A Deep Learning-based Imputation Method to Enhance Crowdsourced Data on Online Business Directory Platforms for Improved Services

Da Xu[1], Paul Jen-Hwa Hu[2], and Xiao Fang[3]

[1]: Department of Information Systems
College of Business
California State University Long Beach, USA
da.xu@csulb.edu
1250 Bellflower Boulevard, Suite 427, Long Beach, CA 90840.

[2]: Department of Operations and Information Systems
David Eccles School of Business
University of Utah
paul.hu@eccles.utah.edu
1655 East Campus Center Drive, Salt Lake City, UT 84112

[3]: Department of Accounting and Management Information Systems
Alfred Lerner College of Business and Economics
University of Delaware
xfang@udel.edu
20 Orchard Road, Newark, DE 19716

Corresponding Author: Paul Jen-Hwa Hu; paul.hu@eccles.utah.edu; 1655 East Campus Center Drive, Salt Lake City, UT 84112; +1-801-587-7785

Author Bio Sketches

Da Xu is Assistant Professor of Information Systems at the College of Business, California State University Long Beach. He received his Ph.D. in Business Administration from the University of Utah. His current research interests include predictive analytics, health informatics, online digital platforms, deep learning, and data mining for business intelligence. He has published in *Journal of the American Medical Informatics Association*, *IEEE Journal of Biomedical and Health Informatics*, and *Journal of Biomedical Informatics*.

Paul Jen-Hwa Hu is ER Dumke Jr Presidential Endowed Chair in Business at the David Eccles School of Business, the University of Utah. He received his Ph.D. in Management Information Systems from the University of Arizona. His current research interests include information technology for health care, technology implementation and management, business analytics, digital transformation, and technology-enabled learning and knowledge management. Hu has published in *Journal of Management Information Systems*, *Management Information Systems Quarterly*, *Information Systems Research*, *Journal of the AIS*, *European Journal of Information Systems*, *Decision Sciences*, *Journal of Medical Internet Research*, *Journal of the American Medical Informatics Association*, *Journal of Biomedical Informatics*, *Decision Support Systems*, and various IEEE and ACM journals and transactions.

Xiao Fang is Professor of MIS and JPMorgan Chase Senior Fellow at Lerner College of Business & Economics and Institute for Financial Services Analytics, University of Delaware. He also holds appointments at Department of Computer Science as well as Department of Electrical and Computer Engineering, University of Delaware. His current research focuses on financial technology, social network analytics, and health care analytics, with methods and tools drawn from reference disciplines including Computer Science (e.g., Machine Learning) and Management Science (e.g., Optimization). He has published in business journals including *Journal of Management Information Systems*, *Management Science*, *Operations Research*, , *MIS Quarterly*, and *Information Systems Research* as well as computer science outlets such as *ACM Transactions on Information Systems* and *IEEE Transactions on Knowledge and Data Engineering*. Professor Fang received the 2017 INFORMS ISS Design Science Award. He co-founded INFORMS Workshop on Data Science and served as an Associate Editor for MIS Quarterly (2017-2021). He currently serves as an Associate Editor for INFORMS Journal on Data Science and Service Science (INFORMS).

Abstract

Popular online business directory (OBD) platforms, such as Yelp and TripAdvisor, depend on voluntarily user-submitted data about various businesses to assist consumers in finding appropriate options for transactions. Yet the crowdsourced nature of such data restricts the availability of attribute values for many businesses on the platform. Crowdsourced data often suffer serious completeness and timeliness constraints, with negative implications for key stakeholders such as users, businesses, and the platform. We thus develop a novel, deep learning–based imputation method, premised in institutional theory, to estimate missing attribute values of individual businesses on an OBD platform. The proposed method leverages a deep model architecture and considers both inter-business and inter-attribute relationships for imputations. An application to a Yelp data set reveals our method's greater imputation effectiveness relative to prevalent methods. To illustrate the method's practical utilities and values, we further examine the efficacy of business recommendations empowered by its imputed business attribute values, in comparison with those enabled by data imputed by benchmark methods. The results affirm that the proposed method substantially outperforms benchmarks for imputing missing attribute values and empowers more effective business recommendations. This study addresses crucial, prominent completeness and timeliness constraints in crowdsourced data on OBD platforms and offers insights for downstream applications that can improve user experiences, firm performance, and platform services.

Keywords: crowdsourced data, missing value imputation, deep learning, online business directory

Introduction

Online business directory (OBD) platforms, such as Yelp and TripAdvisor, register enormous numbers of businesses; consumers often use them as convenient information sources to find desirable products and services [10]. About 21% of U.S. consumers use OBD platforms to search for local businesses on a daily basis.[1] Platforms leverage their rich data to connect consumers with appropriate businesses by offering personalized recommendations. Consider Yelp, with its 5.8 million active claimed local business locations and 83 million unique users monthly (as of 2021): It provides a dedicated webpage for each business and deploys search engine and recommender systems to help users find preferred businesses, request price quotes, make reservations, join waitlists, or complete transactions.[2]

On an OBD platform, most business data pertain to business attributes and characteristically are crowdsourced—that is, contributed voluntarily by users instead of business owners. For example, Yelp relies on users to obtain essential business attribute values (e.g., price, services, amenities), both factual and subjective. It prompts such contributions, such as by asking "Does this restaurant have WiFi?" on its platform webpage.[3] If sufficient responses are submitted by the crowd, the "WiFi" attribute value appears on the restaurant's webpage; otherwise, the value is not available to consumers; i.e., "missing" on its page.

An OBD platform's services essentially help consumers identify, connect, and transact with desired businesses. These services can be enhanced by crowdsourced business attribute data, which benefits all major stakeholders that include consumers, the platform, and businesses on the platform. First, displayed business attributes offer convenient access to users, who often use them

[1] https://www.statista.com/statistics/315709/online-local-business-search-frequency-us-canada/#statisticContainer, accessed on April 20, 2022.
[2] https://www.yelp-ir.com, accessed on April 20, 2022.
[3] On Yelp, each restaurant is described by more than 80 attributes, for which most values are crowdsourced.

to compare different (competing) businesses for transaction choice. According to Yelp, 58% of its users value the availability of business attributes and characteristics for their comparisons and transaction decisions, in addition to reviews and ratings.[4] Second, with crowdsourced business attribute data, OBD platforms can better assist users and improve their experiences. These business attribute data enhance the effectiveness of search engines and recommender systems to suggest appropriate businesses and filter out less relevant options. To illustrate, people planning for a casual dinner with friends can check the "Good for Groups" option on Yelp, when they search businesses to identify preferred restaurants. Similarly, businesses appear in relevant search results when people use specific attributes to define their searches for businesses on Google Search and Google Maps.[5] The facilitated decision-making and reduced cognitive processing should enhance users' experience and satisfaction with the platform [17, 21, 35]. In a related sense, efficacious user–service interactions and recommendations can attract more businesses to join the platform, sharpen its targeted advertising, and create positive network effects for increasing revenues and competitiveness [23]. Third, crowdsourced attribute data benefit individual businesses by elevating their visibility to and searchability by consumers, increasing the likelihood of being recommended by the platform, and fostering their reputations and customer loyalty [47].

However, crowdsourced business attribute data often are incomplete, which makes their direct use difficult and less effective. To have an attribute value appear on a business's webpage, the platform needs to receive sufficient responses or votes from the crowd. But the voluntary nature of crowdsourced data means the platform cannot demand or control data-gathering efforts, which creates difficulties in ensuring data quality [72]. Typically, the crowd consists of many passive

[4] https://blog.yelp.com/2019/10/study-shows-97-of-people-buy-from-local-businesses-they-discover-on-yelp, accessed on April 20, 2022.
[5] https://support.google.com/business/answer/9049526?hl=en, accessed on April 20, 2022.

users and casual content contributors, with no obligations and few motivations to comply with the platform's data solicitation and collection efforts [48, 49, 84]. As a consequence, many businesses' attribute values are not available on their platform pages. In line with Gupta and Singh [26], we consider unavailable business attribute values as "missing," because the business likely has values for these attributes in actuality (e.g., yes or no), but they are not displayed on its platform page due to the lack of sufficient responses. Moreover, the availability of attribute values for individual businesses differs substantially. For example, popular restaurants with an enormous customer base likely have more attribute values available on the platform than small, lesser-known ones that struggle to prompt enough responses from the crowd to have their attribute values shown on platform pages.

This missing value problem becomes even more serious when the OBD platform adds new business entities or additional attributes, because it takes time to receive the needed responses from the crowd. This temporal latency, manifested by the time interval between the introduction of a new business entity or attribute and the availability of its value(s) on OBD pages, worsens the missing data problem. For example, during the COVID-19 pandemic outbreaks, Yelp responded to people's altered dining preferences by introducing new health and safety attributes to businesses' pages, such as curbside pickups, enforcing limited capacity, and accepting contactless payments. However, the values of these important, newly added attributes remained blank for many businesses, due to insufficient responses by the crowd. The increasing completeness and timeliness constraints of crowdsourced data on OBD platforms limited their services and performance, especially during the challenging COVID-19 era.

Data completeness and timeliness need to be properly addressed to alleviate their restrictive effects on the platform's services, such as business profiling, searches, and recommendations. As

Lukyanenko et al. [50] and Wang et al. [79] indicate, completeness and timeliness constitute critical dimensions of data (information) quality. On an OBD platform, businesses with many missing attribute values may be less searchable and therefore are likely to lose valuable transaction opportunities. For example, if a restaurant's "Good for Groups" attribute value is missing on Yelp, it is not visible to consumers who need this feature, regardless of whether it has appealing services and spacious venues for groups. Moreover, the rank order of businesses displayed on the platform also might be influenced by missing values, which is particularly critical for mobile app users who view lists of recommended businesses on a small screen, with profound implications for user satisfaction, conversion, and purchase decisions [22].

In recognition of these important implications, this study seeks to enhance the crowdsourced data by estimating businesses' missing attribute values on an OBD platform, in pursuit of improvements to the services available to both users and businesses on the platform. To address data completeness and timeliness constraints, we propose a novel, deep learning–based imputation method that is premised in institutional theory [55]. According to this theory, businesses in similar institutional contexts (e.g., same sector, similar customers) learn from one another's decisions and practices, so their attribute values might be somewhat related (e.g., similar or complementary). Guided by this theoretical lens, our imputation method considers inter-business relationships and social learning, then imputes missing attribute values of a focal business by leveraging the observed attribute values of related businesses. In addition, the proposed method incorporates a novel learning strategy to emulate the process of imputing missing values by randomly masking observed attribute values of a business and training a model to recuperate these values.

We use a real-world Yelp data set to evaluate the proposed method, in comparison with several prevalent benchmark methods. The results demonstrate its superior imputation efficacy, relative

to existing methods. We illustrate the practical utilities and value of its imputed, complete attribute values for supporting the platform's business recommendations. The results reveal that OBD platforms, supported by our method, can better estimate missing attribute values and leverage them for efficacious services, such as effective business recommendations and timely information availability. Pragmatically, platforms can avoid confusion and potentially misleading information by clearly specifying the sources of displayed business attribute values as crowdsourced or estimated, as exemplified by Yelp's statement that COVID-19–related health and safety attributes are obtained "according to most users."

Literature Review

Several streams of research are closely related to our study. We review representative studies in each stream and specify the gaps that motivate our work.

Online Business Directories and Crowdsourced Business Attribute Data

By connecting businesses and customers, an OBD platform is more than a passive directory listing [58]. It enables the integral participation of users into self-directed searches for businesses with which to transact, according to their needs, wants, and preferences [60]. An OBD platform offers, but is not limited to, business profiling, searches, recommendations, and targeted advertising. Business profiling organizes and delivers essential information to help consumers understand each business and its offerings (e.g., services, products), typically in the form of business attributes that constitute the business's profile. When these attributes have missing values, it can undermine consumers' decision-making and experiences on the platform. The search function allows people to select particular attributes, specify filters, or enter keywords to obtain a choice set that matches their requirements and preferences. Businesses with many missing attribute values are disadvantaged, simply because they cannot be considered for inclusion in the choice set if they do

not have values for user-specified attributes or features. Furthermore, OBD platforms make business recommendations by incorporating users' preferences and contextual factors (e.g., time, location, consumption goal), which can be significantly affected by missing values [86]. For example, a platform may suggest nearby restaurants that match the "offers delivery" feature to users who frequently order takeout on the platform. But if a restaurant's "offers delivery" attribute value is missing, the platform cannot match its attribute and recommend this restaurant to users who want food delivery. In all these scenarios, businesses with fewer attribute values available on their platform pages are less searchable and recommendable, leading to a long-tail distribution phenomenon [2]. Finally, effective targeted advertising by businesses on an ODB platform require precision to identify target customers, which can be hindered by incomplete business attribute data. As a result, both the businesses' revenues and the platform's service performance decrease.

Although crowdsourcing generates enormous amounts of data, they often are of suboptimal quality, especially in terms of completeness and timeliness [44, 51, 68]. Completeness refers to the degree to which all required values are properly collected and available in the data [6]. Timeliness pertains to whether data are available in time for further processing and use [40]. These two fundamental dimensions of data quality not only determine the fitness and value of data for use [4] but also have significant effects on consumer decision-making and firm performance [6]. Because missing values in crowdsourced business attribute data undermine data completeness and timeliness, the data become less useful for both consumers and the platform [51]. However, data completeness and timeliness constraints prevail in OBD platforms and are difficult to address, mainly because users voluntarily contribute business attribute data and have few incentives to do so [49, 50]. Moreover, when a platform introduces new attributes or entities, the crowd needs time to submit sufficient responses. In summary, crowdsourced data significantly vary in completeness

and timeliness and frequently exhibit a long-tail distribution across different businesses.

Prevalent Imputation Methods

Imputation represents a viable, common approach to address missing values in crowdsourced data. In a nutshell, missing values often stem from different mechanisms, such as missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). In general, MCAR, which is completely random, does not relate to any available or missing values; MAR depends on available (i.e., observed) data but not missing data; and MNAR implies that a missing value reflects itself or other missing values [45, 81]. On an OBD platform, missing business attribute values occur due to a lack of sufficient data contributions by users. Such missingness depends on the number of contributing users (i.e., reviewers) instead of the missing values themselves or other unobserved variables. Therefore, the underlying mechanism likely is either MCAR or MAR. Most imputation methods, including multiple imputation [3], SoftImpute [54], and KNNImpute [74], assume MCAR or MAR and thus use complete values to estimate missing values, without explicitly modeling the underlying missing mechanism [45].

