UrbanKG: An Urban Knowledge Graph System

YU LIU and JINGTAO DING, Tsinghua University
YANJIE FU, University of Central Florida
YONG LI, Tsinghua University

60

Every day, our living city produces a tremendous amount of spatial-temporal data, involved with multiple sources from the individual scale to the city scale. Undoubtedly, such massive urban data can be explored for a better city and better life, as what the urban computing community has been dedicating in recent years. Nevertheless, existing studies are still facing the challenges of data fusion for the urban data as well as the knowledge distillation for specific applications. Moreover, there is a lack of full-featured and user-friendly platforms for both researchers and developers in the urban computing scenario. Therefore, in this article, we present UrbanKG, an urban knowledge graph system to incorporate a knowledge graph with urban computing. Specifically, the system introduces a complete scheme to construct a knowledge graph for urban data fusion. Built upon the data layer, the system further develops the multiple layers of construction, storage, algorithm, operation, and applications, which achieve knowledge distillation and support various functions to the users. We perform representative use cases and demonstrate the system capability of boosting performance in various downstream applications, indicating a promising research direction for knowledge-driven urban computing.

CCS Concepts: • **Information systems** → **Spatial-temporal systems**; • **Computing methodologies** → **Knowledge representation and reasoning;**

Additional Key Words and Phrases: Urban computing, knowledge graph, intelligent system

ACM Reference format:
Yu Liu, Jingtao Ding, Yanjie Fu, and Yong Li. 2023. UrbanKG: An Urban Knowledge Graph System. *ACM Trans. Intell. Syst. Technol.* 14, 4, Article 60 (May 2023), 25 pages.
https://doi.org/10.1145/3588577

1 INTRODUCTION

In the past few years, advanced sensing technologies and ubiquitous data sensors have harvested massive multi-source spatial-temporal data from urban spaces,[1] which greatly promote urban

[1]In following sections, we refer to the massive multi-source spatial-temporal data from urban space as the urban data for brevity.

This work was supported in part by the National Key Research and Development Program of China under grant 2020AAA0106000, the National Nature Science Foundation of China under grants 61972223, 61971267, and U1936217, and the Guoqiang Institute of Tsinghua University under grant 2021GQG1005.
Authors' addresses: Y. Liu, J. Ding, and Y. Li (corresponding author), BNRist, Department of Electronic Engineering, Tsinghua University, Beijing, China; emails: liuyu2419@126.com, dingjt15@tsinghua.org.cn, liyong07@tsinghua.edu.cn; Y. Fu, Department of Computer Science, University of Central Florida; email: yanjie.fu@ucf.edu.

2157-6904/2023/05-ART60 $15.00
https://doi.org/10.1145/3588577

computing research [75]. Specifically, the urban data have been explored for various applications, such as cellular data for trajectory prediction [12], ride hailing data for origin-destination flow prediction [63], traffic data for flow prediction [54], check-in data for location recommendation [32] and route recommendation [58, 59, 65], sensor data for air quality measurement [69], image data for socioeconomic prediction [14, 36], and health data for epidemic prediction [56, 57].

Moreover, several recent studies identify the importance and necessity of fusing urban data for specific applications—for example, **Point of Interest (POI)** data and taxi data are directly combined for crime prediction [51], and facility data and road network data are jointly considered for facility planning [68]. However, most existing studies only explore no more than three sources of data, and directly concatenate different sources of data as input, which fail to provide an effective data fusion way with complete urban data considered. Meanwhile, other artificial intelligence domains like natural language processing, computer vision, and recommender systems have been exploring ways to distill explicit knowledge hidden in massive data for performance enhancement and semantic reasoning [15, 24, 71–73], which thus enlightens the path to recent knowledge-driven research in urban computing [33, 34, 37, 46]. However, existing studies fail to explicitly extract the urban knowledge from the urban data for downstream tasks. Hence, we summarize the key challenges of leveraging urban data for further urban computing research in the following two perspectives:

- *Data fusion*: Due to the diverse sources and tools of data collection, the collected urban data are in different structures as well as modalities. On one hand, the urban data are usually stored in different structures like tables, sequences, and graphs—for example, the object property data in table structure, the human trajectory data in sequence structure, and the spatial data in graph structure. On the other hand, the urban data are naturally represented in different modalities like images and texts—for example, the street view images and the text descriptions of urban elements. Therefore, how to fuse the urban data in different structures as well as modalities for complete information remains underexplored.
- *Knowledge distillation*: The massive and comprehensive knowledge about a city lies in the urban data, and we need to distill useful parts for corresponding applications. Specifically, for a specific urban computing task, the knowledge in most parts of urban data may be not useful and even provide negative impacts. Taking the location recommendation task as an example, the air quality data from long-term collection hardly provide useful knowledge for location visiting behaviors, and may introduce extra noises in model learning and further impair performance. Therefore, how to distill task-specific knowledge from the urban data becomes another critical challenge.[2]

As a matter of fact, the preceding challenges faced by urban computing research are quite common in artificial intelligence applications (i.e., learning to fuse massive multi-source domain data and further distill knowledge therein for downstream tasks). For example, in a question answering system [77], the system is required to fuse as many data as possible for topic coverage and then distill useful knowledge to match an input query with the appropriate answer. Additionally, the recommender system [73] also faces similar requirements, which should fuse massive user behavior data as well as item data from multiple sources to satisfy diverse demands, then concentrate the useful part of knowledge to capture users' preferences accurately [48]. Especially, we learn lessons from these areas and find that a successful attempt for such challenges is the **Knowledge Graph**

[2]Here the knowledge distillation describes the process of extracting useful and concentrated information from full urban data for specific tasks.

