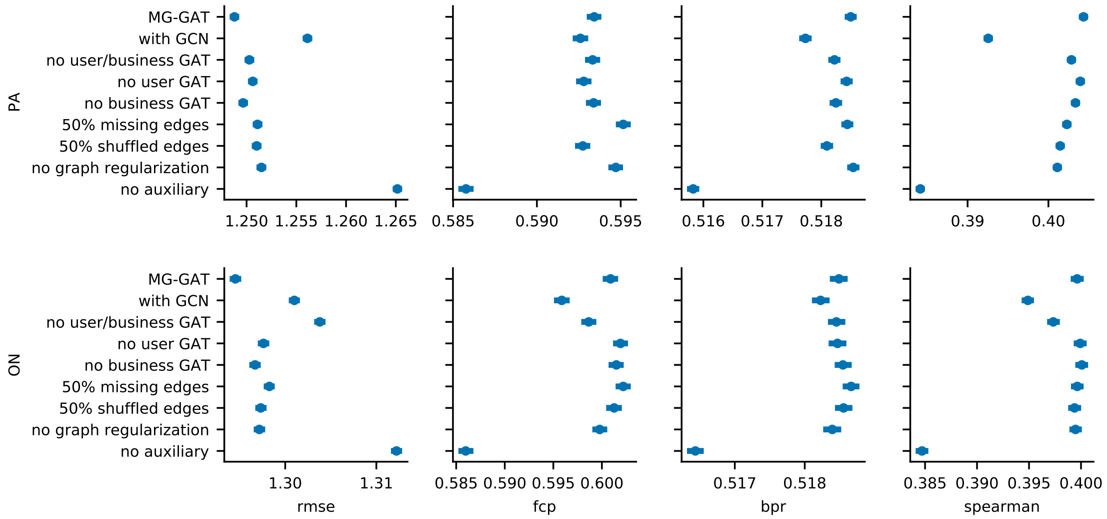
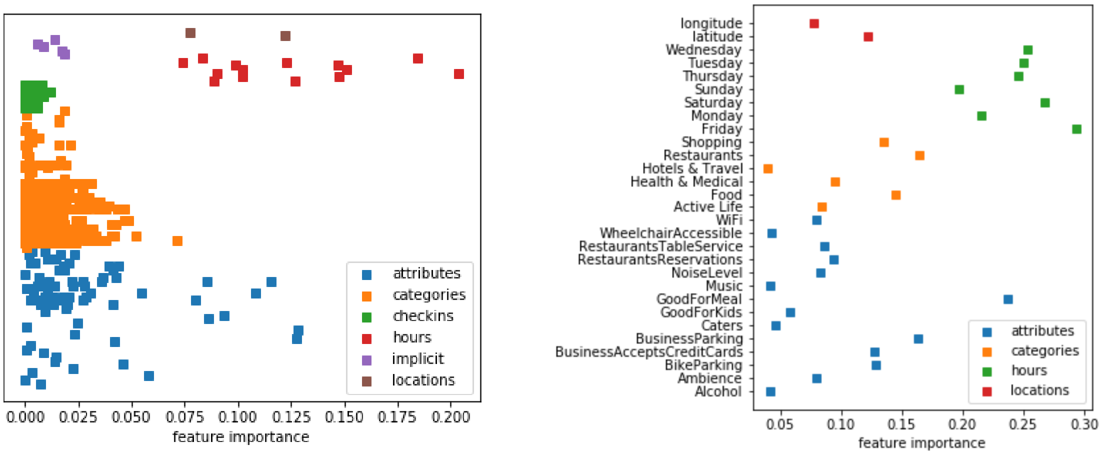


Figure 8 **Ablation study: performance of MG-GAT variants on Yelp. Error bars correspond to 95% confidence interval.**

5.1. Feature relevance to neighbor importance

Results in the previous section demonstrate the effectiveness of the learned neighbor importance in rating prediction. From now on, we focus our analysis on Ontario as an illustration and to avoid repeated analysis. Recall from Eq. (3) that these importance scores are associated with the business or user features of different weights. The Pearson's correlation coefficient between $\mathbf{a}_{b,\text{self}}$ and $\mathbf{a}_{b,\text{nb}}$ is 0.940 (p-value < 0.001) and that between $\mathbf{a}_{b,\text{self}}$ and $\mathbf{a}_{b,\text{nb}}$ is 0.998 (p-value < 0.001). Feature weights, $\mathbf{a}_{b,\text{nb}}$ and $\mathbf{a}_{u,\text{nb}}$, predict neighbors' feature relevance to the focal node. Thus, our model yields feature selection according to their predictive powers to the underlying business and user relationships in the latent space, which then affects ratings. This property makes the black-box algorithm more interpretable, which may help businesses and the Yelp platform design targeting and operation strategies.

Figure 9a shows the feature weights of all business auxiliary information, and Figure 9b displays the top 40 business attributes. Across all 1221 features, it is interesting to see how features in each category show similar patterns along the x-axis in Figure 9a. For example, operation hours and geographic locations have substantially larger weights than others. This suggests that businesses that are similar in terms of geographical location and the operation hours are more relevant to rating prediction, reflecting the nature of Yelp as a local recommendation platform. In comparison, check-ins and implicit features are less relevant. Business attributes are more heterogeneously distributed than others. Among them, *bike parking, business accepts credit cards, restaurant reservations, restaurants table service, good for meal, ambience, good for kids*, and *business parking* are the most highly-ranked. These insights are especially helpful for the Yelp platform to guide businesses,



(a) Feature importance

(b) Top 40 selected features

Figure 9 Business feature importance contributing to neighbor importance. Larger weights along the x-axis correspond to more relevant features in computing the neighbor importance.

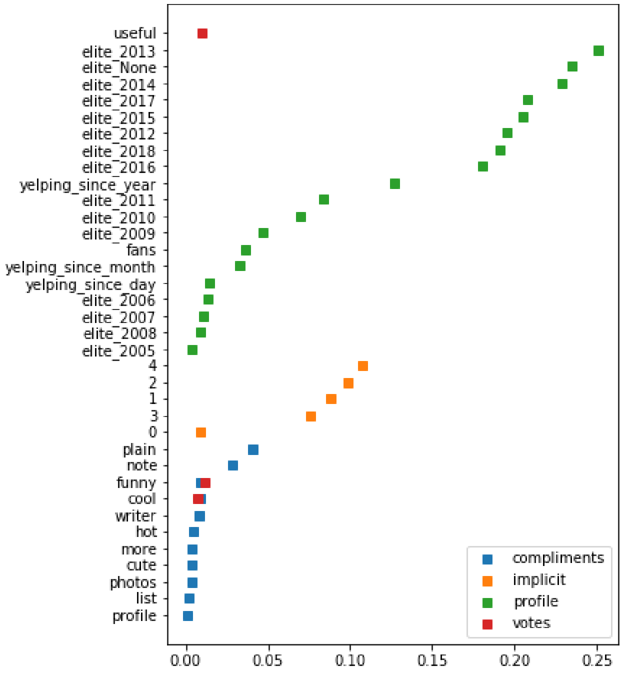


Figure 10 User feature importance contributing to neighbor importance. Larger weights along the x-axis correspond to more relevant features in computing the neighbor importance.

especially new and local ones, to increase consumer exposure. We demonstrate the application of feature weights in the first experiment in Section 6.1.