Existing imputation methods can be categorized as model–, representation learning–, or similarity–based. A model–based method builds a probabilistic, regression, or machine learning model to learn the data structures and inter-attribute relationships [3, 15, 30, 42, 62, 66]. Multiple imputation [3, 15, 42, 66] represents a prevalent model–based method; it relies on available (observed) values to construct a probabilistic model for imputations by replacing each missing value with a set of values derived from a joint or conditional distribution. For example, multiple imputation by chained equations (MICE) imputes missing values by calculating the conditional distribution of one attribute on all other attributes, until all attributes have been imputed [3].[6] Yet

[6] The imputation process is performed iteratively until it reaches a prespecified termination threshold.

the use of a joint or conditional distribution could generate models that are incompatible if the joint distribution cannot exist or the chained equation does not adequately reflect the complex relationships among variables [76]. Regression models are another exemplary form of model–based methods [30, 62, 74]. Typically, they regress variables with incomplete values on those with complete values, excluding variables with missing values, so they may not be effective for imputing missing values. Overall, a model–based method's performance can be confined by model structure or assumptions, such that it provides limited flexibility and scalability.

Representation learning–based methods avoid modeling the variable relationships directly [29, 30, 42, 46, 54, 64, 69, 74]. Instead, they use mapping mechanisms to project the original, incomplete data into a substantially reduced dimensional space; impute the missing values by learning a lower-dimensional approximation; and finally project that approximation back to the original feature space. For example, SoftImpute fits a low-rank matrix approximation to an original data matrix [54], using soft threshold singular value decomposition (SVD) and nuclear norm regularization. The resulting low-rank matrix approximation contains no missing values, so it reconstructs a data matrix with all missing values imputed. This imputation process usually involves the use of a linear transformation to learn a representation in lower-dimensional space, which it then projects back to the original space with the imputed values. Deep autoencoder imputation goes one step further, employing a deep model architecture to perform multiple nonlinear transformations (i.e., mappings) for greater imputation performance [8]. In general, existing representation learning–based methods use mapping mechanisms to learn the relationships among attribute values but often struggle to capture relationships at the instance level, which can be crucial for effective imputations.

Similarity–based methods impute missing values on the basis of instance-level similarity [38,

61, 74, 75, 85]. Similar instances get selected according to a similarity measure, and then their complete values serve to impute missing values of a focal instance. Exemplary similarity-based methods depend on variables [74] or iterations [11, 61]. A variable–based method uses complete variable values to assess instance similarity, whereas an iterative–based method calculates similarity, according to all variable values, and then updates the imputed values iteratively. An important challenge for similarity–based methods is that the calculated similarities might be inaccurate, due to the presence of missing values. The similarity measurement also is sensitive to data sparsity and tends to suffer dimensionality issues [53], because computational complexity greatly increases in high-dimensional spaces.

Research Gaps

This review of extant literature reveals two important gaps. First, existing imputation methods [3, 54, 78] tend to ignore the relationships among instances (e.g., businesses). Conceivably, businesses on an OBD platform face similar institutional environments and could learn from one another's decisions and practices [5, 43], thereby leading to knowledge transfers and homogenization [63]. As a result, their attribute values might be related to some degree. This unique perspective suggests the consideration of inter-business relationships for imputing missing business attribute values, which however has been overlooked by most previous research. Second, many imputation methods oversimplify the relationships among variables, due to their model structure's scalability and flexibility constraints (e.g., regression or probabilistic models). Crowdsourced data are enormous in volume and could have substantial missing values on business attributes that differ in data type (e.g., categorical, numerical) [44, 51]. These attributes might relate in intricate ways [65]. For example, a restaurant with attribute "Good for Groups" might imply "Good for Dinner." Jointly, these challenging characteristics favor the use of deep learning to capture complex relationships

and subtle patterns that are important for missing value imputations, with increasing scalability, fitting ability, and flexibility.

To alleviate these gaps, we develop a novel, deep learning–based imputation method to enhance crowdsourced business attribute data on an OBD platform. The proposed method is novel in three respects. First, guided by institutional theory, this method leverages inter-business relationships to impute missing attribute values of individual businesses. It analyzes common customers among different businesses to identify related businesses, then integrates the observed values of these related businesses through an attention mechanism. Unlike similarity–based methods that rely on attribute values to assess the similarity among businesses, our method takes a network approach to scrutinize inter-business relationships. Second, the proposed method employs an encoder–decoder structure that includes two encoders: one that takes the attribute values of a focal business as inputs and another that uses the integrated attribute values of its related businesses as inputs. Third, our method incorporates an innovative learning strategy for model building. This method is novel relative to conventional autoencoders, in that its encoder–decoder component can emulate the missing value imputation process by masking attributes, and then it learns to impute with an innovative loss function for increased effectiveness.

Theoretical Foundation

Firms' decisions and behaviors can be explained by institutional theory [55], which informs our method conceptualization. Scot and Meyer [67] indicate that organizational environments "are characterized by the elaboration of rules and requirements to which individual organizations must conform if they are to receive support and legitimacy" (p. 149). The institutional theory recognizes the isomorphism that exists among firms facing similar institutional environments or contexts [68], such that "organizational isomorphism increases organizational legitimacy" [14] (p. 1033). This

theoretical lens underscores the significance of notable attributes when comparing organizations, both within and among firms. For example, a company's brand name could be used to infer its other attributes, such as market position, value, and price [24]. In addition, essential attributes frequently used by different firms might correlate, because their deployments by firms imply their importance for establishing organizational legitimacy [13]. Whether symbolic or functional [68], organizational isomorphism features three related but distinct forms: coercive, normative, and mimetic [56].

Coercive isomorphism incites changes through institutional pressures, to which firms respond in their pursuit of legitimacy [16]. Coercive isomorphism may arise, due to pressures from other firms to which a focal firm relates (e.g., competition, dependency) or from the firm's own pressure to conform to the expectations of the market or larger society [56]. As Scott [68] explains, firms comply to gain approval and avoid punishment, regardless of whether these acts relate closely to their core efforts. To illustrate, restaurants are subject to regulations, industry norms, and safety guidelines, so they likely make operational adjustments to existing practices and conditions to meet these expectations. Restaurants' attributes then might reflect their conformity to coercive forces, as in the cases of wheelchair accessibility or takeout options [41, 56]. Coercive isomorphism is symbolic and could exert intra-firm effects, so the value of an attribute might convey information to support inferences about the values of other attributes. For example, a listed average price of "$$$" reflects a high-end restaurant, and this categorization could support inferences about whether it has private parking and good service. Therefore, important relationships may exist among different attribute values, which helps infer a business's missing attribute values from other attribute values. Furthermore, coercive isomorphism prompts firms to learn and respond to external challenges [43] through an adaptive learning process. A restaurant might observe how

other restaurants cope with emerging trends and features, learn from their practices, and adjust its own operations accordingly. As a result, attribute values for different firms might be linked in an important and subtle way [52], which suggests the rationale for and legitimacy of imputing missing attribute values for a business using observed values for other businesses.

Normative isomorphism is a type of functional isomorphism and suggests standard solutions to common business problems. In an identification process, firms establish a cognitive base in their efforts to maintain positive, beneficial relationships with other professional peers and contacts [16]. Many firm-level operations decisions come from managers who tend to share similar backgrounds and characteristics (e.g., education, experiences, inter-firm hiring network) and are likely to adopt similar standards, practices, and operational procedures. In light of normative isomorphism, firms could become increasingly homogenous by adopting solutions implemented by other firms [9, 16], because they often identify with well-regarded peers and professionals, and implement standard practices to gain organizational legitimacy.

Finally, mimetic isomorphism propels firms to establish organizational legitimacy by internalizing the social influences and values exhibited by others. To some extent, firms model their decisions and actions by observing others' choices [20]; as a result, a firm's practices might be shaped in response to those of the larger peer group. This proactive mimicry offers a common response to environmental uncertainty [9, 16]. As Teo et al. [73] (p. 21) indicate, firms imitate "structurally equivalent organizations because those organizations occupy a similar economic network position in the same industry and, thus, share similar goals, produce similar commodities, share similar customers and suppliers, and experience similar constraints." If firms facing same market environments and competing for similar customers continually learn from one another, knowledge transfers should result [63]. This view is congruent with the social learning perspective;

that is, restaurants that attract similar customers observe and learn from one another [43], so they can acquire knowledge, adjust practices, and avoid the costs of trying new things on their own [27, 28, 37, 57, 83]. In this sense, a restaurant's attributes might relate to those that target and attract similar customers, due to learning effects. Through learning, individual businesses might adopt similar or complementary attributes that are beneficial to them. Overall, an institutional theory lens, combined with social learning [5], suggests the importance of inter-business and inter-attribute relationships for addressing completeness and timeliness constraints in crowdsourced business data. We therefore leverage these relationships to better impute a business's missing attribute values on an OBD platform.

Attribute Enhancement Problem and Proposed Imputation Method

In this section, we present our problem formulation and elaborate the proposed method.

Problem Formulation

Let $\mathbb{U} = \{u_1, u_2, \dots, u_M\}$ denote a set of M users and $\mathbb{B} = \{b_1, b_2, \dots, b_N\}$ represent a set of N businesses on an OBD platform. A user–business interaction matrix $\boldsymbol{R} \in \mathbb{R}^{M \times N}$ can be constructed from users' interactions with different businesses on the platform. An element of the matrix, R_{mn}, indicates the interaction between user u_m and business b_n (e.g., u_m rating b_n), where $m = 1,2, \dots, M$ and $n = 1,2, \dots, N$. We can use the user–business interaction matrix $\boldsymbol{R}$ to identify businesses that share common users and further construct inter-business relationships. We represent crowdsourced business attribute data with a matrix $\boldsymbol{D} \in \mathbb{R}^{N \times L}$, which contains attribute values for each of the N businesses over L different attributes. An element D_{nl} of $\boldsymbol{D}$ indicates the value of the lth attribute for business b_n, where $n = 1,2, \dots, N$ and $l = 1,2, \dots, L$; it is set to NA (not available) if the value is missing. Specifically, we use X_{ij} to denote an element of matrix $\boldsymbol{X}$. The ith row of $\boldsymbol{X}$ is represented as $\boldsymbol{X}_{i\cdot}$, whereas $\boldsymbol{X}_{\cdot j}$ is the jth column of the matrix. The attribute

enhancement problem is formally formulated as follows:

Given a user–business interaction matrix $\boldsymbol{R}$ *and crowdsourced business attribute data* $\boldsymbol{D}$*, the objective of the attribute enhancement problem is to enhance* $\boldsymbol{D}$ *by generating accurate imputations for all of its missing values through an effective use of* $\boldsymbol{R}$*.*

Figure 1 presents the overall framework of our proposed method that includes three important components: data preprocessing, auxiliary attribute matrix construction, and imputation. The data preprocessing component prepares inputs for the next two components by processing the business attribute data $\boldsymbol{D}$ and user–business interaction matrix $\boldsymbol{R}$. The auxiliary attribute matrix construction component identifies businesses engaged in social learning and extracts their attribute data. With the imputation component, we develop a novel imputation model to impute missing values in $\boldsymbol{D}$. Next, we describe each component and highlight the novelties of our proposed method.

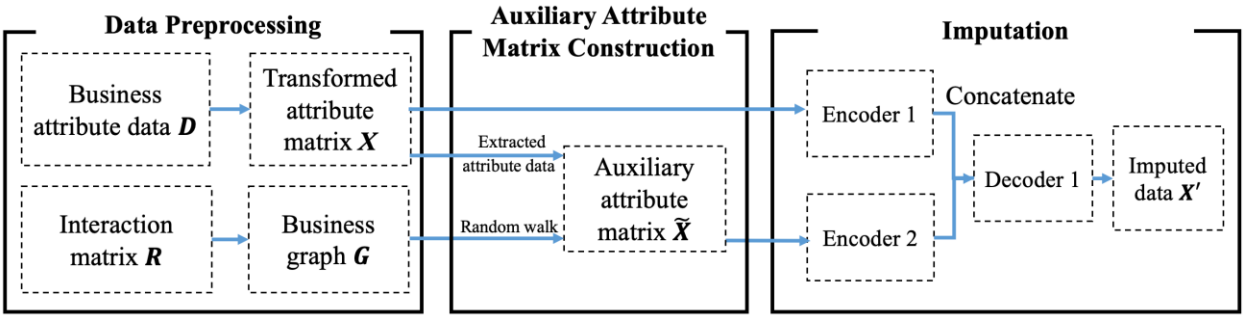

Figure 1: Overall Framework of the Proposed Imputation Method

Data Preprocessing

We follow a common procedure to normalize each numeric element of $\boldsymbol{D}$ through z-score standardization and represent each categorical element of $\boldsymbol{D}$ with one-hot encoding, thereby creating the transformed attribute matrix $\boldsymbol{X}$. A numeric element D_{nl} of $\boldsymbol{D}$ is transformed to $\frac{D_{nl}-\mu(\boldsymbol{D}_{\cdot l})}{\sigma(\boldsymbol{D}_{\cdot l})}$, where $\mu(\boldsymbol{D}_{\cdot l})$ and $\sigma(\boldsymbol{D}_{\cdot l})$ denote the mean and standard deviation of the lth attribute, respectively. If D_{nl} is categorical, its one-hot encoding is a row vector [0, 0, …, 1, …, 0], for which

the size of the vector is equal to the number of possible categories of the *l*th attribute [25]. In this vector, the element that corresponds to the category of D_{nl} is set to 1, and all other elements equal 0.

Example 1: For illustration, consider an example of business attribute data $\boldsymbol{D}$ in Figure 2**,** which describes 8 businesses with 5 attributes. The numerical attribute "Review count" indicates the number of reviews received by a business; its mean and standard deviation are 119 and 120.44, respectively. Applying the z-score standardization, a "Review count" of 349 in $\boldsymbol{D}$ is normalized to 1.91 in $\boldsymbol{X}$. The categorical attribute "Alcohol" takes one of three values: "none" (no alcohol), "beer and wine," or "full bar." Thus, its one-hot encoding is [0, 0, 1]. All missing values in $\boldsymbol{D}$ are still treated as missing (*NA*) in $\boldsymbol{X}$.