(KG) [19, 61]—for example, the KG-based recommender system [15] and the KG-based question answering system [20].

Specifically, the KG stores and represents real-world knowledge with triple facts in the form of (*head entity, relation, tail entity*), where entities are objects, events, situations, or abstract concepts and relations that describe their connections [42]. Moreover, the success of Wikipedia and the advanced information extraction methods in recent years have encouraged the emergence of several large-scale KGs like Freebase [5], DBpedia [25], Wikidata [49], and YAGO [11], which are further leveraged in recommender systems [15], question answering systems [20], and natural language processing [2] for data fusion as well as knowledge distillation. Additionally, recent studies have proposed temporal KG [7] and geographic KG [9, 45, 62] for temporal information enrichment and geographic information management, respectively. Especially, the geographic KG stores and describes geographic knowledge like the height and the coordinates of a mountain instead of elements in a city. Thus, existing studies mainly focus on language knowledge or geographic knowledge while largely ignoring the knowledge in the urban data, failing to apply for practical urban computing research, which thus is still an open issue to be addressed.

To overcome the preceding challenges and unleash the knowledge power of the urban data for urban computing, in this article we propose UrbanKG, an urban knowledge graph system to fuse the urban data and distill the urban knowledge therein. The overall system first builds the data layer to collect and clean the urban data from multiple sources in urban space, then builds multiple layers to achieve data fusion and knowledge distillation. Built upon the data layer, the multiple layers of construction, storage, algorithm, operation, and applications can be easily developed for various requirements in urban computing research. Representative evaluations and use cases are provided to validate the effectiveness of the proposed UrbanKG system.

The main contributions of our work are as follows:

- We build an urban knowledge graph system, which constructs UrbanKG from the massive multi-source spatial-temporal data for data fusion, further develops various KG representation algorithms, and combines basic operations for urban computing applications with knowledge distillation achieved. To the best of our knowledge, UrbanKG is the first KG-based system for the urban scenario that offers a novel insight into urban computing research.
- We present a systematic scheme for UrbanKG construction, which identifies key elements like user, region, and POI in an urban environment as entities, and describes their semantic connections in spatiality, property, affiliation, and so on as relations. The proposed construction scheme provides a general framework to fuse the urban data into the KG and potentially benefits various downstream tasks in the urban scenario.
- We abstract several basic operations from the UrbanKG system, which can be further combined for practical applications. Moreover, representative use cases from individual-level, population-level, and city-level aspects are investigated for effectiveness demonstration.

Moreover, in a preliminary version of this work [35], we proposed the simple version of the UrbanKG system. Here, we build on this prior effort by presenting a much more comprehensive investigation: we introduce the multi-modal entities and cross-modal relations in the construction layer. Moreover, detailed information of the UrbanKG system across multiple layers is provided for building the system. Two more use cases are further developed to evaluate the potential of the UrbanKG system for practical applications.

The rest of the article is organized as follows. In Section 2, we introduce the background of KG and list the requirements derived from typical system implementations that are used to guide our UrbanKG system design. In Section 3, we first present an overview of our proposed UrbanKG system, then dig deeply into the details of the system layer by layer. In Section 4, we discuss several typical use cases with the UrbanKG system. Finally, we discuss the related work in Section 5 and conclude the article in Section 6.

2 PRELIMINARIES AND REQUIREMENTS

2.1 Preliminaries

Here, we formally define the KG as follows [19].

Definition 2.1 (Knowledge Graph). A KG is defined as a multi-relational graph structure $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$, where $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{F}$ are the set of entities, relations, and facts, respectively. Especially, the fact set $\mathcal{F} = \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$ stores the triples in the KG, where a triple $(h, r, t) \in \mathcal{F}$ denotes a directional edge from entity h to entity t with relation type r.

Traditional KGs focus on static knowledge without temporal information considered. However, there is tremendous spatial-temporal knowledge in the urban scenario. For example, a triple of $(user\_i, visit, location\_j)$ can only describe the visiting record of $user\_i$ to $location\_j$ while missing the temporal information. Thus, to capture such spatial-temporal knowledge, we formally define the temporal KG as follows [7].