As a concrete example of feature relevance, we consider one particular store in the business chain Second Cup, and analyze the learned important neighboring businesses to predict this store's rating. In Figure 11, we present the three most (left) and least (right) important neighbors, along

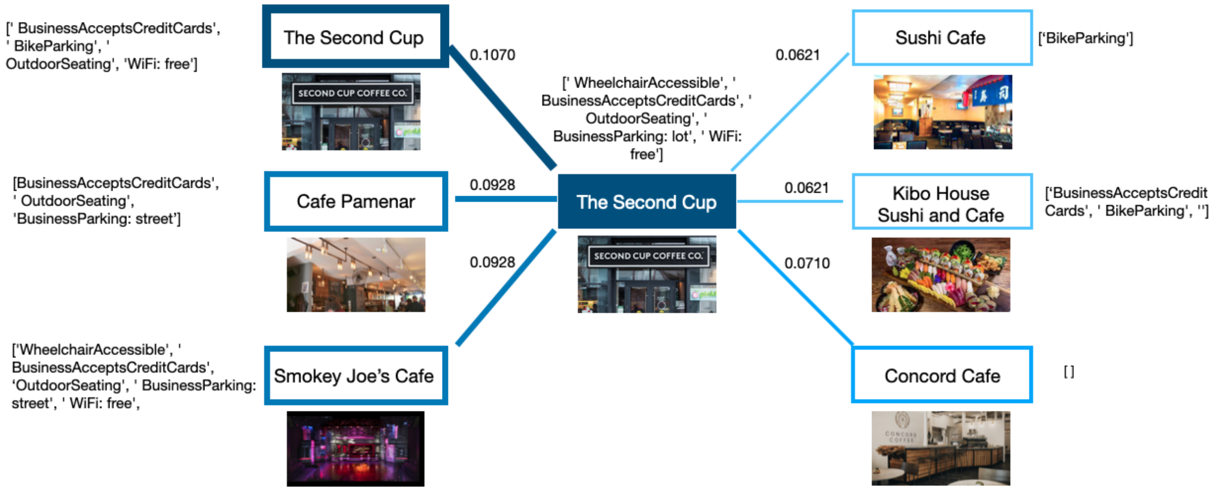

[' BusinessAcceptsCreditCards', ' BikeParking', ' OutdoorSeating', 'WiFi: free']
The Second Cup
SECOND CUP COFFEE CO.
0.1070
[BusinessAcceptsCreditCards', ' OutdoorSeating', 'BusinessParking: street']
Cafe Pamenar
0.0928
0.0928
['WheelchairAccessible', ' BusinessAcceptsCreditCards', 'OutdoorSeating', ' BusinessParking: street', ' WiFi: free',
Smokey Joe's Cafe
[' WheelchairAccessible', ' BusinessAcceptsCreditCards', ' OutdoorSeating', ' BusinessParking: lot', ' WiFi: free']
The Second Cup
SECOND CUP COFFEE CO.
0.0621
Sushi Cafe
['BikeParking']
0.0621
Kibo House Sushi and Cafe
['BusinessAcceptsCreditCards', ' BikeParking', '']
0.0710
Concord Cafe
[]

Figure 11 An Illustration of the top and bottom attention weights using the business, The Second Cup.

with the features of significant weights. We see that more important neighbors share more common features (e.g., *business accepts credit cards*) with the focal business than the less important ones. This demonstrates how the selected neighboring businesses contribute to predicting the focal businesses' ratings in the proposed recommendation framework. Understanding this process helps Yelp interpret the behavior of the black-block algorithm, and may lead to business insights by itself. For instance, this can be especially helpful for the Yelp Ads functionality, one of the main revenue sources for Yelp, to make better recommendations and automatically generate business insights.

We also perform a similar analysis on user feature weights, as shown in Figure 10. We see that the elite status contributes more to high neighbor importance, which is expected. Interestingly, we observe that elite status is less important before 2012, which is when Yelp became a public company. Additionally, compliments users receive on their comments (e.g., funny and cool) are not as important as elite status because compliments are already taken into account by Yelp when issuing the elite status. Unlike businesses, implicit features (colored in orange) are more relevant to user neighbor importance. This may be because business auxiliary information is richer and explains more variations in implicit features. While the user auxiliary information contains fewer features, hence the implicit features contribute more.

5.2. Analysis of the neighbor importance

An essential ingredient in our framework is the relationships (neighbor importance) between business pairs and user pairs, which capture the latent predictive relationships. This section presents more detailed analyses of the neighbor importance, i.e., the strength of neighbor importance and the neighbor importance graph.

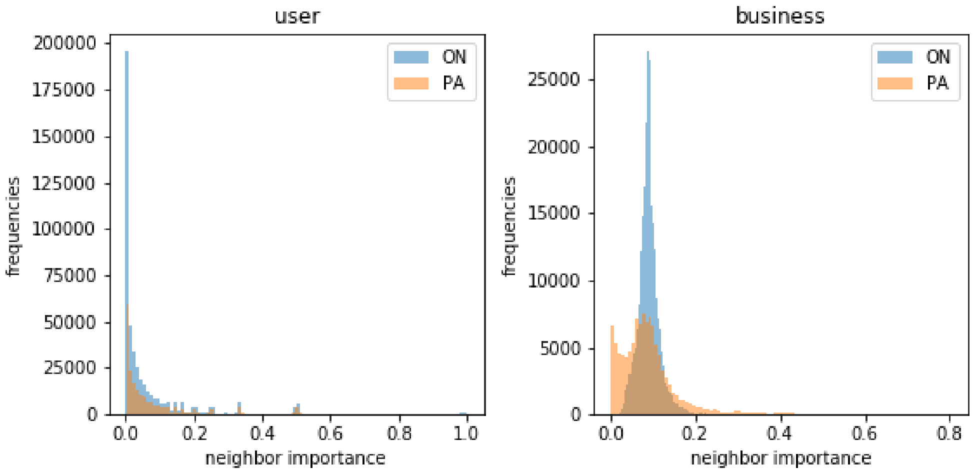


Figure 12 **Distribution of neighbor importance for users (left) and businesses (right).**

5.2.1. Distribution of the strength of neighbor importance We first analyze the distribution of neighbor importance for both users and businesses, as shown in Figure 12. The distributions for users and businesses present different patterns. User neighbor importance presents a truncated power-law distribution, with a dense mass around 0; for businesses, the distribution is less skewed with the mass centered around 0.1. Both figures suggest that equally-weighted neighbors do not provide strong predictive power, since there is strong heterogeneity in the learned neighbor importance. This implies that the attention mechanism can identify relevant neighbors for each user or business and assign weights accordingly, which is different from frameworks without attention, e.g., the graph convolutional network (Kipf and Welling 2017), where the weights on all network links are predefined and treated equally by construction.

5.2.2. Important nodes in the neighbor importance graph Contribution of a specific business (or user) to other businesses (or users), in terms of the neighbor importance, demonstrates its relative importance to the predictive performance of other businesses (users). A strong contribution means that the business presents large predictive power to other businesses. Specifically, we can quantify this by investigating the contribution of one business (user) to the ratings of all other businesses (users). This can be thought of as computing the out-degree centrality on the neighbor importance graph. Recall that the weight on the link from business i to business k is $\alpha^b_{i\to k}$, which captures contribution from i in updating k's embedding. The nodal importance of i to other businesses, or its out-degree centrality, can then be defined mathematically as,

$$\text{centrality}_i^{\text{out-degree}} = \sum_{k\in N_i^b} \alpha^b_{i\to k}. \tag{11}$$

The out-degree centrality can be similarly defined for users.