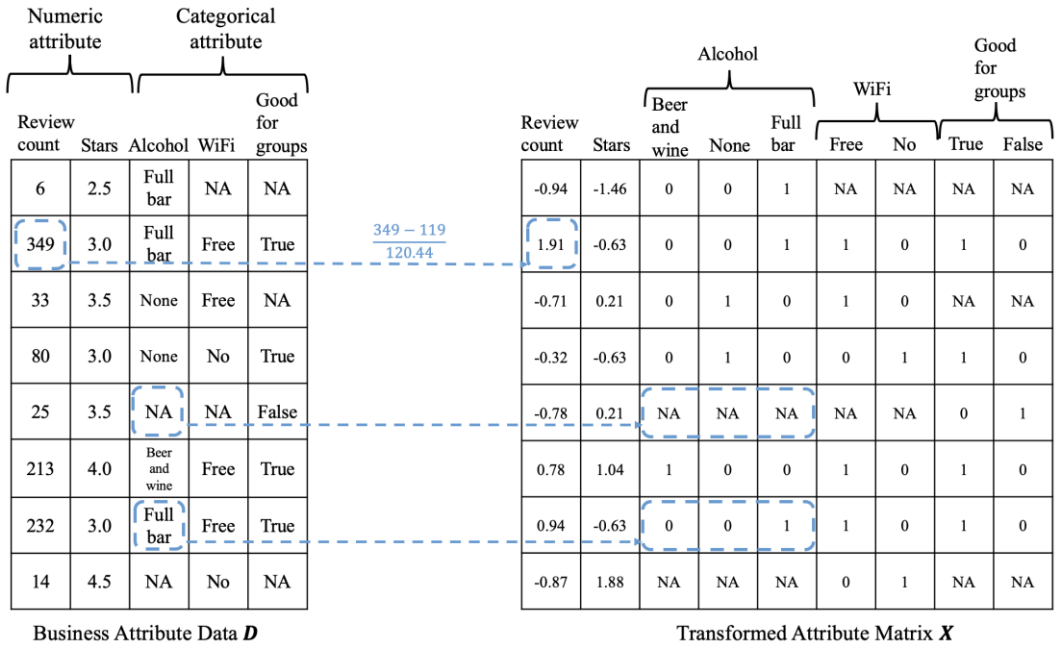

Figure 2: An Example of Transforming Business Attribute Data

We preprocess the user–business interaction matrix $\boldsymbol{R}$ to construct a business graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$. Each vertex in $\boldsymbol{V} = \{v_1, v_2, \dots, v_N\}$ represents a business, and each edge in $\boldsymbol{E}$ connects a pair of vertices. According to institutional theory [55], businesses that attract similar customers

could engage in social learning, so they might exhibit isomorphism in business attribute values. If two vertices (i.e., businesses) v_i and v_j involve the same group of customers (e.g., rated by them), they are linked by edge $e_{ij} \epsilon \boldsymbol{E}$, with the edge weight w_{ij} indicating the number of common users.

Auxiliary Attribute Matrix Construction

We use the transformed attribute matrix $\boldsymbol{X}$ and business graph $\boldsymbol{G}$ to construct the auxiliary attribute matrix $\widetilde{\boldsymbol{X}} \in \mathbb{R}^{N \times L}$, which complements $\boldsymbol{X}$, to impute missing business attribute values. In light of institutional theory [55], the missing value of a business's attribute could be imputed from the observed values of that attribute for businesses that serve as sources of its social learning, which we refer to as its neighboring (i.e., related) businesses. Therefore, for each business, we seek to identify its "neighboring businesses" and represent their attribute values in $\widetilde{\boldsymbol{X}}$, which mitigates the sparsity of $\boldsymbol{X}$ and relaxes the data availability constraint.

Two businesses in the business graph $\boldsymbol{G}$ are linked by an edge if they share the same group of customers and the weight of the edge can be determined by the number of common users. To identify neighboring businesses of a focal business, we need to consider two factors: (1) the distance between the focal business and another business, measured by the number of edges on the paths connecting them, and (2) the weights of these edges. Smaller distances and larger edge weights imply a higher likelihood that these businesses are neighbors of the focal business and thus exert greater influences. To consider both distance and edge weights, we apply a random walk algorithm, which is preferable over techniques that directly select vertices connected to the focal vertex according to a certain distance and edge weights (e.g., *k*-th order neighbor [1]). Our choice of random walk offers two key benefits. First, the random walk can better model the combined effect of distance and edge weight, without using prespecified cut-off values, whereas techniques that select neighboring businesses directly from the network may require distance and edge weight

thresholds. Second, businesses with smaller distances and larger edge weights can be visited multiple times by the random walk. As a result, these businesses carry larger weights when we calculate the auxiliary attribute matrix, reflecting their likely greater influences on the focal business. Accordingly, we develop an algorithm to identify and represent neighboring businesses by incorporating both factors. For each focal business, a sequence of K neighboring businesses can be identified by the random walk, starting from the vertex representing the focal business. By considering the weights of the edges, we define the transition probability of the random walk as:

$$P(s^{next} = v_n | s^{current} = v_m) = \frac{w_{mn}}{\sum_{v_l \in V} w_{ml}}, \quad (1)$$

where $s^{current}$ and s^{next}, respectively, refer to the current and next vertexes visited by the walk; $v_m, v_n \in \boldsymbol{V}$; and w_{mn} and w_{ml} are the weights of the edges e_{mn} and e_{ml}, respectively, such that $w_{mn} = 0$ or $w_{ml} = 0$ if edge e_{mn} or e_{ml} does not exist. At each step, the random walk returns to its starting point (i.e., vertex representing the focal business) with probability θ, $\theta \in (0,1)$, or else proceeds to the next vertex, according to Equation (1), with probability $1 - \theta$. This restart procedure ensures that businesses closer to the focal business are more likely and frequently visited by the random walk and get included in its neighboring businesses, thereby reflecting the distance factor. In summary, for each focal business b_i, a random walk with restart navigates from vertex $s^{(0)} = v_i$ and produces a K-sequence $S^i = \{s^{(1)}, s^{(2)}, \dots, s^{(k)}, \dots, s^{(K)}\}$, where $s^{(k)}$ denotes the kth neighboring businesses of b_i, $k = 1,2, \dots, K$. Important neighboring businesses that likely have significant impacts on the focal business's attribute values could appear multiple times in its K-sequence of neighboring businesses.

After identifying the sequence of neighboring businesses S^i for a focal business b_i, we construct its neighboring attribute matrix $\boldsymbol{A}^i \in \mathbb{R}^{K \times L}$, the kth row of which contains attribute values of its kth neighboring business, $k = 1,2, \dots, K$. Inspired by Vaswani et al. [77], we develop

a focal-neighbor attention mechanism to integrate attribute values of b_i's neighboring businesses into a vector $\widetilde{\boldsymbol{X}}_{i\cdot}$, which is the weighted sum of the row vectors in $\boldsymbol{A}^i$:

$$\widetilde{\boldsymbol{X}}_{i\cdot} = \sum_{k=1}^{K} \alpha_k^i \boldsymbol{A}_{k\cdot}^i , \tag{2}$$

where the vector $\boldsymbol{\alpha}^i = [\alpha_1^i, \alpha_2^i, \dots, \alpha_K^i]$ contains the weights of b_i's neighboring business, $\sum_{k=1}^{K} \alpha_k^i = 1$, and $\boldsymbol{A}_{k\cdot}^i$ is the kth row of $\boldsymbol{A}^i$. A neighboring business should have a larger weight if it is more related (i.e., similar) to the focal business, in terms of attribute values. We compute the weight of b_i's kth neighboring business α_k^i as:

$$\alpha_k^i = \frac{exp(\phi(\boldsymbol{q}_i, \boldsymbol{p}_k)/\lambda)}{\sum_{j=1}^{K} exp(\phi(\boldsymbol{q}_i, \boldsymbol{p}_j)/\lambda)}, \tag{3}$$

where $k = 1,2, \dots, K$, and $\lambda > 0$. In this equation, $\boldsymbol{q}_i = \boldsymbol{X}_{i\cdot}\boldsymbol{W}^A$, $\boldsymbol{p}_k = \boldsymbol{A}_{k\cdot}^i\boldsymbol{W}^A$, and $\boldsymbol{W}^A \in \mathbb{R}^{L\times d_p}$ is a weight matrix to be learned, through which the business attribute values of b_i and those of its kth neighboring business can be mapped to a d_p-dimensional hidden space. Function $\phi(\cdot,\cdot)$ computes the cosine similarity of vectors $\boldsymbol{q}_i$ and $\boldsymbol{p}_k$. The parameter λ controls the effect of the attention mechanism: A smaller λ increases weights for neighboring businesses more similar to the focal business and thus enhances this effect. Figure 3 depicts the attention mechanism. We elaborate the algorithm for constructing the auxiliary attribute matrix in Figure 4.

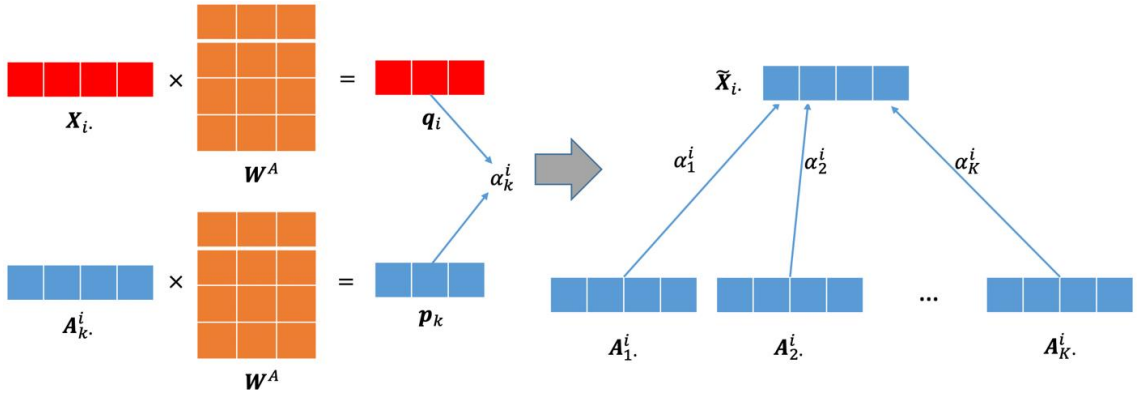

Figure 3: Focal-Neighbor Attention Mechanism

Input: transformed attribute matrix $\boldsymbol{X}$ and business graph $\boldsymbol{G}$
Output: auxiliary attribute matrix $\widetilde{\boldsymbol{X}}$

For each vertex v_i in $\boldsymbol{G}$
$S^i = \{\}$. // initialize neighboring businesses of b_i
$s^{current} = v_i$.
While $length(S^i) < K$
With probability $1-\theta$, generate s^{next} according to transition probability $P(s^{next}|s^{current})$.
With probability θ, $s^{next} = v_i$.
If $s^{next} \neq v_i$
Add s^{next} to S^i.
End if
$s^{current} = s^{next}$.
End while
Create neighboring attribute matrix $\boldsymbol{A}^i$ based on S^i.
Compute auxiliary attribute vector $\tilde{X}_{i\cdot}$ with $\boldsymbol{A}^i$ and the focal-neighbor attention mechanism.
End for
Stack vectors $\widetilde{\boldsymbol{X}}_{i\cdot}$, $i = 1,2,\dots,N$, to form $\widetilde{\boldsymbol{X}}$.
Return $\widetilde{\boldsymbol{X}}$

Figure 4: Algorithm for Constructing Auxiliary Attribute Matrix $\tilde{X}$

Imputation

With $\widetilde{\boldsymbol{X}}$ constructed, the imputation component proceeds as illustrated in Figure 5. The inputs to the component include the transformed attribute vector $\boldsymbol{X}_{i\cdot}$, which contains attribute values of business b_i (both observed and missing), and the auxiliary attribute vector $\widetilde{\boldsymbol{X}}_{i\cdot}$ represents the aggregated attribute values of b_i's neighboring businesses. The output of the component is vector $\boldsymbol{X'}_{i\cdot}$ that contains imputed values for b_i's attributes that originally were missing. In this component, encoder1 takes the corrupted attribute vector $\widehat{\boldsymbol{X}}_{i\cdot}$ as input, encoder2 has vector $\widetilde{\boldsymbol{X}}_{i\cdot}$ as input, and the decoder outputs vector $\boldsymbol{X'}_{i\cdot}$. The corrupted attribute vector $\widehat{\boldsymbol{X}}_{i\cdot}$ is derived from $\boldsymbol{X}_{i\cdot}$, which we detail next. In line with the naming conventions for autoencoder [33], the inputs to an encoder constitute an input layer, the output of the decoder is the output layer, and hidden layers lie between them.

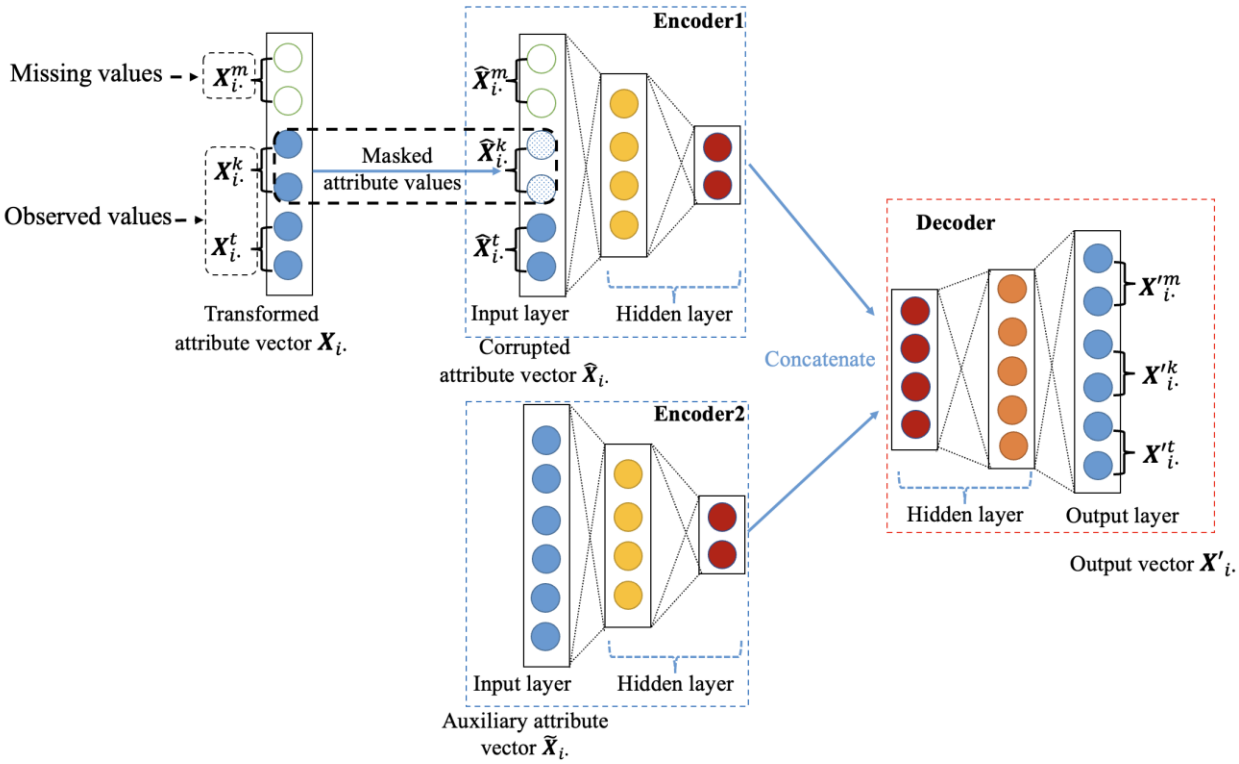

Figure 5: Imputation Component of the Proposed Method

The imputation component emulates the process of missing value imputation. As Figure 5 illustrates, we randomly select some available attributes (i.e., those with observed values) in $\boldsymbol{X}_{i.}$, mask their values, and treat them as missing values. The imputation component is then applied to recover the masked values. The masked values are tracked and compared with the recovered values for model training. When sufficiently trained, imputation component can impute values for attributes with actual missing values. Specifically, we divide the attributes in $\boldsymbol{X}_{i.}$ into three categories: attributes $\boldsymbol{X}_{i.}^{m}$ with missing values, attributes $\boldsymbol{X}_{i.}^{k}$ with observed but masked values, and attributes $\boldsymbol{X}_{i.}^{t}$ with observed values (i.e., intact attributes). Next, we create the corrupted attribute vector $\hat{\boldsymbol{X}}_{i.}$ with the same dimensions as $\boldsymbol{X}_{i.}$. The values of intact attributes in $\boldsymbol{X}_{i.}$ are directly copied to $\hat{\boldsymbol{X}}_{i.}$, so $\hat{\boldsymbol{X}}_{i.}^{t} = \boldsymbol{X}_{i.}^{t}$. Following a dropout procedure [70], we set value-masked attributes to 0 in $\hat{\boldsymbol{X}}_{i.}$, or $\hat{\boldsymbol{X}}_{i.}^{k} = \boldsymbol{0}$. In addition, attributes with missing values are set to 0 in $\hat{\boldsymbol{X}}_{i.}$, such that $\hat{\boldsymbol{X}}_{i.}^{m} = \boldsymbol{0}$. As a

result, attributes with corrupted or missing values do not affect the computations in the hidden and output layers [70]. We illustrate how to construct $\widehat{\boldsymbol{X}}_{i\cdot}$ from $\boldsymbol{X}_{i\cdot}$ with another example.