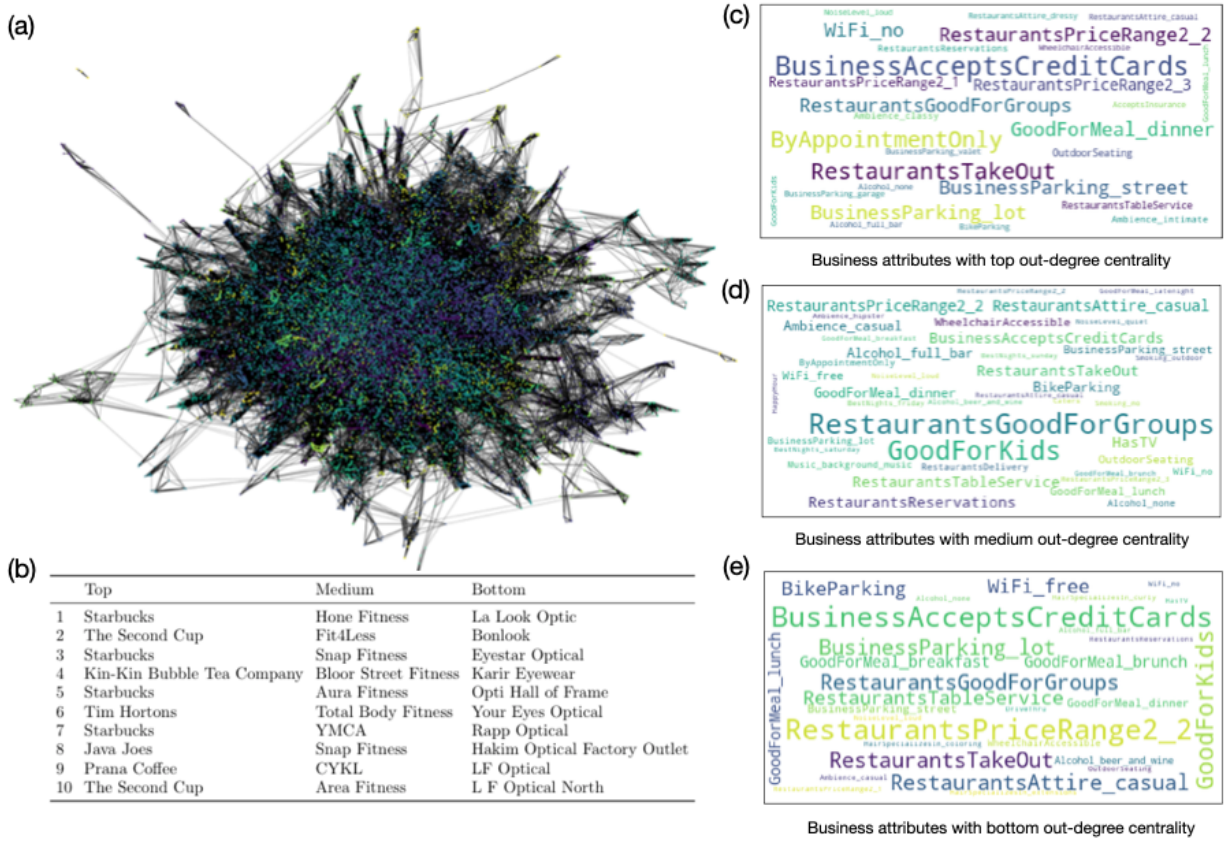
In Figure 13a, we display the neighbor importance graph for businesses and compute the out-degree centrality of each business. We present the top-, medium- and bottom-ranked businesses in terms of out-degree centrality (Figure 13b). We see that the top-ranked businesses are mostly coffee or tea chain stores, while the bottom-ranked ones are mostly optical stores. The top-ranked businesses may have reflected businesses that provide more predictive power to other businesses on the Yelp platform. We further analyze the attributes of these businesses in Figure 13c-e. We see that *business accepts credit cards, restaurants takeout* are more common attributes for the top- businesses in the business neighbor importance graph. At the same time, *restaurants good for groups, good for kids* features more prominently among medium-ranked businesses. The bottom-ranked businesses share similar features with top- and medium-ranked businesses, while not as prominent.

Similarly, we study the characteristics of the top-, medium- and bottom-ranked users by out-degree centrality in the neighbor importance graph. Since there is less auxiliary information for users, we propose to analyze and compare business features that predict high (e.g., 5) versus low (e.g., 1) ratings for users with different out-degree centrality. This analysis enables us to understand the decision-making process of the users. To this end, we use the Cohen's d to measure the effective size in comparing the means of a specific attribute between high- and low-rating businesses. The Cohen's d (C_d) is defined as:

$$C_d = \frac{m_{h,q} - m_{l,q}}{\mathrm{sd}_{\mathrm{pooled}}}, \tag{12}$$

where $m_{h,q}$ and $m_{l,q}$ are the means for attribute q in high- and low-rating businesses, respectively. $\mathrm{sd}_{\mathrm{pooled}}$ is the estimate of the pooled standard deviation of the two groups, which can be calculated as $\mathrm{sd}_{\mathrm{pooled}} = \sqrt{\frac{v_{h,q}+v_{l,q}}{n_h+n_l-2}}$, where $v_{h,q}$ and $v_{l,q}$ are the variance of attribute q in the high- and low-rating businesses, and n_h and n_l are the number of samples in the two groups, respectively.

We present the top-15 features by Cohen's d in Figure 14b-d, for the top-, medium- and bottom-ranked users by out-degree centrality in the user graph. We color the attributes according to their types. We see that two parking facilities, *Business parking: street* and *bike parking*, are distinguishable in all three groups. There also exist features unique to different groups. For top-ranked users (i.e., those that are more influential to others), we see that *bike parking*, *wheelchair accessible*, and *medium restaurants price range* are the most distinguishable features for high- and low-rating businesses. *Good for meals* takes up four out of fifteen features, while this feature is not as distinguishable for the bottom users. For medium-ranked users, *caters, best nights: Saturday, good for kids* are the top three features. *Best for nights* and *ambience* each take up two features out of the top 15 features for medium users, while it does not show up for top and bottom features. In particular, *street parking* is the most distinguishable feature whose effective size is by some distance



(b)

	Top	Medium	Bottom
1	Starbucks	Hone Fitness	La Look Optic
2	The Second Cup	Fit4Less	Bonlook
3	Starbucks	Snap Fitness	Eyestar Optical
4	Kin-Kin Bubble Tea Company	Bloor Street Fitness	Karir Eyewear
5	Starbucks	Aura Fitness	Opti Hall of Frame
6	Tim Hortons	Total Body Fitness	Your Eyes Optical
7	Starbucks	YMCA	Rapp Optical
8	Java Joes	Snap Fitness	Hakim Optical Factory Outlet
9	Prana Coffee	CYKL	LF Optical
10	The Second Cup	Area Fitness	L F Optical North

Business attributes with top out-degree centrality

Business attributes with medium out-degree centrality

Business attributes with bottom out-degree centrality

Figure 13 Business neighbor importance graph. (a) Visualization of business neighbor importance network. (b) Names of the top-, medium-, and bottom-business according to the out-degree centrality. (c)–(e) Word cloud of the frequencies of business attributes for top-, medium-, and bottom-business according out-degree centrality.

the largest among all three groups. For bottom-ranked users, *good for kids, restaurants takeout*, and *business parking: street* are the most distinguishing features. The price range is distinguishable among high- and low-rating businesses both for the top and the bottom users, while the range is lower for bottom users.

These results enable businesses to design different strategies targeting different users on the Yelp platform. Moreover, Yelp can recommend businesses with the particular attributes that users care about. For instance, whether a business offers street parking or not is a distinguishable feature to users' ratings of the business. The platform and businesses might benefit more from meeting the requirement or preferences of important users (those with high out-degree centrality) in the user graph, given that they exert a strong predictive influence on ratings of other users (Kumar and Hosanagar 2019). Though our analysis does not offer a causal relationship for designing policy, the feature ranking provides a set of hypotheses for businesses and Yelp to test upon.

5.2.3. Community detection in neighbor importance graphs Business and user segmentation are important to streamline the strategies to a specific target market that is more profitable

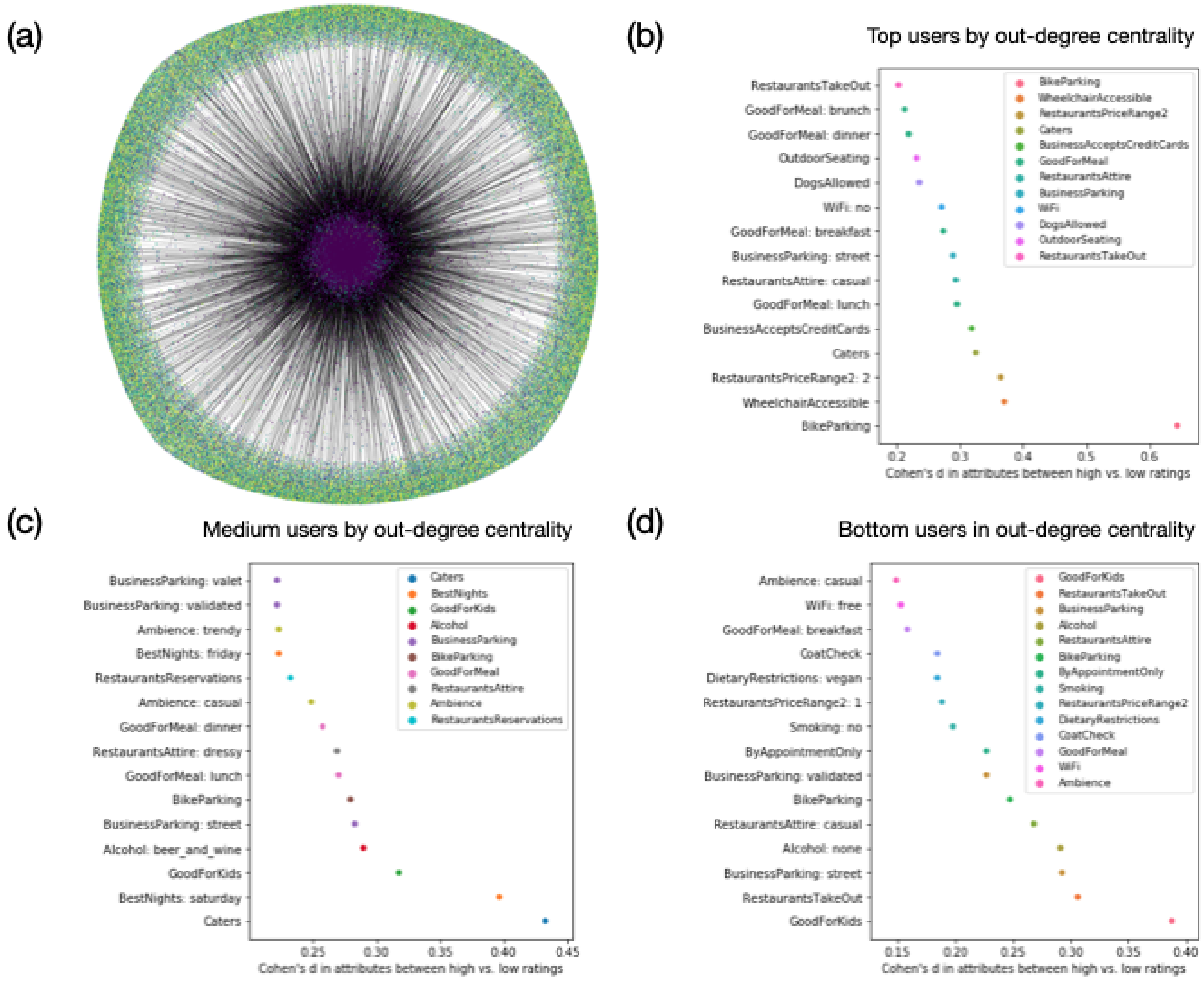
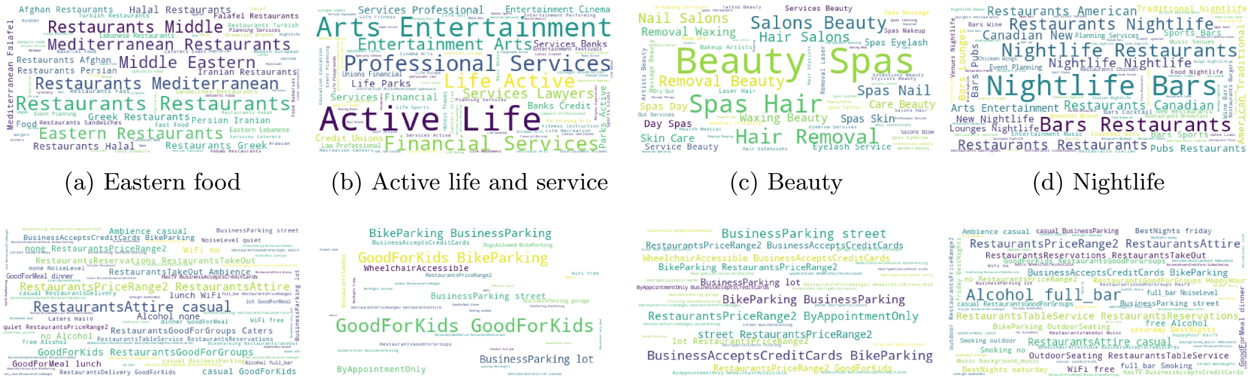


Figure 14 Visualization of user attention network.

with lower operating costs. For example, we can design and automate marketing strategies for a specific segmentation of businesses that share certain understandable characteristics. To get a perception of the underlying attention network and demystify information contained in neighbor importance score, we first apply the Louvain method to the neighbor importance graphs to detect communities of businesses and users (Blondel et al. 2008, Lu et al. 2015). To this end, We visualize the business communities in Figure 13a and user communities in Figure 14a, where colors index the communities. Proximity in the graph space (or being in the same community) indicates a higher likelihood that two business (users) share similar characteristics (preferences). We see that there are many niche communities in the business graph, while the user graph presents a core-periphery structure (Rombach et al. 2014). Indeed, out of 135,173 users, 64.0% are on the periphery. This is in line with the sparse and power-law structure of friendship networks. It is also worth noting that stronger heterogeneity exists in the community structures for users than businesses: we obtain 101 business communities and 88,432 user communities.



(a) Eastern food (b) Active life and service (c) Beauty (d) Nightlife

(e) Casual attire (f) Good for kids (g) Parking (h) Alcohol

Figure 15 Analysis of business categories (a–d) and attributes (e–h) of different business communities in Ontario.

As a case study, we perform analysis of the four largest business communities, where we study the auxiliary business information of each community. Specifically, we analyze the features of businesses in each community and visualize the frequency of different business categories and business attributes separately by using the word cloud in Figure 15. The community size decreases from the first to the last column. We see a clear separation of business categories and differences in the business attributes. The first community is strongly associated with *Mediterranean and Middle Eastern restaurants*, and the *casual attire* dress code, while the second contains businesses that are related to *active life* and *services*, and *good for kids*. The third and fourth communities are associated with *beauty*, *spas*, *parking* facilities and *nightlife, alcohol*, respectively. These results provide insights into

5.3. Business and user embedding

In addition to the feature relevance and neighbor importance graph, we are interested in understanding the learned business and user embedding. Intuitively, these are the latent representations that capture the underlying characteristics of businesses and users.

We are first interested in understanding the information contained in the learned latent business representations $\mathbf{B}$, concerning business attributes and categories. To get a perception of what the embeddings represent, we compress the embedding vectors into a two-dimensional space using t-SNE (Maaten and Hinton 2008) and visualize them in Figure 16a. We perform the k-means algorithm to group the businesses into k clusters where the optimal number of clusters, k, is chosen using the elbow method. Notice that the business clusters obtained here are slightly different from those discussed in the previous section. While those communities are obtained based on how a business is important to another business (via the neighbor importance score), their ratings are only implicitly related. In this sense, the business clusters from the embedding vectors, which