Example 2: Consider the transformed attribute vector $\boldsymbol{X}_{i\cdot}$ of business b_i in Figure 6. Among b_i's five attributes, the values of attributes "Review count," "Alcohol," "Stars," and "WiFi" are observed, but the value of "Good for Groups" is missing. We randomly select attributes "Stars" and "WiFi" and mask their values, setting them to 0 in the corrupted attribute vector $\widehat{\boldsymbol{X}}_{i\cdot}$. The values of the intact attributes in $\widehat{\boldsymbol{X}}_{i\cdot}$ (i.e., "Review count" and "Alcohol") remain the same as those in $\boldsymbol{X}_{i\cdot}$, and the value of the missing attribute "Good for Groups" is set to 0 in $\widehat{\boldsymbol{X}}_{i\cdot}$.

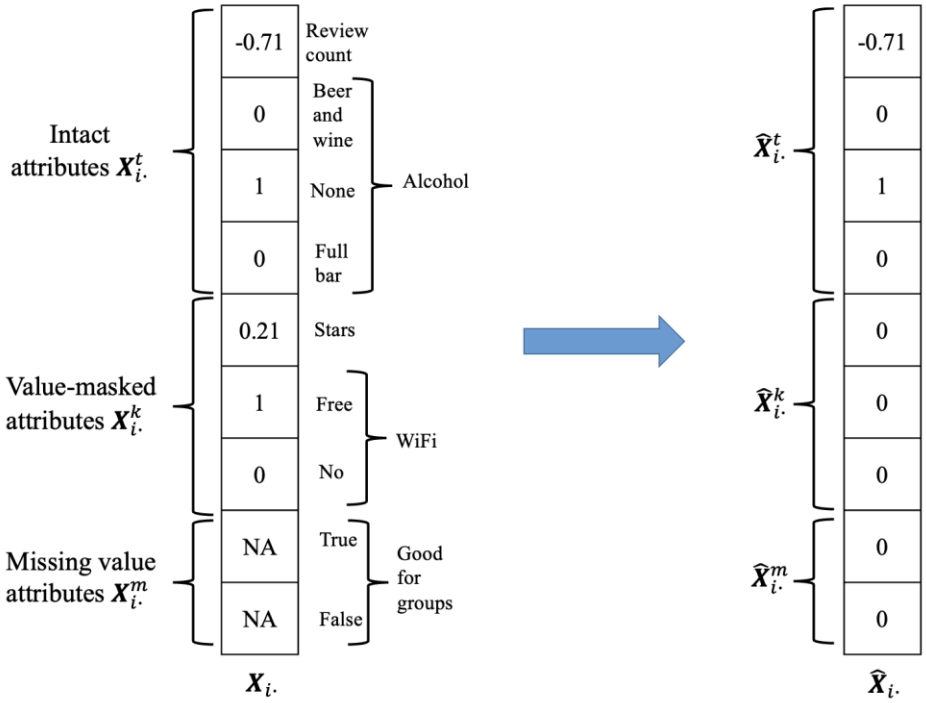

Figure 6: Constructing Corrupted Attribute Vector $\widehat{X}_{i\cdot}$

In Figure 5, encoder1 maps the corrupted attribute vector $\widehat{\boldsymbol{X}}_{i\cdot}$ as follows:

$$\boldsymbol{h}_i^{(1)} = f\left(\boldsymbol{W}^{(1)}\widehat{\boldsymbol{X}}_{i\cdot} + \boldsymbol{b}^{(1)}\right), \text{ and} \tag{4}$$

$$\boldsymbol{h}_i^{(m)} = f\left(\boldsymbol{W}^{(m)}\boldsymbol{h}_i^{(m-1)} + \boldsymbol{b}^{(m)}\right), m = 2,3,\dots,M, \tag{5}$$

where M is the number of hidden layers in the encoder; $\boldsymbol{W}^{(m)}$ and $\boldsymbol{b}^{(m)}$ denote, respectively, the

weight matrix and bias vector of the hidden layer m in the encoder, $m = 1,2,\dots,M$; and $f(\cdot)$ is the rectified linear unit activation function. Encoder2 employs the following equations to map the auxiliary attribute vector $\widetilde{\boldsymbol{X}}_{i\cdot}$:

$$\widetilde{\boldsymbol{h}}_i^{(1)} = f\left(\widetilde{\boldsymbol{W}}^{(1)}\widetilde{\boldsymbol{X}}_{i\cdot} + \widetilde{\boldsymbol{b}}^{(1)}\right), \text{ and} \tag{6}$$

$$\widetilde{\boldsymbol{h}}_i^{(m)} = f\left(\widetilde{\boldsymbol{W}}^{(m)}\widetilde{\boldsymbol{h}}_i^{(m-1)} + \widetilde{\boldsymbol{b}}^{(m)}\right), m = 2,3,\dots,M, \tag{7}$$

where $\widetilde{\boldsymbol{W}}^{(m)}$ and $\widetilde{\boldsymbol{b}}^{(m)}$ denote, respectively, the weight matrix and bias vector of the hidden layer m in encoder2, $m = 1,2,\dots,M$. By Equations (4)–(7), encoders 1 and 2 map $\widehat{\boldsymbol{X}}_{i\cdot}$ and $\widetilde{\boldsymbol{X}}_{i\cdot}$ to their corresponding representation $\boldsymbol{h}_i^{(M)}$ and $\widetilde{\boldsymbol{h}}_i^{(M)}$. These two representations are then concatenated to form vector $\boldsymbol{z}'_i$, which serves as the input to the decoder.

The decoder has M layers: $M-1$ hidden layers and an output layer. The first layer of the decoder, or the $(M+1)$th layer in the imputation component, takes $\boldsymbol{z}'_i$ as its input to produce representation $\boldsymbol{h}_i^{(M+1)}$, fed into the next hidden layer to produce the next representation, as specified in Equations (8) and (9):

$$\boldsymbol{h}_i^{(M+1)} = f\left(\boldsymbol{W}^{(M+1)}\boldsymbol{z}'_i + \boldsymbol{b}^{(M+1)}\right), \text{ and} \tag{8}$$

$$\boldsymbol{h}_i^{(M+m)} = f\left(\boldsymbol{W}^{(M+m)}\boldsymbol{h}_i^{(M+m-1)} + \boldsymbol{b}^{(M+m)}\right), m = 2,3,\dots,M-1. \tag{9}$$

The output layer takes the representation $\boldsymbol{h}_i^{(2M-1)}$ as input and produces the output vector $\boldsymbol{X}'_{i\cdot}$:

$$\boldsymbol{X}'_{i\cdot} = g(\boldsymbol{W}^{(2M)}\boldsymbol{h}_i^{(2M-1)} + \boldsymbol{b}^{(2M)}). \tag{10}$$

In Equation (10), the element-wise function $g(\cdot)$ is an identity function for a numerical attribute or a softmax function for a categorical attribute:

$$g(x) = \begin{cases} x, \text{if } x \text{ is numerical,} \\ \text{softmax}(x), \text{if } x \text{ is categorical.} \end{cases} \tag{11}$$

Again, in Equations (8)–(10), $\boldsymbol{W}^{(M+m)}$ and $\boldsymbol{b}^{(M+m)}$ denote the weight matrix and bias vector,

respectively, of layer m in the decoder, $m = 1,2, \dots, M$.

Similar to $\boldsymbol{X}_{i\cdot}$, attributes in $\boldsymbol{X'}_{i\cdot}$ are segmented into three groups, each corresponding to its counterpart in $\boldsymbol{X}_{i\cdot}$. Specifically, $\boldsymbol{X'}_{i\cdot}^{m}$ contains the imputed values for attributes $\boldsymbol{X}_{i\cdot}^{m}$; $\boldsymbol{X'}_{i\cdot}^{t}$ and $\boldsymbol{X'}_{i\cdot}^{k}$ recover the values of intact attributes $\boldsymbol{X}_{i\cdot}^{t}$ and value-masked attributes $\boldsymbol{X}_{i\cdot}^{k}$, respectively. The learning objective is to minimize the recovery loss such that $\boldsymbol{X'}_{i\cdot}^{m}$ can be an effective imputation of $\boldsymbol{X}_{i\cdot}^{m}$. In turn, the loss function for business b_i is defined as:

$$L(b_i) = \gamma L\left(\boldsymbol{X}_{i\cdot}^{k}, \boldsymbol{X'}_{i\cdot}^{k}\right) + L(\boldsymbol{X}_{i\cdot}^{t}, \boldsymbol{X'}_{i\cdot}^{t}). \tag{12}$$

Recall that $\boldsymbol{X'}_{i\cdot}^{k}$ is derived from $\widehat{\boldsymbol{X}}_{i\cdot}^{k}$ through the encoder–decoder structure of the imputation component. Obtaining $\boldsymbol{X'}_{i\cdot}^{k}$ from $\widehat{\boldsymbol{X}}_{i\cdot}^{k}$ ($\widehat{\boldsymbol{X}}_{i\cdot}^{k} = \boldsymbol{0}$) emulates the process of imputing the values of $\boldsymbol{X}_{i\cdot}^{k}$, and the loss $L\left(\boldsymbol{X}_{i\cdot}^{k}, \boldsymbol{X'}_{i\cdot}^{k}\right)$ is essentially an imputation loss. While $\boldsymbol{X}_{i\cdot}^{t}$ is observed input to the component, $\boldsymbol{X'}_{i\cdot}^{t}$ can be considered a reconstruction of $\boldsymbol{X}_{i\cdot}^{t}$. The loss $L(\boldsymbol{X}_{i\cdot}^{t}, \boldsymbol{X'}_{i\cdot}^{t})$ is a reconstruction loss that complements the imputation loss and improves the efficiency of model learning. Thus, no loss is calculated for missing attributes $\boldsymbol{X}_{i\cdot}^{m}$. The hyperparameter γ controls the relative weight between the imputation loss and the reconstruction loss. In the implementation, we set $\gamma > 1$, so the imputation component learns how to impute missing values instead of reconstructing observed values, as is usually performed by a conventional denoising autoencoder [78].

Both $L\left(\boldsymbol{X}_{i\cdot}^{k}, \boldsymbol{X'}_{i\cdot}^{k}\right)$ and $L(\boldsymbol{X}_{i\cdot}^{t}, \boldsymbol{X'}_{i\cdot}^{t})$ are calculated as the sums of their respective component attribute losses. Taking loss $L\left(\boldsymbol{X}_{i\cdot}^{k}, \boldsymbol{X'}_{i\cdot}^{k}\right)$ as an example,

$$L\left(\boldsymbol{X}_{i\cdot}^{k}, \boldsymbol{X'}_{i\cdot}^{k}\right) = \sum_{j=1,}^{J} l\left(X_{ij}^{k}, X'^{k}_{ij}\right), \tag{13}$$

where $X_{ij}^{k} \in \boldsymbol{X}_{i\cdot}^{k}$ is an attribute in $\boldsymbol{X}_{i\cdot}^{k}$, $X'^{k}_{ij} \in \boldsymbol{X'}_{i\cdot}^{k}$ is X_{ij}^{k}'s counterpart in $\boldsymbol{X'}_{i\cdot}^{k}$, and J denotes the number of attributes in $\boldsymbol{X}_{i\cdot}^{k}$. Attribute loss $l\left(X_{ij}^{k}, X'^{k}_{ij}\right)$ measures the difference between X_{ij}^{k} and X'^{k}_{ij}. It is calculated using the squared-error loss for numerical attributes or the cross-entropy loss for

categorical attributes [25]. Therefore, we have

$$l\left(X_{ij}^{k}, X'^{k}_{ij}\right) = \begin{cases} (X_{ij}^{k} - X'^{k}_{ij})^2, \text{if } X_{ij}^{k} \text{ is numerical,} \\ -\sum_{p=1}^{P_j} X_{ij}^{k\,(p)} \log\left(X'^{k\,(p)}_{ij}\right), \text{if } X_{ij}^{k} \text{ is categorical.} \end{cases} \quad (14)$$

In Equation (14), P_j refers to the number of possible categories for X_{ij}^{k}, $X_{ij}^{k\,(p)}$ is an element of the one-hot encoding vector converted from X_{ij}^{k}, and $X'^{k\,(p)}_{ij}$ denotes the probability that X_{ij}^{k} belongs to category p. With loss $L(b_i)$ for business b_i defined, the total loss is calculated as

$$L = \sum_{i=1}^{N} L(b_i), \quad (15)$$

where N is the number of businesses. By minimizing the total loss L, our method learns its parameters, including the weight matrix $\boldsymbol{W}^{A}$ of the focal-neighbor attention mechanism and the weight matrices and bias vectors of the imputation component, and then it produces the output vector $\boldsymbol{X'}_{i\cdot}$ that contains imputed values for b_i's missing values, $i = 1,2,\ldots,N$.

The proposed method advances previous research in three important ways. First, as conceptualized by institutional theory, our method leverages inter-business relationships for missing value imputations. It operationalizes these inter-business relationships by constructing an auxiliary attribute matrix that it employs to impute missing values through the imputation component. Second, the proposed method considers both inter-business and inter-attribute relationships by integrating two encoders in the imputation component, one for the corrupted attribute vector and another for the auxiliary attribute vector. Third, our method emulates missing value imputation through the encoder–decoder structure. Unlike a conventional denoising autoencoder, the proposed method imputes the missing values (i.e., corrupted values) instead of reconstructing the inputs to learn robust representations, as Figure 5 shows. This process is achieved with an innovative loss function that incorporates both imputation and reconstruction

losses (Equation (12)), rather than relying only on the reconstruction loss commonly employed by a denoising autoencoder for data reconstruction.