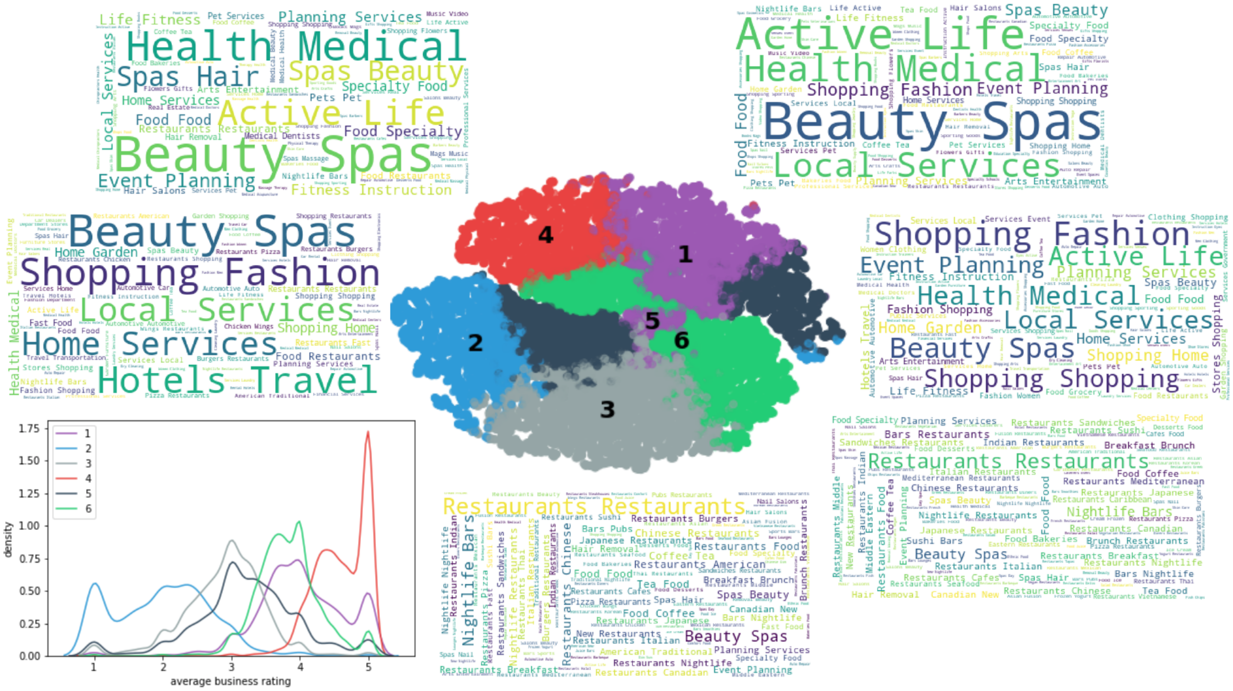


Figure 16 Business embedding and word cloud of clusters

are obtained via factorization of the rating matrix, are more directly related to business ratings. Indeed, when evaluating the business rating distributions within each cluster (lower-left panel of Figure 16), we observe a clear separation between different clusters in terms of ratings. For example, clusters four is mostly associated with higher ratings (4 or 5), while cluster two has relatively low ratings (1 or 2). We further study the categories of businesses for each cluster in the word cloud in Figure 16. We see that cluster four is strongly associated with *beauty* and *spas*, while cluster six contains many restaurants. *Hotels, travels*, and *home services* are only common in cluster two, which suggests that most ratings for these business categories are low. We also see that businesses related to *active life, health*, and *medical* are more common in clusters one, four, and five, which together have a more uniform spread of rating distributions.

We further study the user embeddings in Figure 17. Similar to the business case, we use t-SNE to project the embedding matrix $\mathbf{U}$ onto a two-dimensional space and visualize the clusters obtained via k-means in Figure 17a. The percentages of users in clusters one to four are 81.7%, 3.3%, 10.8%, and 4.1%, respectively. In terms of ratings, users in cluster one tend to rate businesses with more extreme scores (1 or 5). Users in other clusters all tend to provide relatively higher ratings, with cluster two most strongly associated with rating 4. To gain more insights into the users' rating behavior, we use the Cohen's d to examine what attributes (Figure 17c) and categories (Figure 17d) explain the differences between a business with a rating of five versus another with a rating of one. We present the top 15 features for both cases. We see that *bike parking* and *business parking: street*

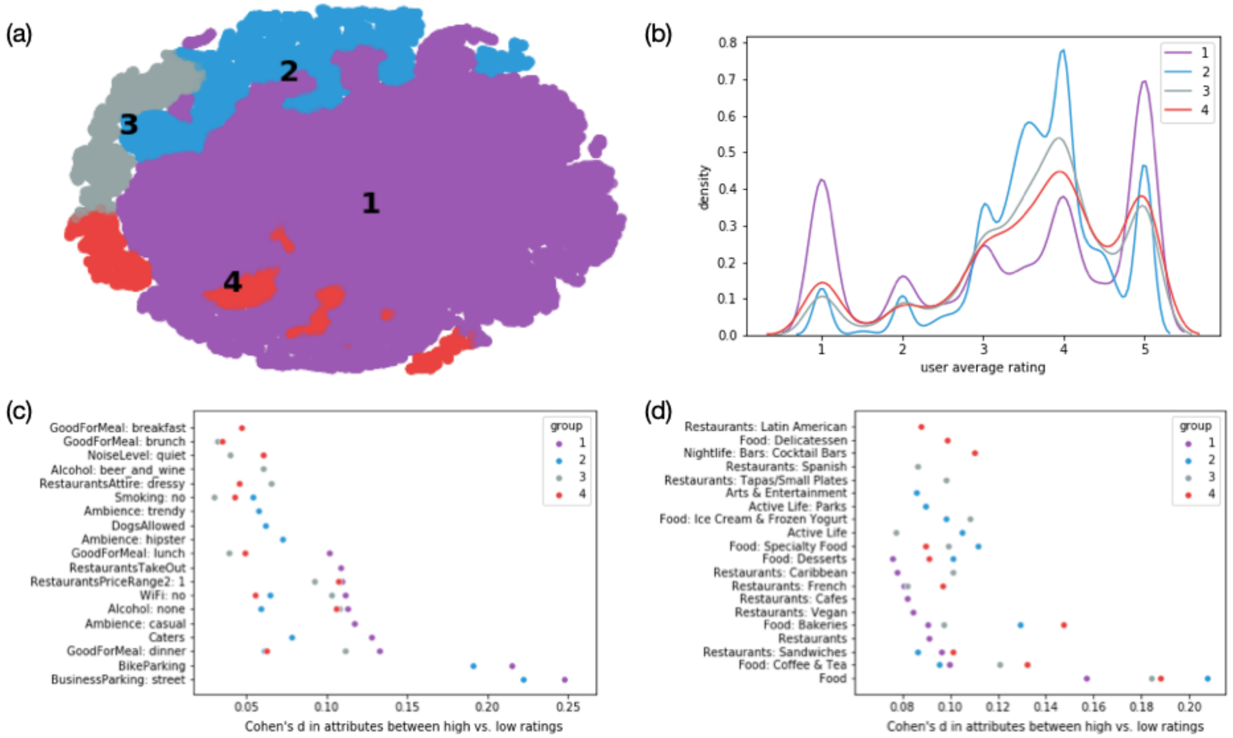


Figure 17 **User embedding analysis**

are important features that differentiate a high and low rating for users in clusters one and two. Recall that these two features are also prominent in feature importance and for users with top-, medium-, and bottom-centrality users on the neighbor importance graph. *Good for meal: dinner, WiFi: no* and *Alcohol: none* are important features for users in all four clusters. Users in clusters three and four both find features such as *good for brunch*, *quite*, and *dressy attire* important, while users in cluster two care a distinct set of features, e.g., *trendy* or *hipster* ambience, and whether dogs are allowed or not.

In terms of categories, a high Cohen's d in one business category indicates a large difference in the ratings between that particular category and other businesses. We see that *food, coffee & tea*, and *bakeries* are shared in all clusters. We also notice that there are fewer categories that are shared across clusters than attributes. For example, *vegan* and *cafes* are unique for users in cluster one, *arts & entertainments* and *parks* are unique for users in cluster two, *Spanish and tapas/small palates* are specific to cluster three, and *Latin American, delicatessen*, and *cocktail bars* are unique for users in cluster four. These analyses may help better interpret predictions made by the proposed deep learning framework, by extracting features that are predictive of high- versus low-rating businesses.

In summary, the results in this section demonstrate that the proposed framework can not only perform rating predictions well but also yield several qualitative/quantitative insights regarding user preferences and business characteristics, via analyses in both the neighbor importance graph space and the low-dimensional embedding space.

6. Managerial applications

This section demonstrates how our method can be used in different managerial applications, especially the neighbor importance graph. We first study how the feature weights and neighbor importance graph can be used in information acquisition by the Yelp platform. Next, we study a personalized search setting to learn how the neighbor importance graph can be used with different information sets (user preferences and search criteria). We then demonstrate how our method can be useful in designing targeted advertising strategies. We close this section with an illustration of how feature importance can be used for interpretable recommendations in a cold-start setting.

6.1. Information acquisition

Information required in predictive modeling tasks is not immediately available and is usually acquired at a cost (Saar-Tsechansky et al. 2009). With limited budget and time, algorithmic decision-making may help balance a trade-off between the cost of information acquisition and the quality of the predictions. We now demonstrate how feature relevance and neighbor importance graph can determine what new information would be the most useful to acquire.