Data and Evaluation Design

To demonstrate its imputation efficacy and practical utilities, we use real-world crowdsourced data to evaluate the proposed method, in comparison with several prevalent methods. In this section, we describe the data set, evaluation design, and benchmark methods.

Data Set

We obtained from Yelp a crowdsourced data set that consists of 175,000 businesses in 11 metropolitan areas, prompting 5.26 million reviews and ratings by 1.3 million users.[7] Yelp maintains a set of business categories (e.g., restaurants, arts & entertainment), and business attributes vary across categories. To ensure comparable attributes among businesses, we focus only on the restaurant category—the largest one in the Yelp data set. Our evaluation involves 72,065 restaurants, noted by 960,262 users in 3,635,685 reviews. Each business can be described by 85 attributes: 4 numerical (e.g., number of reviews) and 81 categorical (e.g., Good for Groups). Numerical attributes are factual (e.g., number of reviews, ratings, location in latitude and longitude), whereas most categorical attributes provided by users are perceptual or opinion-based. Yelp uses these business attributes to support recommender system and business searches. Appendix A presents some examples across different categories, reflective of Yelp's actual business search engine. On average, a business has missing values for 64.1% of its attributes. We also constructed a graph with each business as a vertex. If two vertices (i.e., restaurants) receive reviews from same users, they are linked by an edge, the weight of which reflects the number of common customers. Table 1 presents descriptive statistics of the business attribute data and

[7] https://www.yelp.com/dataset.

business graphs.

Number of businesses	72,065
Number of attributes	85 (4 numerical, 81 categorical)
Average missing rate	64.1%

Number of vertices	72,065
Number of edges	43,531,648
Maximum edge weight	916
Average edge weight	1.771

Table 1: Descriptive Statistics of Business Attribute Data and Business Graphs

Evaluation Design and Performance Metrics

To examine the proposed method's imputation effectiveness, we randomly removed a specific ratio of observed values (e.g., 10%, 20%, 30%) for each variable to create missing values in the business attribute data, then used the removed values as holdout values for testing. We repeated our evaluation and examined different removal ratios for the observed values, ranging from 10% to 50% in increments of 10%. We tracked all holdout values and used them as the ground truth to evaluate the respective methods' imputation performance. Next, we applied each method to the remaining data and examined imputation effectiveness by comparing its imputed values and the holdout values, measured by commonly adopted metrics. In line with Oba et al. [59] and Stekhoven [71], we used the normalized root mean square error (*NRMSE*) to assess the imputation error for numerical attributes. Unlike the regular root mean square error (RMSE), *NRMSE* divides variable variances, such that variables with larger variances are less likely to dominate the average imputation error across all attributes, which should reduce bias in the evaluation results. For a numerical attribute *j*, let $\boldsymbol{H}_j$ be its holdout values and $\boldsymbol{I}_j$ represent corresponding values imputed by a method. We computed each method's imputation error for numerical attribute *j* ($NRMSE_j$) as

$$NRMSE_j = \sqrt{\frac{\sum_{k=1}^{z}(H_{jk}-I_{jk})^2/z}{var(\boldsymbol{H}_j)}}, \tag{16}$$

where H_{jk} and I_{jk} represent a holdout value and its corresponding imputed value, respectively; z is the number of holdout values; and $var(\boldsymbol{H}_j)$ denotes the variance in the holdout values. Each

method's imputation error for all the numerical attributes is calculated as the average of $NRMSE_j$ across different numerical attributes. Moreover, we used the proportion of falsely classified entries (*PFC*) to measure the imputation error for categorical attributes [71], defined as the proportion of holdout values misclassified by a method. To calculate each method's *PFC*, we took the average *PFC* across different categorical attributes. To illustrate the practical utilities and value of our method, we also compared the business recommendations empowered by the attribute values imputed by the respective methods, as detailed in the next section.

Benchmark Methods

We identified prevalent methods that represent different methodological categories and included them as benchmarks in the evaluation. For model–based methods, we selected MICE [3] and MIDAS [42], which are multiple imputation methods using chained equations and denoising autoencoder, respectively. Both methods have been frequently applied to deal with incomplete data [7, 80]. Five representation learning–based methods also were included as benchmarks: SoftImpute [54], MLPImpute [69], deep autoencoder (AE) [8], denoising autoencoder (DAE) [78], and missSOM [64]. SoftImpute runs a soft threshold SVD iteratively; it has been commonly used to complete data matrixes [12]. Meanwhile, MLPImpute uses multiple fully connected layer to represent input data and imputes missing values by mapping the data back into the original feature space. As a deep learning–based method, AE employs a deep autoencoder to learn low-dimensional representations of incomplete business attribute data for missing value imputations. We also included DAE, a variant of AE, as a benchmark in the evaluation. In addition, missSOM extends the Kohonen algorithm to compute self-organizing maps; it involves a nonlinear data projection and imputes missing values iteratively. Finally, *k*-nearest neighbor imputation (KNNImpute) [74] and ClustImpute [61], both similarity–based methods, provided two additional

benchmark methods. To impute an attribute of a business with a missing value, KNNImpute identifies *k* other businesses that are most similar to the focal business and have an observed value for that attribute. Then, it uses the weighted average of these observed values to impute the missing value. ClustImpute applies k-means clustering to identify similar businesses and imputes missing values based on similar businesses assigned to the same cluster. Appendix B lists all the benchmark methods used in the evaluation. Moreover, we analyzed each method's computational processing requirement using Big O. Appendix C presents the respective methods' computational complexity; as shown, our method incurs comparable computational processing and can impute missing in a similar, efficient way.

Important parameters of each method were set to ensure its best performance, according to a series of parameter-tuning analyses. We set the maximum iteration number for MICE to 100 and set the maximum iteration number for SoftImpute to 100, based on the parameter-tuning results. For MIDAS, we adopted a 6-layer denoising autoencoder: input-128-96-96-128-output. That is, there are four hidden layers between the input and output layers, and the number of neurons is determined for each hidden layer (e.g., 128 neurons in the first hidden layer). We also followed Lall and Robinson [42] to apply exponential linear unit as the activation function and utilize both root mean squared error and cross-entropy loss for model training. For MLPImpute, we followed previous research [69] to construct multiple hidden layers with the same hidden size and used hyperbolic tangent activation function for each hidden layer. According to the parameter tuning analyses, we chose three hidden layers with a hidden size of 48. For AE, we followed the parameter-tuning analysis results and adopted a 7-layer deep autoencoder with a symmetric structure: input-96-64-32-64-96-output. To avoid overfitting, we applied dropout to each hidden layer with a keep probability of 0.6. The DAE features the same model structure as AE and applies

dropout to the input layer with a keep probability of 0.8. For missSOM, we constructed a map space comprised of neurons that are arranged as a 50 x 50 hexagonal to produce best imputation performances. Guided by the parameter tuning results, we set k to 100 for KNNImpute and number of clusters to 100 for ClustImpute. For the proposed method, we set the random walk restart probability θ to 0.8 and the number K of neighboring businesses visited by the walk to 400. We employed a 4-layer structure (input-96-64-32) for both encoders 1 and 2 and adopted a 3-layer structure (64-96-output) for the decoder. Dropout was applied to each hidden layer with a keep probability of 0.9. With the parameter tuning results, we masked 20% of each business's observed values to train the imputation method, set parameter γ in Equation (12) to 7, and trained our method using the Adam optimizer [39].

Evaluation Results

Imputation Effectiveness

Following the evaluation procedure, we randomly removed 30% of the observed values to create missing values in the business attribute data, and we used the removed (i.e., holdout) values as the ground truth in the testing. We then applied each method to impute the removed values. To examine the performance of a method, we compared the imputed values with the ground truth to calculate its *NRMSE* and *PFC*. For numerical attributes, we performed the evaluation procedure 10 times and obtained each method's average *NRMSE*. In general, a lower *NRMSE* reflects a smaller imputation error. Table 2 presents each method's *NRMSE* and indicates the *NRMSE* reduction by the proposed method over each benchmark. As shown, the *NRMSE* reduction by our method over KNNImpute is 44.5%: (.968 – .537) / .968.

Method	NRMSE	Improvement in NRMSE by Our Method (%)	PFC	Improvement in PFC by Our Method (%)
MICE	0.832	35.5	0.152	9.9

MIDAS	0.880	39.0	0.154	11.0
SoftImpute	0.843	36.3	0.151	9.3
MLPImputation	0.956	43.8	0.164	16.5
AE	0.955	43.8	0.167	18.0
DAE	0.949	43.4	0.160	14.4
missSOM	0.861	37.6	0.163	16.0
KNNImpute	0.968	44.5	0.191	28.3
ClustImpute	1.017	47.2	0.202	32.2
Our Method	**0.537**		**0.137**	

Table 2. Imputation Effectiveness of Proposed Method in Comparison with Benchmark Methods

The proposed method substantially outperforms all the benchmark methods in both *NRMSE* and *PFC*. Compared with the benchmarks, our method reduces the imputation error by 35.5% to 47.2% for numerical attributes, and by 9.3% to 32.2% for categorical attributes. In Appendix D, we demonstrate that the proposed method's imputations for numerical attributes are closer to actual values than those by the best-performing benchmark (MICE). The paired t-test results show that our method significantly outperforms each benchmark in both *NRMSE* and *PFC* ($p < .001$). Its superior performance stems from two methodological novelties: the consideration of inter-business relationships and the emulation of missing value imputation in its encoder–decoder component. Among the benchmark methods, KNNImpute and ClustImpute have the worst performances; they cannot effectively assess the similarities among businesses when there are many missing values in the business attribute data.

To ensure the robustness of our evaluation results, we considered different removal ratios, ranging from 10% to 50% in increments of 10%. For each removal ratio, we performed the evaluation procedure 10 times. As we present in Table 3, the proposed method consistently and substantially outperforms all the benchmarks in both *NRMSE* and *PFC*, across these different removal ratios. For numerical attributes, the improvement in *NRMSE* relative to the best-performing benchmark ranges between 33.6% and 37.6%. For categorical attributes, our method

attains an 8.2%–9.7% improvement in *PFC* over the best-performing benchmark. The paired t-test results again show that our method significantly outperforms each benchmark in *NRMSE* and *PFC* across different removal ratios ($p < .001$). For all the investigated methods, imputation errors appear to increase with the removal ratio, due to the diminished data available for model training as more observed values get removed. The robustness analysis results confirm the significant imputation improvements of our method, relative to all the benchmarks. This proposed method enhances crowdsourced data on an OBD platform by accurately imputing missing attribute values for businesses, which could enhance the effectiveness of business searches, because businesses with more complete attribute information are more likely to be matched with users' specified search terms and conditions than otherwise. Furthermore, it helps users identify businesses that promise to meet their needs, wants, and preferences more effectively.

Method	10%		20%		30%		40%		50%	
	NRMSE	PFC	NRMSE	PFC	NRMSE	PFC	NRMSE	PFC	NRMSE	PFC
MICE	0.779	0.145	0.806	0.148	0.832	0.152	0.855	0.156	0.888	0.162
MIDAS	0.853	0.149	0.866	0.151	0.880	0.154	0.897	0.157	0.909	0.161
SoftImpute	0.800	0.145	0.821	0.147	0.843	0.151	0.863	0.154	0.884	0.158
MLPImputation	0.947	0.157	0.951	0.161	0.956	0.164	0.959	0.168	0.965	0.174
AE	0.947	0.158	0.950	0.163	0.955	0.167	0.963	0.169	0.966	0.180
DAE	0.942	0.153	0.944	0.156	0.949	0.160	0.954	0.163	0.957	0.165
missSOM	0.794	0.155	0.826	0.159	0.861	0.163	0.893	0.168	0.924	0.174
KNNImpute	0.958	0.191	0.953	0.197	0.968	0.191	0.979	0.187	0.985	0.187
ClustImpute	0.983	0.193	1.001	0.198	1.017	0.202	1.039	0.206	1.045	0.209
Our Method	**0.517**	**0.131**	**0.521**	**0.134**	**0.537**	**0.137**	**0.544**	**0.141**	**0.552**	**0.145**

Table 3: Imputation Effectiveness across Different Removal Ratios

Further Analysis of the Proposed Method's Imputation Performance

We scrutinized the proposed method's performance improvement by separating its consideration of inter-business relationships and emulation of missing value imputation. To analyze the marginal contribution of the respective methodological novelties, we dropped them from the proposed

method, one at a time. First, if we exclude inter-business relationships, we can test a method, EMULATE, that only emulates missing value imputations. Second, by excluding the emulation of missing value imputation, we derive a method, BUS-REL, that features only consideration of inter-business relationships. With both novelties removed, the proposed method reduces to AE.

Table 4 presents the average performance of the respective methods across 10 evaluative trials, with a removal ratio of 30%. As shown, AE has the largest imputation errors for both numerical and categorical attributes. By incorporating the emulation of missing value imputation in EMULATE, the imputation errors decrease substantially. The imputation emulating process forces the model to perform imputation during training, which facilitates effective learning of the attribute relationships and better imputes missing attribute values using observed ones, thereby decreasing imputation errors for both numerical and categorical attributes. For BUS-REL, the results show that the consideration of inter-business relationships is more important for numerical attributes, though it still can reduce the imputation errors for categorical attributes. Inter-businesses relationships are critical for imputing missing attribute values, and auxiliary attribute data provide additional information to enhance imputation performance, especially for numerical attributes. The inclusion of both methodological novelties further lowers imputation errors for both numerical and categorical attributes, suggesting their combined contribution is greater than each individual contribution. Paired t-tests confirm the statistical significance of the improvements by each novelty ($p < .001$). Moreover, we analyzed the proposed method's imputation performance in relation to the number of neighboring businesses. As Appendix E illustrates, our method achieves greater imputation effectiveness for businesses, regardless of the number of neighboring businesses.

Method	NRMSE	Improvement over AE in NRMSE (%)	PFC	Improvement over AE in PFC (%)
AE	0.955		0.167	
EMULATE	0.755	21.0	0.141	15.6

BUS-REL	0.602	37.0	0.146	12.6
Our Method	**0.537**	**43.8**	**0.137**	**18.0**

Table 4. Analysis of the Contribution of Each Methodological Novelty in Proposed Method

Use of Imputed Data to Enhance Platform's Business Recommendations

To demonstrate the practical utilities and value of our method, we examined the effectiveness of business recommendations empowered by imputed business attribute data versus those enabled by data imputed by each benchmark method. We employed neural matrix factorization [32], a widely applied recommendation model, which fuses generalized matrix factorization and multi-layer perceptron to derive recommendations. In line with Hu and Ester [34], we assume a user's review of a business results from a visit; that is, each review indicates a visit. We followed He et al. [32] and employed users' last visits (i.e., last visit of each user) as testing data and their remaining visits, together with the imputed business attribute data, as training data. The recommendation model also leverages business attribute data in the training process, as summarized in Panel A of Table 1. These business attribute data have an average missing rate of 64.1%. The recommendation model takes users' visit data, except their last visit, and the business attribute data imputed by each method as inputs, then generates a list of recommended businesses for each user as its output.