We first study the value of feature relevance in platform information acquisition. When new businesses join the Yelp platform, they need to complete a survey with many features; some of the features provide more predictive power than others, as shown in Section 5.1. Feature weights allow us to determine which features are critical to collect when businesses join the platform. Specifically, we evaluate our method's predictive performance, assuming that only a limited number of auxiliary information are available. We experiment with two scenarios: one with the top fifty features according to feature importance; the other with a random fifty features. As shown in the upper panel of Figure 18, the method with the top features according to feature importance produces a better performance than randomly selected features. This result demonstrates how feature importance in Def. 4 can be used to understand what information would be most useful to acquire for new businesses under budget or time constraint.

In the second experiment, we demonstrate how the neighbor importance graph can be used to acquire rating information on businesses to optimize performance. In this experiment, we acquire 1,000 new observations on business ratings: one experiment uses the observations on the centrally-positioned businesses on the neighbor importance graph, and another experiment uses observations on random features. As shown in the lower panel of Figure 18, additional information collected according to the neighbor information graph can improve the performance to a greater extent than random observations. This result demonstrates how neighbor importance graphs can be used to acquire new rating observations to improve rating predictions.

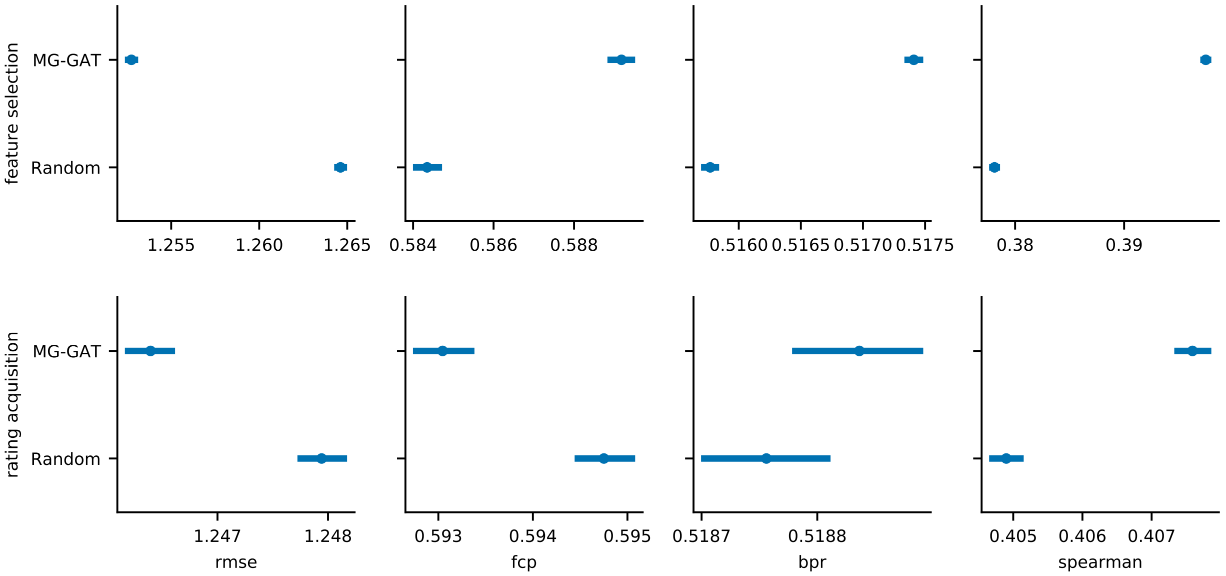


Figure 18 MG-GAT for information acquisition. Error bars correspond to 95% confidence interval.

6.2. Personalized search

We next demonstrate how to use our model to make personalized recommendations for a specific user based on user queries involving different business attributes (Ghose et al. 2014). When a user actively looks for businesses with a particular attribute, either by typing the features or by recent browsing (searching) histories with similar features, we can leverage this extra information to obtain a personalized ranking of businesses that not only have specific characteristics but also are more likely to be highly-rated by the focal user based on the preferences revealed by the users rating histories. For example, we consider a random user, e.g., user id 759, set the searching criterion for restaurants with live music. We can use the neighbor importance to identify a ranked list of businesses that are in close proximity to the businesses that 1) meet this criterion and 2) have been highly rated by this individual. The ranked businesses conditioned by three different search criteria are shown in Table 4. We see that the platform displays different results for the users according to different keywords.

Table 5 demonstrates the relevant set of personalized ranking for three users when they searched for the same criterion, i.e., dressy attire. The rankings differ across users because it is driven by the variations in their preferences. We can see that user id 59022, 134652, and 519 have different preferences: they like restaurants, clothes, and bakery, respectively. This experiment demonstrates that our method can provide rankings based on the limited information on keyword searches.

6.3. Platform targeting

Like other crowd-sourced review platforms, Yelp offers targeted advertising as a profitable way to help businesses attract consumers. Our model provides a way to make targeting more granularized.

Table 4 Top recommended businesses for user 759 based on attribute searches

	Nightlife: Bars: Wine Bars	RestaurantsAttire: dressy	Music: live
1	Hawaii Bar	The Boot Social Pizzeria	Carlaw Bistro
2	Eton House Tavern	Teriyaki Experience	Saint James Hotel
3	Tap Works Pub	Teriyaki	Thompson Toronto
4	Four Barrel Holly's	Banzai Sushi	Stage West All Suite Hotel & Theatre Restaurant
5	Yonge Street Warehouse	DonDon Izakaya	Foxxes Den
6	Busters by the Bluffs	Zakkushi	Radisson Admiral Hotel Toronto-Harbourfront
7	Olde Village Free House	Naomi	The Riverside
8	Filly & Co	Teara Lab	Windsor Arms Hotel Spa
9	Tudor Arms Pub	Teriyaki Experience	Best Western Voyageur Place Hotel
10	The Filly & Firkin	Chou Izakaya	Stillwater Spa

Table 5 Top ranked businesses based on the search of keyword: Dressy

	User 59022	**User 134652**	**User 519**
1	The Boot Social Pizzeria	Dutil Denim	Bake Code
2	Teriyaki Experience	Jeansfirst	T & T Supermarket
3	Teriyaki	Levi's Outlet	Longo's
4	Banzai Sushi	Andrew's Formal Rentals	Michidean Limited
5	DonDon Izakaya	Trend Custom Tailors	Loblaws
6	Zakkushi	Moores Clothing for Men	The Big Cannoli
7	Naomi	Suitsupply	Simply Yummy Bakery
8	Teara Lab	Moores Clothing for Men	COBS Bread
9	Teriyaki Experience	Made2Measure Clothing	Cobs Bread
10	Chou Izakaya	Jacflash	Reesor's Farm Market

Yelp currently advertises businesses to consumers nearby, based on the geographic location. However, exposure does not mean high ratings (Leng et al. 2020b). For example, if a restaurant without free WiFi access was targeted at consumers who highly value this in their rating, this would not benefit the business. This would especially affect local businesses with only a few ratings, who also need more attention and high ratings for success. Even though this can be done through business-based collaborative filtering and predict ratings to new businesses, the traditional method suffers from the inability to explain why a particular business is recommended and a high computation cost for each query. Our model, especially the neighbor importance, enables us to target consumers who might rate the businesses high.

With MG-GAT, we identify businesses similar to the focal advertising business on the neighbor importance graph and then query consumers who have rated that business high. Therefore, this process is efficient, scalable, and without any need for re-training the model. Let us think about a business, Sansotei Ramen, which plans to do advertisements. Table 6 illustrates the example of the top five businesses that are most similar to Sansotei Ramen, based on the neighbor importance. It is interesting to see that the businesses that are deemed similar to Sansotei Ramen are all noodle stores and share similar environmental characteristics, such as *good for kids, business accepts credit cards, bike parking, no WiFi, restaurants attire: casual.* The platform can then recommend Sansotei

Table 6 Five businesses most similar to the given business: Sansotei Ramen

Business name	Rating: 4	Rating: 5
Red King Kong	[110841]	[79452]
1915 Lan Zhou Ramen	[66236 90398 ... 120304 107567]	[11925 60395 128647 77690]
Magic Noodle	[21848 54070 ... 122205 97285]	[70561 97446]
KINTON RAMEN	[63023 74660 ... 57633 13312]	[33375 35943 ... 65577 81432]
Homemade Ramen	[91918 91097 ... 109270 65263]	[48398 127574 ... 7644 85916]

Table 7 Cold-start recommendations: similar businesses to Assembly Chef's Hall

	Feature weights	Cosine similarity
1	Nakayoshi Izakaya	Bloomer's
2	Bombay Palace	Page One
3	Hello 123	Si Lom Thai Bistro
4	Spring Rolls	Caffe Di Portici
5	Momofuku Noodle Bar	Cherry Street Bar-B-Que

Ramen to users who have high ratings for those similar businesses. The ability to identify businesses with the closest proximity in the latent space while simultaneously accounting for user preferences is highly beneficial in the day-to-day operation of a recommendation system.

6.4. Cold-start problem and interpretable recommendations

New businesses join the Yelp platform on a continual basis. Ratings and recommendations on those businesses are often not easy due to lack of reliable information. This is the typical cold-start problem faced by many existing recommender algorithms. As our method integrates auxiliary information on businesses, a natural and straightforward way is to compute a similarity measure using cosine similarity or Euclidean distance based on auxiliary information, and infer the embedding of the new business based on other similar businesses. The problem with this method is the ad-hoc similarity measure chosen. Regardless of the method, it is difficult to understand the contribution and importance of different attributes. Feature relevance (as shown in Section 5.1) learned from the graph attention network provides a more principled way to compute the neighbor importance score. Based on this information, we will be able to add the new business into the existing neighbor importance graph to perform other applications. As an example, we pick a new business, Assembly Chef's Hall. We use the learned feature weights to compute neighbor importance on all other businesses and compare with the ones using cosine similarity in Table 7. Assembly Chef's Hall is a fancy food court with creative culinary. The recommendations made by our algorithms are more reasonable. We see that "Nakayoshi Izakaya" and "Momofuku Noodle Bar" have a modern design and creative culinary. The other three businesses are cost-effective restaurants, in a similar price range as the food court. A more accurate list of similar businesses helps the platform to recommend the new business to more relevant users (e.g., those that rate similar businesses high).

In summary, our model is useful in supporting several management applications in a variety of managerial contexts.

7. Conclusion

In this paper, we propose a novel geometric deep learning framework for interpretable personalized recommendations by predicting users' preferences on businesses that they have not yet rated. Our framework possesses the following advantages over existing RS. First, it can handle multiple sources of heterogeneous information, including spatial and temporal information, relational (network) information, and other types of metadata on users or businesses. Second, the proposed method effectively filters information depending on its relevance and assigns larger weights on information that is more predictive of the user preference. This unique feature is enabled by the graph attention network. This, therefore, provides a general way of selectively aggregating the most relevant information for the task at hand, and ensures interpretability simultaneously. Testing our method on the rich and high-dimensional Yelp data set, we demonstrate that both the direct incorporation of network structure and selective aggregation of information from relevant neighbors in the network is essential to the learning performance in a prediction task. Our study has clear, practical implications for both the platforms and the business owners. First, our results demonstrate the value of auxiliary information (location, operations hours, business attributes, business categories, user profiles) and relational information in recommendation tasks in practice. Second, we cast doubt on the common assumption in the management literature that observed social network describes the actual relationship, due to measurement errors (Newman 2018) and heterogeneity of social relationships (Granovetter 1977, Biddle 1986). The heterogeneity of the neighbor importance shows that not all "observed" neighbor of a given user (business) contributes equally to understanding that user's preferences (business characteristics). Therefore, it is essential to filter out the noise and only focus on the relevant network connections. MG-GAT can be regarded as an automatic pipeline to achieve this objective. Third, it is important for companies to interpret the learned patterns on user and business representations so that business decisions can be combined with domain knowledge. The feature importance, neighbor importance graphs, and the embeddings enable such an interpretation (as demonstrated in Section 5). These outputs from our method enable a wide range of applications. Platforms can utilize important features (e.g., *business parking*) to design effective targeting strategies for new businesses (as shown in Section 6.4). Moreover, both the feature importance and the neighbor importance graph can help the platform design information acquisition strategy (as shown in Section 6.1). For businesses, we show that different attributes are important for different user segmentation to obtain higher ratings (as demonstrated in Section 5.3). They can utilize this information to target consumers and design advertising strategies accordingly.

Our study points out several future directions. First, the objective of this study is to make recommendations that best predict users' preferences. However, some businesses may be further interested in maximizing revenue based on inferred preferences. The maximization of business revenue can thus be incorporated into the objective function of the learning framework. Second, the information our framework handles is static. One interesting direction is developing an adaptive and online learning framework that can handle and incorporate temporal data and dynamic relational (network) information. Lastly, our paper's prediction framework only reveals associations between business characteristics and ratings and does not claim causality. Further studies are required through observational causal inference studies or random controlled experiments to design strategies to improve business ratings.

References

Agarwal A, Hosanagar K, Smith MD (2011) Location, location, location: An analysis of profitability of position in online advertising markets. *Journal of marketing research* 48(6):1057–1073.

Ansari A, Li Y, Zhang JZ (2018) Probabilistic topic model for hybrid recommender systems: A stochastic variational bayesian approach. *Marketing Science* 37(6):987–1008.

Aswani A, Shen ZJ, Siddiq A (2018) Inverse optimization with noisy data. *Operations Research* 66(3):870–892.

Bao J, Zheng Y, Wilkie D, Mokbel M (2015) Recommendations in location-based social networks: A survey. *GeoInformatica* 19(3):525–565.

Battaglia P, et al. (2018) Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261.*

Berg Rvd, Kipf TN, Welling M (2018) Graph convolutional matrix completion. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* .

Bergstra J, Yamins D, Cox D (2013) Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *International conference on machine learning*, 115–123.

Biddle BJ (1986) Recent developments in role theory. *Annual review of sociology* 12(1):67–92.

Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* (10):P10008 (12pp).

Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. *Knowledge-based systems* 46:109–132.

Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P (2017) Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine* 34(4):18–42.

Bruna J, Zaremba W, Szlam A, LeCun Y (2014) Spectral networks and deep locally connected networks on graphs. *International Conference on Learning Representations (ICLR).*

Cai D, He X, Han J, Huang TS (2010) Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence* 33(8):1548–1560.

Cai H, Zheng VW, Chang KCC (2018) A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering* 30(9):1616–1637.

Chen Y, Yao S (2017) Sequential search with refinement: Model and application with click-stream data. *Management Science* 63(12):4345–4365.

Cheng C, Yang H, King I, Lyu MR (2012) Fused matrix factorization with geographical and social influence in location-based social networks.

Cheng X, Zhang J, Yan L (2020) Understanding the impact of individual users rating characteristics on the predictive accuracy of recommender systems. *INFORMS Journal on Computing* 32(2):303–320.

Choi EW, Özer Ö, Zheng Y (2020) Network trust and trust behaviors among executives in supply chain interactions. *Management Science* .

Culotta A, Cutler J (2016) Mining brand perceptions from twitter social networks. *Marketing science* 35(3):343–362.

Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, 3844–3852.

Dewan S, Ho YJ, Ramaprasad J (2017) Popularity or proximity: Characterizing the nature of social influence in an online music community. *Information Systems Research* 28(1):117–136.

Dong X, Thanou D, Frossard P, Vandergheynst P (2016) Learning laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing* 64(23):6160–6173.

Dziugaite GK, Roy DM (2015) Neural network matrix factorization. *arXiv preprint arXiv:1511.06443* .

Fan W, Ma Y, Li Q, He Y, Zhao E, Tang J, Yin D (2019) Graph neural networks for social recommendation. *The World Wide Web Conference*, 417–426.

Ghose A, Ipeirotis PG, Li B (2014) Examining the impact of ranking on consumer behavior and search engine revenue. *Management Science* 60(7):1632–1654.

Ghose A, Li B, Liu S (2015) Trajectory-based mobile advertising.

Gilpin LH, Bau D, Yuan BZ, Bajwa A, Specter M, Kagal L (2018) Explaining explanations: An overview of interpretability of machine learning. *IEEE International Conference on Data Science and Advanced Analytics*, 80–89.

Goel S, Goldstein DG (2014) Predicting individual behavior with social networks. *Marketing Science* 33(1):82–93.