We evaluated the effectiveness of these recommendations against the testing data, using the hit ratio (*HR*) and discounted cumulative gain (*DCG*) [31, 82]. In general, *HR* indicates the proportion of user-visited businesses in the testing data that are correctly included in the recommended lists [31]. Let B^v be the set of businesses visited by users in the testing data; in addition, $b_j^v \in B^v, j = 1,2, \dots, |B^v|$, denotes a visited business in B^v. For each b_j^v, R_j represents its corresponding recommendation list. We determine if b_j^v is included in R_j by

$$I(b_j^v, R_j) = \begin{cases} 1, \text{if } b_j^v \in R_j, \\ 0, \text{otherwise.} \end{cases} \quad (17)$$

Accordingly, *HR* can be computed as follows [31]:

$$HR = \frac{\sum_{j=1}^{|B^v|} I(b_j^v, R_j)}{|B^v|}. \tag{18}$$

The value of *HR* ranges between 0 and 1, where 0 indicates that none of the visited businesses in the testing data are included in their corresponding recommendation lists, and 1 signals that the recommendation lists include all visited businesses. Conceivably, a higher value of *HR* signifies a better recommendation performance.

With *DCG*, we gauge the ranking quality of a recommendation list quantitatively [31]. For each visited business in the testing data b_j^v and its corresponding recommendation list R_j, let $r_j^k \in R_j$ be the kth ranked business in the list. Then we quantify the ranking quality of R_j as [31]:

$$DCG_j = \sum_{k=1}^{|R_j|} \frac{2^{I\left(b_j^v, r_j^k\right)} - 1}{\log_2(k+1)}, \tag{19}$$

where $I\left(b_j^v, r_j^k\right) = 1$ if $b_j^v = r_j^k$, and $I\left(b_j^v, r_j^k\right) = 0$ otherwise. According to Equation (19), if a user-visited business is ranked higher in R_j, DCG_j should be greater, so the ranking quality of R_j is better. We take the average of DCG_j across the recommendation lists. The resulting DCG ranges between 0 and 1, with higher values signifying better ranking quality of the lists.

Consistent with Yelp's practice, we set the size of a recommendation list, or the number of recommended businesses, to 5. Table 5 presents the recommendation performance attained with the complete business attribute data imputed by each method; it shows that our method attains the best performance. The business recommendations enabled by its imputed attribute data noticeably outperform those using the complete data imputed by each benchmark method, showing a 3.5%–8.5% improvement in HR and a 5.9%–10.1% improvement in DCG. Together, these results affirm the proposed method's greater utilities for business recommendations than the benchmarks.

Method	HR	Improvement in HR by Our Method (%)	DCG	Improvement in DCG by Our Method (%)
MICE	0.607	3.5	0.577	5.9
MIDAS	0.597	5.2	0.568	7.6
SoftImpute	0.591	6.3	0.567	7.8
MLPImputation	0.594	5.7	0.562	8.7
AE	0.586	7.2	0.563	8.5
DAE	0.601	4.5	0.568	7.6
missSOM	0.599	4.8	0.575	6.3
KNNImpute	0.589	6.6	0.555	10.1
ClustImpute	0.579	8.5	0.563	8.5
Our Method	**0.628**		**0.611**	

Table 5: Recommendation Performance with Complete Business Attribute Data Imputed by Each Method

Discussion

The proposed method helps address completeness and timeliness constraints in crowdsourced data on OBD platforms, so it can enhance the effectiveness of search and recommendation services by the platform. Our method's advantages stem from several factors. First, it leverages a deep model architecture to learn important relationships among business attributes. Unlike many imputation methods that use relatively simplistic model structures [3, 54, 74], the proposed method's deep model architecture can capture essential relationships among attributes and thereby impute missing business attribute values more effectively. Second, our method recognizes the importance of inter-business relationships for imputing missing attribute values on OBD platforms. We incorporate inter-business relationships, as guided by institutional and social learning theories, and leverage the attribute values of related businesses to impute a focal business's missing attribute values. On an OBD platform, inter-business relationships are important and can provide auxiliary information to better impute missing business attribute values than existing, general methods that do not consider them. Third, our method employs a novel learning strategy to steer its imputations of

missing attribute values by emulating the imputation process. This strategy offers greater imputation effectiveness and prevents overfitting. As our evaluation illustrates, the proposed method is both feasible and superior, and the empirical results confirm that its imputed business attribute data can enhance business searches and recommendations, compared with those produced by several prevalent methods. Effective business profiling, searches, recommendations, and targeted advertising by an OBD platform require complete, accurate, and timely business attribute data. In this sense, the proposed method offers practical values to enhance the platform's services when its business attribute data contain substantial missing values.

This study provides several insights for further research. First, inter-business relationships are important and represent a new perspective for imputing missing values in crowdsourced data. As our study illustrates, incorporating these additional data relationships and dimensionalities can account for social learning among individual businesses to improve imputation efficacy. Inter-business relationships should be recognized as essential considerations when imputing missing values in business attribute data. Further research should include this perspective to extend general imputation methods by adding relationships among data dimensionalities that capture essential underlying characteristics among firms. Second, the innovative learning strategy proposed by our method is critical for imputation research to bridge deep leaning methods and imputation tasks. By applying data corruption and imputation loss, deep learning models can emulate the imputation process and achieve improved imputation performance. Third, OBD platforms' business search and recommendation services could benefit from more complete, timely business attribute data. As we show empirically, the proposed method is capable of alleviating data completeness and timeliness constraints; it represents a viable, effective way to enhance crowdsourced data on OBD platforms. Research into effective ways to leverage crowdsourced data could benefit from our

method to address fundamental data quality and availability challenges, as well as create higher-quality data that support more reliable estimations and accurate predictions with fewer biases.

This study offers several implications for practice as well. First, OBD platforms can consider the proposed method to augment their business attribute data collections for improved business profiling, searches, recommendations, and targeted advertising. These benefits should result in greater user experience and satisfaction, increase business visibility and searchability, and enhance the platform's services to both businesses and users. For example, OBD platforms could apply our method to impute missing business attribute values and display the estimated values on individual businesses' pages with an explicit clarification statement indicating that the attribute values are crowdsourced or estimated (i.e., imputed). Customers then might learn more about each business, which might facilitate their purchase decisions. Supported by the proposed (or a similar) method, OBD platforms also can deliver more precise advertisements to users, make more appropriate business recommendations with respect to users' needs and preferences, and optimize search engines to reflect their requirements more comprehensively, which could generate more revenue for both businesses and the platform. As Jannach and Jugovac [36] indicate, a small improvement in a recommender system's quality can generate millions of dollars in revenue annually for social media firms. More effective business recommendations also help increase businesses' visibility and exposure to relevant consumers. As Fang [18] reports, doubling consumers' exposure to Yelp can increase a new, high-quality independent restaurant's revenue by 8%–20% and improve its survive rate by 7–19 basis points.

Second, our method can balance the rankings of different recommended businesses on an OBD platform by alleviating disadvantages created by substantial missing attribute values. Because OBD platforms rely on crowdsourced data, businesses new to the platform or those with

small customer bases tend to suffer missing attribute values, which seriously restricts their visibility, searchability, and recommendation probability, as well as users' attention and purchases. Addressing the fundamental missing value challenge facilitates a more equitable, fairer approach to identify businesses for the platform to recommend, even if they are subject to substantial missing values in the crowdsourced business attribute data. The appeal of such equity in business ranking mechanisms might attract more new businesses to join the OBD platform. Third, our method supports effective, efficient knowledge acquisition and complements existing data gathering processes. Instead of collecting all business attribute data from users, a platform could apply the proposed method to estimate some of them, which offers additional potential benefits, such as reduced time, monetary cost, and data processing requirements.

Conclusion

Missing values prevail in crowdsourced data and have negative effects for users, businesses, and the platform. We address the data completeness and timeliness constraints. We use institutional theory as an anchor, emphasize inter-attribute and inter-business relationships, and develop a novel, deep learning–based imputation method to enhance crowdsourced data on an OBD platform for improved services. Using Yelp data, we evaluate the proposed method's imputation effectiveness and recommendation performance, in comparison with several prevalent benchmark methods. The comparative results reveal that our method achieves superior efficacy for missing value imputations and can support the platform's business recommendation services more effectively. To make important business attribute values available to users in a timely manner, OBD platforms should apply our method periodically and perform imputations with appropriate time intervals.

This study contributes to extant literature in several ways. First, we address data completeness and timeliness constraints in crowdsourced data and develop a novel method that incorporates

inter-business relationships that have been overlooked by previous research, despite their importance to ODB platforms. Second, the proposed method employs a deep model architecture and adopts an innovative learning strategy to emulate the imputation process for missing value imputations. This method can accommodate different types of variables (e.g., categorical, numerical); it is general and can be extended to various scenarios that feature essential, subtle inter-attribute and inter-business relationships, which would facilitate efforts to leverage data-driven analytics for improved services even more. For example, applications of our method to health care analytics might impute missing values in clinical data to create valuable insights for patients and health care providers [19]. Third, we produce empirical results that evince the proposed method's effectiveness for both missing value imputations and business recommendations. The evaluation includes several out-of-sample testing data sets created by different removal ratios, examines the predictive efficacy of the complete data set imputed by our method versus several prevalent methods for supporting an OBD platform's recommendations, and confirms the importance of the methodological novelties (i.e., inter-business relationships and emulation of missing value imputation).

This study can be extended in several directions. The missing data patterns, attribute values, and variable types might be somewhat specific to the Yelp data set that we use to evaluate imputation effectiveness and recommendation performance. Additional and more diverse data are needed to confirm the robust performance of our method. We adopt a deep model architecture with fully connected layers, which is appropriate for our study. Further research should consider other architectures (e.g., recurrent or convolutional neural networks) that support effective imputations. Along with inter-business relationships, other characteristics and factors (e.g., business category, location) could affect imputation effectiveness. Continued investigations should analyze additional,

essential characteristics of crowdsourced business attribute data to identify other factors that might enhance business attribute value imputations further. For optimal imputation performance, future research also should consider the frequency to perform imputations using the data collected over time. Moreover, this study assumes that the business attribute data are MAR and uses randomly selected observed values as holdouts in the evaluation. Yet some missing value scenarios might be more complex and pertain to MNAR. Therefore, studies should consider different missing mechanisms and assess their impacts on imputation efficacy and downstream applications. Finally, user experience and satisfaction with the platform's services (e.g., business recommendations) are essential and warrant attention. In addition to the quantitative evaluations we conduct, further research could use experimental designs to examine users' decision-making, experience, and satisfaction with the platforms' services explicitly.

References

1. Acia I.; Mohamed W.; and Abderrahim G. Modeling the Structure of Social Networks by Using the Pythagorean Sprial. *Journal of Theoretical and Applied Information Technology*, 97, 6 (2019), 1856-1869
2. Anderson C. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hachette Books, 2006.
3. Azur M.J.; Stuart E.A.; Frangakis C.; and Leaf P.J. Multiple Imputation by Chained Equations: What Is It and How Does It Work? *International Journal of Methods in Psychiatric Research*, 20,1 (2011), 40–49.
4. Baesens B.; Bapna R.; Marsden J.R.; Vanthienen J.; and Zhao J.L. Transformational Issues of Big Data and Analytics in Networked Business. *MIS Quarterly*, 40,4 (2016), 807–818.
5. Bandura A. and Walters R.H. *Social Learning Theory*. Prentice-hall Englewood Cliffs, NJ, 1977.
6. Batini C.; Cappiello C.; Francalanci C.; and Maurino A. Methodologies for Data Quality Assessment and Improvement. *ACM Computing Surveys, (CSUR)*, 41, 3 (2009), 1–52.
7. Beaulieu-Jones B.K.; Lavage D.R.; Snyder J.W.; Moore J.H.; Pendergrass S.A.; and Bauer C.R. Characterizing and Managing Missing Structured Data in Electronic Health Records: Data Analysis. *JMIR Medical Informatics*, 6, 1 (2018), e11.
8. Beaulieu-Jones B.K.; and Moore J.H. Missing Data Imputation in the Electronic Health Record Using Deeply Learned Autoencoders. *Pacific Symposium on Biocomputing*, Vol. 0 (2017), 207–218.
9. Beckert J. Institutional Isomorphism Revisited: Convergence and Divergence in Institutional Change. *Sociological Theory*, 28,2 (2010), 150-166.
10. Bhattacharyya, S.; Banerjee, S.; Bose, I.; and Kankanhalli, A. Temporal Effects of Repeated

Recognition and Lack of Recognition on Online Community Contributions. *Journal of Management Information Systems*, 37, 2 (2020), 536–562.
11. Brás L.P.; and Menezes J.C. Improving Cluster-Based Missing Value Estimation of DNA Microarray Data. *Biomolecular Engineering*, 24, 2 (2007), 273–282.
12. Cao Z.; Wang L.; and De Melo G. Link Prediction via Subgraph Embedding-Based Convex Matrix Completion. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, (2018), 2803–2810.
13. Dacin M.T. Isomorphism in Context: The Power and Prescription of Institutional Norms. *Academy of Management Journal*, 40, 1 (1997), 46-81.
14. Deephouse D.L. Does Isomorphism Legitimate? *Academy of Management Journal*, 39, 4 (1996), 1024-1039.
15. Deng Y.; Chang C.; Ido M.S.; and Long Q. Multiple Imputation for General Missing Data Patterns in the Presence of High-Dimensional Data. *Scientific Reports*, 6, 1 (2016), 1–10.
16. DiMaggio P.J.; and Powell W.W. The Iron Cage Revisited: Institutional Isomorphism and Collective Rationality in Organizational Fields. *American Sociological Review*, (1983), 147-160.
17. Ebrahimi S.; Ghasemaghaei M.; and Benbasat I. The Impact of Trust and Recommendation Quality on Adopting Interactive and Non-Interactive Recommendation Agents: A Meta-Analysis. *Journal of Management Information Systems*, 39, 3 (2022), 733-764.
18. Fang, L. The effects of online review platforms on restaurant revenue, survival rate, consumer learning and welfare. (2019).
19. Fang X.; Gao Y.; and Hu P.J.-H. A Prescriptive Analytics Method for Cost Reduction in Clinical Decision Making. *MIS Quarterly*, 45, 1 (2021), 83-115.
20. Fang X.; and Hu P.J. Top Persuader Prediction for Social Networks. *MIS Quarterly*, 42, 1 (2018), 63-82.
21. Fang, X.; Hu P.J.; Chau, M.; Hu, H.; Yang, Z.; and Liu Sheng, O. A Data-Driven Approach to Measure Web Site Navigability. *Journal of Management Information Systems*, 29, 2 (2012), 173–212.
22. Ghose A.; Ipeirotis P.G.; and Li B. Examining the Impact of Ranking on Consumer Behavior and Search Engine Revenue. *Management Science*, 60, 7 (2014), 1632-1654.
23. Ghoshal A.; Mookerjee V.S.; and Sarkar S. Recommendations and Cross-Selling: Pricing Strategies When Personalizing Firms Cross-Sell. *Journal of Management Information Systems*, 38, 2 (2021), 430-456.
24. Glynn M.A.; Abzug R. Institutionalizing Identity: Symbolic Isomorphism and Organizational Names. *Academy of Management Journal*, 45, 1 (2002), 267-280.
25. Goodfellow I., Bengio Y., Courville A., and Bengio Y., *Deep Learning*. MIT Press, 2016.
26. Gupta N.; and Singh S. Collective Factorization for Relational Data: An Evaluation on the Yelp Datasets. *Technical Report*, (2015), Yelp Dataset Challenge, Round 4.
27. Haunschild P.R. Interorganizational Imitation: The Impact of Interlocks on Corporate Acquisition Activity. *Administrative Science Quarterly*, (1993), 564-592.
28. Haunschild P.R.; and Miner A.S. Modes of Interorganizational Imitation: The Effects of Outcome Salience and Uncertainty. *Administrative Science Quarterly*, (1997), 472-500.
29. Hastie T.; Mazumder R.; Lee J.D.; and Zadeh R. Matrix Completion and Low-Rank SVD via Fast Alternating Least Squares. *Journal of Machine Learning Research*, 16, 1 (2015), 3367–3402.
30. Hastie T.; Tibshirani R.; Sherlock G.; Eisen M.; Brown P.; and Botstein D. Imputing Missing

Data for Gene Expression Arrays. *Stanford University Statistics Department Technical report*, (1999).
31. He J.; Fang X.; Liu H.; and Li X.D. Mobile App Recommendation: An Involvement-Enhanced Approach. *MIS Quarterly*, 43, 3 (2019), 827–849.
32. He X.; Liao L.; Zhang H.; Nie L.; Hu X.; and Chua T.-S. Neural Collaborative Filtering. *Proceedings of the 26th International Conference on World Wide Web*, (2017), 173–182.
33. Hinton G.E.; and Salakhutdinov R.R. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313, 5786 (2006), 504–507.
34. Hu B.; and Ester M. Spatial Topic Modeling in Online Social Media for Location Recommendation. *Proceedings of the 7th ACM Conference on Recommender Systems*, (2013), 25-32.
35. Hu P.J.-H.; Hu H.; and Fang X. Examining the Mediating Roles of Cognitive Load and Performance Outcomes in User Satisfaction with a Website: A Field Quasi-Experiment. *MIS Quarterly*, 41, 3 (2017), 975–987.
36. Jannach, D.; and Jugovac, M. Measuring the Business Value of Recommender Systems. *ACM Transactions on Management Information Systems*, 10, 4 (2019), 1–23.
37. Jin Y.; Lee H.C.B.; Ba S; and Stallaert, J. Winning by Learning? Effect of Knowledge Sharing in Crowdsourcing Contests. *Information Systems Research*, 32, 3 (2021), 836-859.
38. Kim H.; Golub G.H.; and Park H. Missing Value Estimation for DNA Microarray Gene Expression Data: Local Least Squares Imputation. *Bioinformatics*, 21, 2 (2005), 187–198.
39. Kingma D.P.; and Ba J. Adam: A Method for Stochastic Optimization. *ArXiv Preprint ArXiv: 1412.6980*, (2014).
40. Kittur A.; Nickerson J.V.; Bernstein M.; Gerber E.; Shaw A.; Zimmerman J.; Lease M.; and Horton J. The Future of Crowd Work. *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, (2013), 1301–1318.
41. Kondra A.Z.; and Hinings C.R. Organizational Diversity and Change in Institutional Theory. *Organization Studies*, 19, 5 (1998), 743-767.
42. Lall, R.; and Robinson, T. The MIDAS Touch: Accurate and Scalable Missing-Data Imputation with Deep Learning. *Political Analysis*, 30, 2 (2022), 179-196.
43. Levitt B.; and March J.G. Organizational Learning. *Annual Review of Sociology*, 14, 1 (1988), 319-338.
44. Li G.; Wang J.; Zheng Y.; and Franklin M.J. Crowdsourced Data Management: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 28, 9 (2016), 2296–2319.
45. Little R., and Donald B.R. *Statistical Analysis with Missing Data*. John Wiley & Sons, Vol. 793, 2019.
46. Liu Y.; Dillon T.; Yu W.; Rahayu W.; and Mostafa F. Missing Value Imputation for Industrial IoT Sensor Data with Large Gaps. *IEEE Internet of Things Journal*, 7, 8 (2020), 6855–6867.
47. Lu J.; Wu D.; Mao M.; Wang W.; and Zhang G. Recommender System Application Developments: A Survey. *Decision Support Systems*, 74 (2015), 12–32.
48. Lukyanenko R.; and Parsons J. Is Traditional Conceptual Modeling Becoming Obsolete? *Lecture Notes in Computer Science*, 8217, LNCS (2013), 61–73.
49. Lukyanenko R.; Parsons J.; and Wiersma Y.F. The Impact of Conceptual Modeling on Dataset Completeness: A Field Experiment. *Proceedings of the 35th International Conference on Information Systems, ICIS 2014*, (2014).
50. Lukyanenko R.; Parsons J.; and Wiersma Y.F. The IQ of the Crowd: Understanding and Improving Information Quality in Structured User-Generated Content. *Information Systems*

Research, 25, 4 (2014), 669–689.
51. Lukyanenko R.; Parsons J.; Wiersma Y.F.; and Maddah M. Expecting the Unexpected: Effects of Data Collection Design Choices on the Quality of Crowdsourced User-Generated Content. *MIS Quarterly*, 43, 2 (2019), 623–648.
52. March J.G. Exploration and Exploitation in Organizational. *Organization Science*, 2, 1 (1991), 71–87.
53. Marimont R.B.; and Shapiro M.B. Nearest Neighbour Searches and the Curse of Dimensionality. *IMA Journal of Applied Mathematics*, 24, 1 (1979), 59–70.
54. Mazumder R.; Hastie T.; and Tibshirani R. Spectral Regularization Algorithms for Learning Large Incomplete Matrices. *Journal of Machine Learning Research*, 11 (2010), 2287–2322.
55. Meyer J.W.; and Rowan B. Institutionalized Organizations: Formal Structure as Myth and Ceremony. *American Journal of Sociology*, 83, 2 (1977), 340-363.
56. Mizruchi M.S.; and Fein L.C. The Social Construction of Organizational Knowledge: A Study of the Uses of Coercive, Mimetic, and Normative Isomorphism. *Administrative Science Quarterly*, 44, 4 (1999), 653-683.
57. Nikolaeva R. Interorganizational Imitation Heuristics Arising from Cognitive Frames. *Journal of Business Research*, 67, 8 (2014), 1758-1765.
58. Nikulin V. Hybrid Recommender System for Prediction of the Yelp Users Preferences. *Proceedings of the Industrial Conference on Data Mining*, (2014), 85–99.
59. Oba S.; Sato M.A.; Takemasa I.; Monden M.; Matsubara K.I.; and Ishii S. A Bayesian Missing Value Estimation Method for Gene Expression Profile Data. *Bioinformatics*, 19, 16 (2003), 2088–2096.
60. Qian X.; Feng H.; Zhao G.; and Mei T. Personalized Recommendation Combining User Interest and Social Circle. *IEEE Transactions on Knowledge and Data Engineering*, 26, 7 (2014), 1763–1777.
61. Pfaffel, O. Clustimpute: An R Package for K-Means Clustering with Build-In Missing Data Imputation.
62. Raghunathan T.; Lepkowski J.; Van Hoewyk J.; and Solenberger P. A Multivariate Technique for Multiply Imputing Missing Values Using a Sequence of Regression Models. *Survey Methodology*, 27, 1 (2001), 85–96.
63. Reagans R.; and McEvily B. Network Structure and Knowledge Transfer: The Effects of Cohesion and Range. *Administrative Science Quarterly*, 48, 2 (2003), 240–267.
64. Rejeb S.; Duveau C.; and Rebafka T. Self-Organizing Maps for Exploration of Partially Observed Data and Imputation of Missing Values. *arXiv:2202.07963*. (2022).
65. Ren, X.; Yu C.M.; Yu, W.; Yang, S.; Yang, X.; McCann, J.A.; and Philip, S.Y. LoPub: high-dimensional crowdsourced data publication with local differential privacy. *IEEE Transactions on Information Forensics and Security*, 13, 9 (2018), 2151-2166.
66. Rubin D.B. *Multiple Imputation for Nonresponse in Surveys*. Wiley Series in Probability and Statistics, Wiley, 2004.
67. Scot W.; and Meyer W. The Organization of Societal Sectors in Organizational Environments: Ritual and Rationality. In *Organizational Environments*, (1983), Beverly Hills, CA.
68. Scott W.R. The Adolescence of Institutional Theory. *Administrative Science Quarterly*, (1987), 493-511.
69. Silva-Ramírez E.L.; Pino-Mejías R.; and López-Coello M. Single Imputation with Multilayer Perceptron and Multiple Imputation Combining Multilayer Perceptron And K-Nearest Neighbours for Monotone Patterns. *Applied Soft Computing*, 29 (2015), 65-74.

70. Srivastava N.; Hinton G.; Krizhevsky A.; Sutskever I.; and Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1 (2014), 1929–1958.
71. Stekhoven D.J. MissForest: Nonparametric Missing Value Imputation Using Random Forest. *Astrophysics Source Code Library*, (2015), ascl-1505.
72. Tang Q.; Gu B.; and Whinston A.B. Content Contribution for Revenue Sharing and Reputation in Social Media: A Dynamic Structural Model. *Journal of Management Information Systems*, 29, 2 (2012), 41-76.
73. Teo H.H.; Wei K.K.; and Benbasat I. Predicting Intention to Adopt Interorganizational Linkages: An Institutional Perspective. *MIS Quarterly*, 27, 1 (2003), 19–49.
74. Troyanskaya O.; Cantor M.; Sherlock G.; Brown P.; Hastie T.; Tibshirani R.; Botstein D.; and Altman R.B. Missing Value Estimation Methods for DNA Microarrays. *Bioinformatics*, 17, 6 (2001), 520–525.
75. Tsai C.-F.; Li M.-L.; and Lin W.-C. A Class Center Based Approach for Missing Value Imputation. *Knowledge-Based Systems*, 151 (2018), 124–135.
76. Van Buuren S.; and Groothuis-Oudshoorn K. Mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, (2010), 1–68.
77. Vaswani A.; Shazeer N.; Parmar N.; Uszkoreit J.; Jones L.; Gomez A.N.; Kaiser Ł.; and Polosukhin I. Attention Is All You Need. *Advances in Neural Information Processing Systems*, Vol. 30 (2017), 5998–6008.
78. Vincent P.; Larochelle H.; Lajoie I.; Bengio Y.; Manzagol P.A.; and Bottou L. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*, 11, 12 (2010).
79. Wang R.Y.; and Strong D.M. Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems*, 12, 4 (1996), 5-33.
80. Wells B.J.; Chagin K.M.; Nowacki A.S.; and Kattan M.W. Strategies for Handling Missing Data in Electronic Health Record Derived Data. *EGEMS*, 1, 3 (2013), 1035. 1–7.
81. Xu D.; Hu P.J.-H.; Huang T.S.; Fang X.; and Hsu C.C. A Deep Learning–Based, Unsupervised Method to Impute Missing Values in Electronic Health Records for Improved Patient Management. *Journal of Biomedical Informatics*, 111. 103576 (2020).
82. Yang C.; Bai L.; Zhang C.; Yuan Q.; and Han J. Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2017), 1245–1254.
83. Yang M.; and Hyland M. Who Do Firms Imitate? A Multilevel Approach to Examining Sources of Imitation in the Choice of Mergers and Acquisitions. *Journal of Management*, 32, 3 (2006), 381-399.
84. Yin J.; Luo J.; and Brown S.A. Learning from Crowdsourced Multi-Labeling: A Variational Bayesian Approach. *Information Systems Research*, 32, 3 (2021), 752-773.
85. Zhang S.; Zhang J.; Zhu X.; Qin Y.; and Zhang C. Missing Value Imputation Based on Data Clustering. In *Transactions on Computational Science I*, Berlin, Heidelberg: Springer-Verlag, (2008), 128–138.
86. Zhu H.; Chen E.; Xiong H.; Yu K.; Cao H.; and Tian J. Mining Mobile User Preferences for Personalized Context-Aware Recommendation. *ACM Transactions on Intelligent Systems and Technology*, 5, 4 (2014), 1–27.

A Deep Learning-based Imputation Method to Enhance Crowdsourced Data on Online Business Directory Platforms for Improved Services

Online Appendix A: Business Attributes Available on Yelp

We summarize the attributes according to different categories that appear in Yelp's search engine. As Figure A1 shows, users can search for businesses by selecting attributes important to their interests and preferences.