Granovetter MS (1977) The strength of weak ties. *Social networks*, 347–367 (Elsevier).

Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2018) A survey of methods for explaining black box models. *ACM Computing Surveys* 51(5).

Hamilton WL, Ying R, Leskovec J (2017) Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin* 40(3):42–74.

Hartford J, Graham DR, Leyton-Brown K, Ravanbakhsh S (2018) Deep models of interactions across sets. *International Conference on Machine Learning (ICML)* .

He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. *Proceedings of the 26th international conference on world wide web*, 173–182.

Huang Z, Zeng DD, Chen H (2007) Analyzing consumer-product graphs: Empirical findings and applications in recommender systems. *Management science* 53(7):1146–1164.

Hug N (2020) Surprise: A python library for recommender systems. *Journal of Open Source Software* 5(52):2174, URL `http://dx.doi.org/10.21105/joss.02174`.

Jannach D, Resnick P, Tuzhilin A, Zanker M (2016) Recommender systemsbeyond matrix completion. *Communications of the ACM* 59(11):94–102.

Kalofolias V, Bresson X, Bronstein MM, Vandergheynst P (2014) Matrix completion on graphs. *arXiv:1408.1717*.

Kaya E, Dong X, Suhara Y, Balcisoy S, Bozkaya B, et al. (2018) Behavioral attributes and financial churn prediction. *EPJ Data Science* 7(1):41.

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*.

Kivelä M, Arenas A, Barthelemy M, Gleeson JP, Moreno Y, Porter MA (2014) Multilayer networks. *Journal of Complex Networks* 2:203–271.

Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 426–434.

Koren Y, Sill J (2013) Collaborative filtering on ordinal user feedback. *Twenty-third international joint conference on artificial intelligence*.

Kumar A, Hosanagar K (2019) Measuring the value of recommendation links on product demand. *Information Systems Research* 30(3):819–838.

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521:436–444.

Lee D, Hosanagar K (2020) How do product attributes and reviews moderate the impact of recommender systems through purchase stages? *Management Science* .

Lee DD, Manzoor E, Cheng Z (2018) Focused concept miner (FCM): Interpretable deep learning for text exploration. *Available at SSRN: https://ssrn.com/abstract=3304756* .

Lee JB, Rossi RA, Kim S, Ahmed NK, Koh E (2019) Attention models in graphs: A survey. *ACM Transactions on Knowledge Discovery from Data* 13(6).

Leng Y, Dong X, Moro E, et al. (2018) The rippling effect of social influence via phone communication network. *Complex Spreading Phenomena in Social Systems*, 323–333 (Springer).

Leng Y, Dong X, Pentland A (2020a) Learning quadratic games on networks. *International Conference on Machine Learning (ICML)* .

Leng Y, Sella Y, Ruiz R, Pentland A (2020b) Contextual centrality: going beyond network structure. *Scientific Reports* 10(1):1–10.

Leng Y, Sowrirajan T, Pentland A (2020c) Interpretable stochastic block influence model: measuring social influence among homophilous communities. *arXiv preprint arXiv:2006.01028* .

Li Q, Zeng DD, Xu DJ, Liu R, Yao R (2020) Understanding and predicting users rating behavior: A cognitive perspective. *INFORMS Journal on Computing* .

Li WJ, Yeung DY (2009) Relation regularized matrix factorization. 1126.

Lu H, Halappanavar M, Kalyanaraman A (2015) Parallel heuristics for scalable community detection. *Parallel Computing* 47:19–37.

Ma L, Krishnan R, Montgomery AL (2014) Latent homophily or social influence? an empirical analysis of purchase within a social network. *Management Science* 61(2):454–473.

Maaten Lvd, Hinton G (2008) Visualizing data using t-sne. *Journal of machine learning research* 9(Nov):2579–2605.

Mnih V, Heess N, Graves A, Kavukcuoglu K (2014) Recurrent models of visual attention. *CoRR* abs/1406.6247, URL `http://arxiv.org/abs/1406.6247`.

Monti F, Boscaini D, Masci J, Rodolà E, Svoboda J, Bronstein MM (2017a) Geometric deep learning on graphs and manifolds using mixture model CNNs. *Conference on Computer Vision and Pattern Recognition (CVPR).*

Monti F, Bronstein M, Bresson X (2017b) Geometric matrix completion with recurrent multi-graph neural networks. *Advances in Neural Information Processing Systems*, 3697–3707.

Newman M (2018) Network structure from rich but noisy data. *Nature Physics* 14(6):542–545.

Panniello U, Gorgoglione M, Tuzhilin A (2016) Research notein carss we trust: How context-aware recommendations affect customers trust and other business performance measures of recommender systems. *Information Systems Research* 27(1):182–196.

Rao N, Yu HF, Ravikumar PK, Dhillon IS (2015a) Collaborative filtering with graph information: Consistency and scalable methods. *Advances in Neural Information Processing Systems (NIPS).*

Rao N, Yu HF, Ravikumar PK, Dhillon IS (2015b) Collaborative filtering with graph information: Consistency and scalable methods. *Advances in neural information processing systems*, 2107–2115.

Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2012) Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* .

Rendle S, Zhang L, Koren Y (2019) On the difficulty of evaluating baselines: A study on recommender systems. *arXiv preprint arXiv:1905.01395* .

Rishika R, Ramaprasad J (2019) The effects of asymmetric social ties, structural embeddedness, and tie strength on online content contribution behavior. *Management Science* 65(7):3398–3422.

Rombach MP, Porter MA, Fowler JH, Mucha PJ (2014) Core-periphery structure in networks. *SIAM Journal on Applied mathematics* 74(1):167–190.

Rudin C, Carlson D (2019) The secrets of machine learning: Ten things you wish you had known earlier to be more effective at data analysis. *Operations Research & Management Science in the Age of Analytics*, 44–72 (INFORMS).

Rust RT, Huang MH (2014) The service revolution and the transformation of marketing science. *Marketing Science* 33(2):206–221.

Saar-Tsechansky M, Melville P, Provost F (2009) Active feature-value acquisition. *Management Science* 55(4):664–684.

Scott J (2012) Social Network Analysis. *New York, NY: SAGE Publications* .

Shang C, Liu Q, Chen KS, Sun J, Lu J, Yi J, Bi J (2018) Edge attention-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1802.04944* .

Shuman DI, Narang SK, Frossard P, Ortega A, Vandergheynst P (2013) The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine* 30(3):83–98.

Sismeiro C, Mahmood A (2018) Competitive vs. complementary effects in online social networks and news consumption: A natural experiment. *Management Science* 64(11):5014–5037.

Smola AJ, Kondor R (2003) Kernels and regularization on graphs. *Learning theory and kernel machines*, 144–158 (Springer).

Timoshenko A, Hauser JR (2019) Identifying customer needs from user-generated content. *Marketing Science* 38(1):1–20.

van den Berg R, Kipf TN, Welling M (2017) Graph convolutional matrix completion. *arXiv:1706.02263.*

Van Roy B, Yan X (2010) Manipulation robustness of collaborative filtering. *Management Science* 56(11):1911–1929.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. *Advances in Neural Information Processing Systems*, 5998–6008.

Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2018) Graph attention networks. *International Conference on Learning Representations (ICLR).*

Von Luxburg U (2007) A tutorial on spectral clustering. *Statistics and computing* 17(4):395–416.

Wang X, He X, Wang M, Feng F, Chua TS (2019) Neural graph collaborative filtering. *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 165–174.

Wasserman S, Faust K (1994) Social Network Analysis: Methods and Applications. *Cambridge University Press* .

Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS (2020) A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* .

Xu C, Tao D, Xu C (2013) A survey on multi-view learning. *arXiv preprint arXiv:1304.5634* .

Ye M, Yin P, Lee WC, Lee DL (2011) Exploiting geographical influence for collaborative point-of-interest recommendation. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 325–334 (ACM).

Zhang M, Chen Y (2020) Inductive matrix completion based on graph neural networks. *International Conference on Learning Representations (ICLR)* .

Zhou T, Shan H, Banerjee A, Sapiro G (2012) Kernelized probabilistic matrix factorization: Exploiting graphs and side information. *Proceedings of the 2012 SIAM international Conference on Data mining*, 403–414 (SIAM).