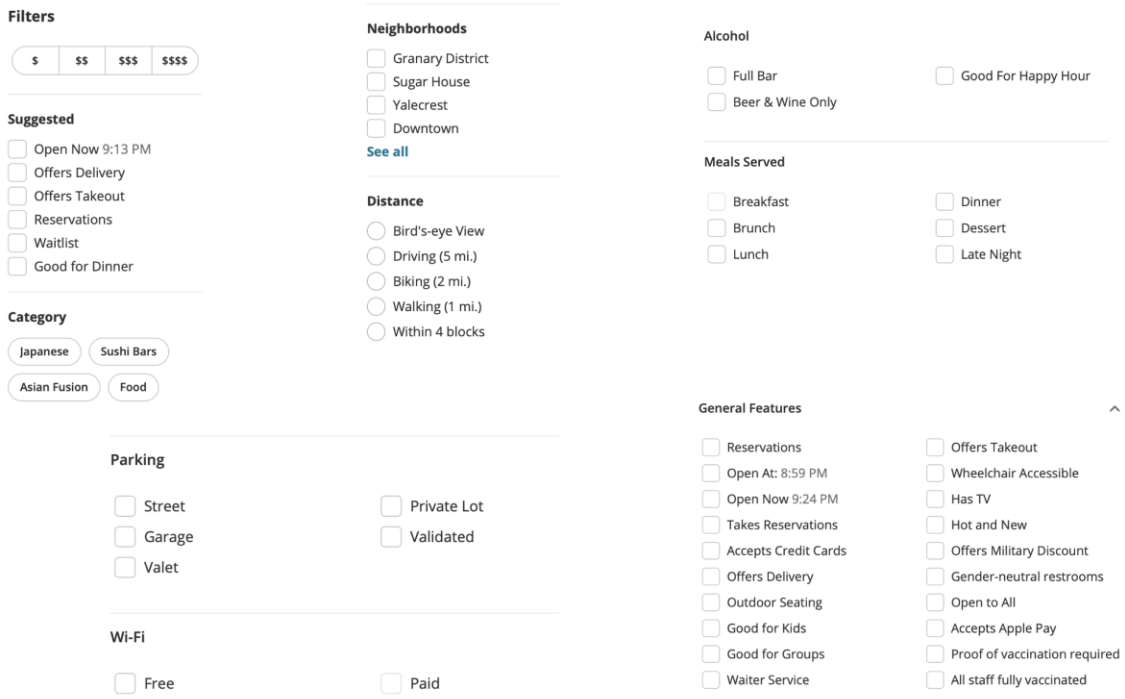

Figure A1: Examples of Business Attributes Used by Yelp's Search Engine

Online Appendix B: Summary of Benchmark Methods

Method	Category	Description
MICE	Model–based	Imputes missing values by iteratively regressing one attribute on all other attributes [2, 3].
MIDAS	Hybrid (model– and representation learning– based)	Imputes missing values using multiple imputation and denoising autoencoder [6].
SoftImpute	Representation learning–based	Represents and imputes business attribute data using SVD [7].
MLPImputation	Representation learning–based	Represents and imputes business attribute data using MLP [10].
AE	Representation learning–based	Represents and imputes business attribute data using deep autoencoder [4].
DAE	Representation learning–based	Represents and imputes business attribute data using denoising autoencoder [12].
missSOM	Representation learning–based	Represents and imputes business attribute data using Kohonen algorithm [9].
KNNImpute	Similarity–based	Imputes missing values with observed values from most similar businesses [11].
ClustImpute	Similarity–based	Imputes missing values using k-means clustering algorithm [8].

Online Appendix C: Computational Complexity Analysis Using Big O

To assess each method's computational processing requirement, we used Big O to analyze its computational complexity in the worst-case scenario [2, 7]. Consider a data set with m attributes and n businesses. SoftImpute performs truncated SVD iteratively over the data. It incurs a computational cost of $O(|\Omega|r + (n + m)r^2)$ in each iteration, where $|\Omega|$ is the total number of observed instances, and r denotes the rank of SVD performed in the iteration. Typically, r is much smaller than $min\{n, m\}$, and $|\Omega|$ is not greater than $n \times m$. The computational complexity of each iteration is less than $O(nm^2)$. That is, the total computational cost of SoftImpute with i iterations is $O(nm^2i)$. Similarly, MICE performs iterative imputations. In each iteration, it imputes the missing values for one attribute with a computation cost of $O(nm^2)$, which results in a cost of $O(nm^3)$ for all attributes. The total computational cost of MICE approximates $O(nm^3i)$ if it performs the imputation i times. Also, missSOM has a computational complexity comparable to that of standard Kohonen algorithm [9]; the complexity is $O(mh + ph^2)i)$, where h is the total number of neurons, p is the dimension of each neuron, and i is the number of iterations [5]. Meanwhile, KNNImpute applies a distance function to calculate similarities among businesses. The computational cost of similarity calculation is approximately $O(n^2m)$, which represents the computational complexity of KNNImpute. Due to its use of the k-means clustering algorithm to impute missing values, ClustImpute has a time complexity of $O(n^2)$, identical to that of the k-means clustering algorithm [1]. Moreover, MIDAS, MLPImputation, AE, and DAE are neural network–based and perform matrix multiplication on each fully connected layer. Their overall computational cost is approximately $O(nmz^{(1)} + n\sum_{k=1}^{K} z^{(k)}\, z^{(k+1)})$, where K indicates the total number of hidden layers, and $z^{(k)}$ is the number of hidden nodes on the kth hidden layer. The first layer of MIDAS, MLPImputation, AE or DAE is the widest and requires the greatest

computational processing. The total complexity is less than $O(nmz^{(1)}Ki)$, where i is the number of epochs for model training. Like AE and DAE, the proposed method's total complexity is $O(nmz^{(1)}Ki)$. We summarize the respective methods' computational complexity in Table C1. As shown, our method has a comparable computational processing requirement and is able to impute missing values efficiently.

Method	Computational Complexity
MICE	$O(nm^2i)$
MIDAS	$O(nmz^{(1)}Ki)$
SoftImpute	$O(\ \lvert\Omega\rvert r\ +\ (n\ +\ m)r^2\ )$
MLPImputation	$O(nmz^{(1)}Ki)$
AE	$O(nmz^{(1)}Ki)$
DAE	$O(nmz^{(1)}Ki)$
missSOM	$O(mh\ +\ ph^2)i)$
KNNImpute	$O(n^2m)$
ClustImpute	$O(n^2)$
Our Method	$\boldsymbol{O(nmz^{(1)}Ki)}$

Table C1: Analysis of Each Method's Computational Complexity

Online Appendix D: Imputation Performance for Numerical Attributes

We demonstrate the proposed method's imputation performance for numerical attributes by plotting its imputations versus those by the best-performing benchmark (MICE), relative to the actual values. We randomly select 100 imputed values for each numerical attribute across different latitudes and longitudes of restaurant location, number of reviews, and review rating. As Figure D1 illustrates, our method's imputations (red dots) are closer to the actual values (blue dots) than those by MICE (yellow dots), which reveals its greater performance.

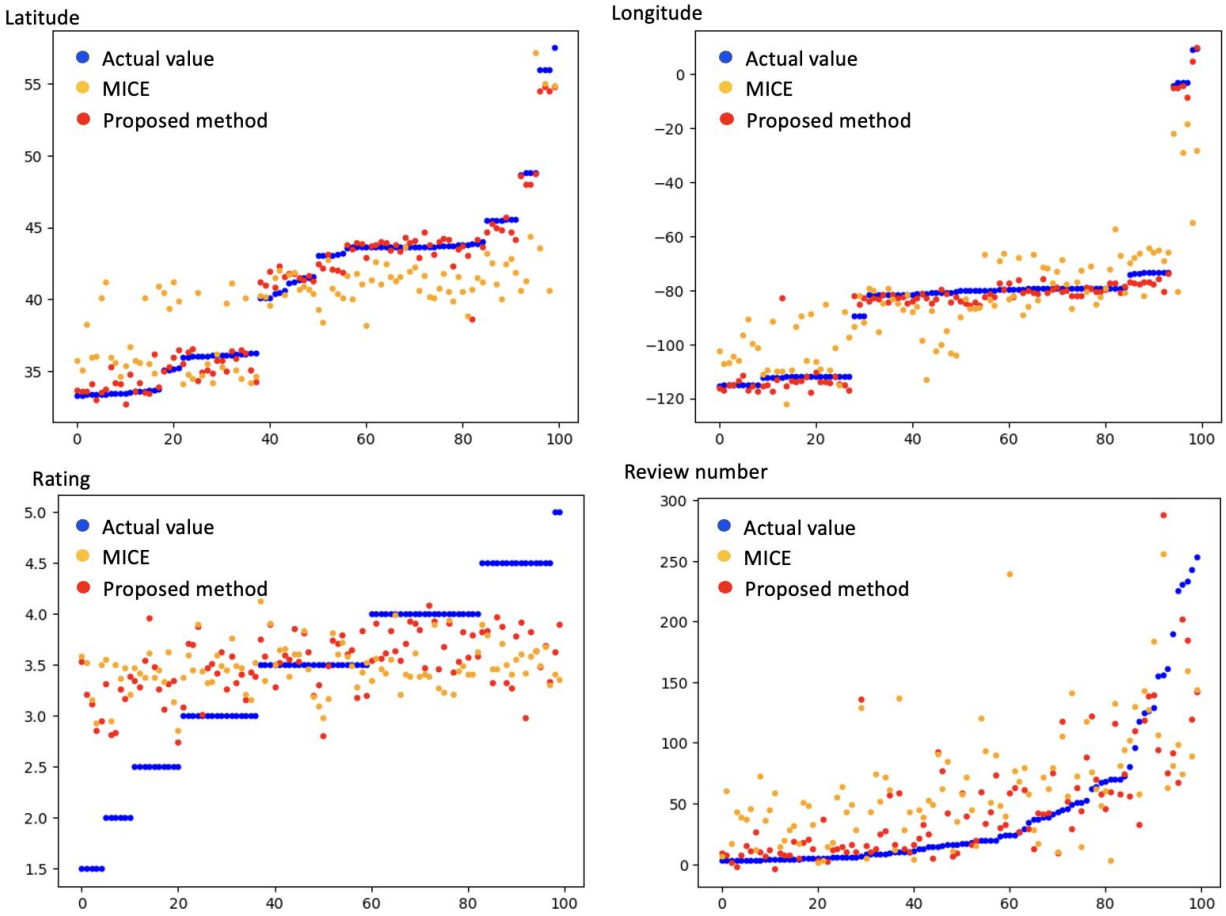

Figure D1: Imputations for Numerical Attributes Relative to Best-Performing Benchmark

Online Appendix E: Analysis of Imputation Performance in Relation to the Number of Neighboring Businesses

We analyzed the proposed method's imputation performance in relation to the number of neighboring businesses, by dividing the businesses into five quantile groups according to their number of neighboring businesses (i.e., 0.2, 0.4, 0.6, 0.8, 1.0). As Figure E1 shows, businesses with very few (0.2) or very many (1.0) neighboring businesses achieve relatively higher NRMSE on numerical attributes than those in the middle quantiles (0.4–0.8). The performance on categorical attributes appears consistent across different numbers of neighboring businesses. As Table 6 indicates, numerical attributes greatly benefit from inter-businesses relationships and also are more sensitive to the number of neighboring businesses. Similar to KNN, our method tends to overfit for businesses with few neighbors and underfit for those with many neighbors, leading to the U-shaped curve for numerical attributes. Overall, the proposed method offers greater imputation effectiveness for businesses, regardless of the number of neighboring businesses.

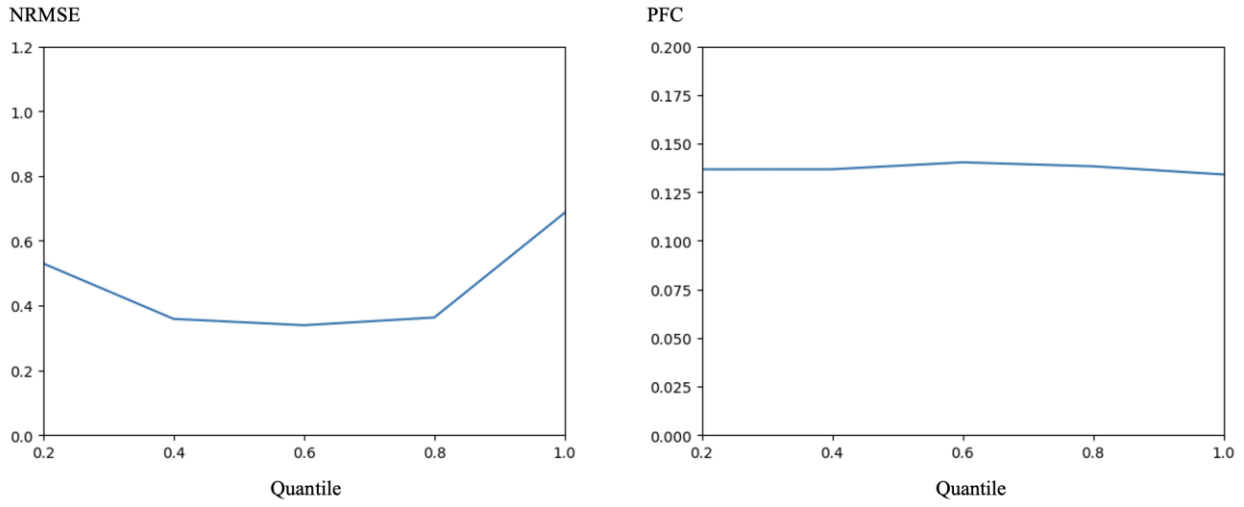

Figure E1: Imputation Performance in Relation to the Number of Neighboring Businesses

References in Appendices

1. Ahmed, M.; Seraj, R; and Islam, S.M.S. The k-means Algorithm: A Comprehensive Survey and Performance Evaluation. *Electronics*, 9, 8 (2020), 1295.
2. Azur M.J.; Stuart E.A.; Frangakis C.; and Leaf P.J. Multiple Imputation by Chained Equations: What Is It and How Does It Work? *International Journal of Methods in Psychiatric Research*, 20,1 (2011), 40–49.
3. Beaulieu-Jones B.K.; Lavage D.R.; Snyder J.W.; Moore J.H.; Pendergrass S.A.; and Bauer C.R. Characterizing and Managing Missing Structured Data in Electronic Health Records: Data Analysis. *JMIR Medical Informatics*, 6, 1 (2018), e11.
4. Beaulieu-Jones B.K.; and Moore J.H. Missing Data Imputation in the Electronic Health Record Using Deeply Learned Autoencoders. *Pacific Symposium on Biocomputing*, Vol. 0 (2017), 207–218.
5. Besson L. Self-Organizing Maps and DSOM From Unsupervised Clustering Algorithms to Models of Cortical Plasticity. (2016).
6. Lall, R.; and Robinson, T. The MIDAS Touch: Accurate and Scalable Missing-Data Imputation with Deep Learning. *Political Analysis*, 30, 2 (2022), 179-196.
7. Mazumder R.; Hastie T.; and Tibshirani R. Spectral Regularization Algorithms for Learning Large Incomplete Matrices. *Journal of Machine Learning Research*, 11 (2010), 2287–2322.
8. Pfaffel, O. Clustimpute: An R Package for K-Means Clustering with Build-In Missing Data Imputation.
9. Rejeb S.; Duveau C.; and Rebafka T. Self-Organizing Maps for Exploration of Partially Observed Data and Imputation of Missing Values. *arXiv:2202.07963*. (2022).
10. Silva-Ramírez E.L.; Pino-Mejías R.; and López-Coello M. Single Imputation with Multilayer Perceptron and Multiple Imputation Combining Multilayer Perceptron And K-Nearest Neighbours for Monotone Patterns. *Applied Soft Computing*, 29 (2015), 65-74.
11. Troyanskaya O.; Cantor M.; Sherlock G.; Brown P.; Hastie T.; Tibshirani R.; Botstein D.; and Altman R.B. Missing Value Estimation Methods for DNA Microarrays. *Bioinformatics*, 17, 6 (2001), 520–525.
12. Vincent P.; Larochelle H.; Lajoie I.; Bengio Y.; Manzagol P.A.; and Bottou L. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*, 11, 12 (2010).