Definition 2.2 (Temporal Knowledge Graph). A temporal KG is defined as a KG with time timestamps $\mathcal{G}^T = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F}^T\}$, where $\mathcal{E}$, $\mathcal{R}$, $\mathcal{T}$, and $\mathcal{F}^T$ are the set of entities, relations, timestamps, and temporal facts, respectively. Especially, the temporal fact set $\mathcal{F}^T = \{(h, r, t, \tau) | h, t \in \mathcal{E}, r \in \mathcal{R}, \tau \in \mathcal{T}\}$ stores the quadruples in a temporal KG, where a quadruple $(h, r, t, \tau) \in \mathcal{F}^T$ denotes a directional edge from entity h to entity t with relation type r and an edge attribute of temporal information τ.

Hence, the preceding visiting record can be represented as $(user\_i, visit, location\_j, time\_k)$ with the visiting time enriched in the temporal KG. Note that the traditional KG can be seen as the special case of the temporal KG with all temporal facts in the same timestamp.

2.2 Requirement

To guide our system design for a full-featured and user-friendly platform in urban scenario, we summarize the requirements as the following five aspects:

- *Construction and storage compatibility*: The urban data are in various structures and modalities, which should be unified in construction and storage.
- *Algorithm universality*: Since the embeddings of entities and relations in a KG support various downstream tasks, the system should implement a universal KG representation algorithm, which can be easily incorporated with other modules.
- *Operation flexibility*: For user-friendly usage as well as diverse development, highly abstracted programmatic as well as declarative (SQL-like) interfaces are necessary [13]—that is, the system should provide flexible operations.
- *Application coverage*: The urban scenario involves various applications across different elements as well as scales, such as the elements of users, POIs and regions, and the scales of individual-level, population-level, and city-level ones. Thus, the system should cover quite comprehensive applications from different aspects.
- *Data freshness*: The urban data are produced all the time, and several use cases require time-sensitive data for application. Therefore, the system should support periodical updates for data freshness.

Hence, we build the UrbanKG system with such requirements in mind, which is introduced in the following.

3 THE URBAN KNOWLEDGE GRAPH SYSTEM

In this section, we first present an overview of our designed UrbanKG system, then introduce the specific details from the layering perspective.

3.1 System Overview

The high-level system architecture of the UrbanKG system is shown in Figure 1. The different components of all layers are described as follows.

Data. This layer supports the data uploading from both internal developers and external users, where the massive multi-source spatial-temporal data from the web, the sensor, and so on are collected and cleaned for data preparation. Especially, this layer periodically updates the data to the upper layers such that the UrbanKG system can absorb more and fresher knowledge over time.

Construction. This layer provides the construction scheme for UrbanKG. It first defines the schema—that is, the high-level structure of KG, including the types of entities and relations. Then various techniques are developed to extract entities as well as relations from the urban data with different structures as well as modalities considered. Furthermore, both entities and relations are enriched with additional attributes matched. In this way, the constructed UrbanKG successfully fuses the urban data together.

Storage. This layer provides the storage interface for UrbanKG. All triples in the constructed UrbanKG are transformed into the RDF data structure, which are then fed into the Virtuoso [10] database for storage and later operations like query. Especially, based on the cluster configuration, the storage interface supports paralleled operation execution for efficiency.

Algorithm. This layer provides various KG representation algorithms for UrbanKG representation. For easy use of UrbanKG in higher-layer applications, the KG representation algorithm converts the discrete triples to continuous representations (i.e., embeddings), which designs a scoring function on embeddings of entities and relations to measure the plausibility of the triple. The learned embeddings provide knowledgeable representations for entities and relations, which successfully distill generalized knowledge in the urban data.

Operation. This layer provides several operations to access the UrbanKG or customize specific functions for the higher-layer applications. The basic operations include KG query via SPARQL and embedding access, as well as three operations corresponding to the common tasks of node classification, link prediction, and graph pooling. Moreover, this operation layer and the data uploading function in the data layer are integrated together as the software development toolkit for developers to build more customized applications with the UrbanKG system.

Application. This layer provides the interfaces for representative applications. Built upon the operation layer, this layer calls various operations or operation combinations to support specific applications, including query-based application, node-based application, link-based application, and graph-based application. Furthermore, the applications are divided into individual level, population level, and city level for use cases in Section 4.

3.2 Data Layer

The data layer provides the functions of data uploading, data collection, and data cleaning. Especially, the urban data are uploaded from various urban scenarios, which are collected from the web service, environment sensors, cellular networks, and and so on. Moreover, the urban data are categorized into the following four aspects.

Spatial Data. The spatial data mainly include objects with spatial/location information in the city, such as the POI with longitude and latitude information denoted as $x^{\mathrm{POI}} = (\mathrm{lng}^{\mathrm{POI}}, \mathrm{lat}^{\mathrm{POI}})$, the business area with closed location curve denoted as $x^{\mathrm{Ba}} = \{(\mathrm{lng}_i^{\mathrm{Ba}}, \mathrm{lat}_i^{\mathrm{Ba}}), \ldots, (\mathrm{lng}_{i+k}^{\mathrm{Ba}}, \mathrm{lat}_{i+k}^{\mathrm{Ba}})\}$, and

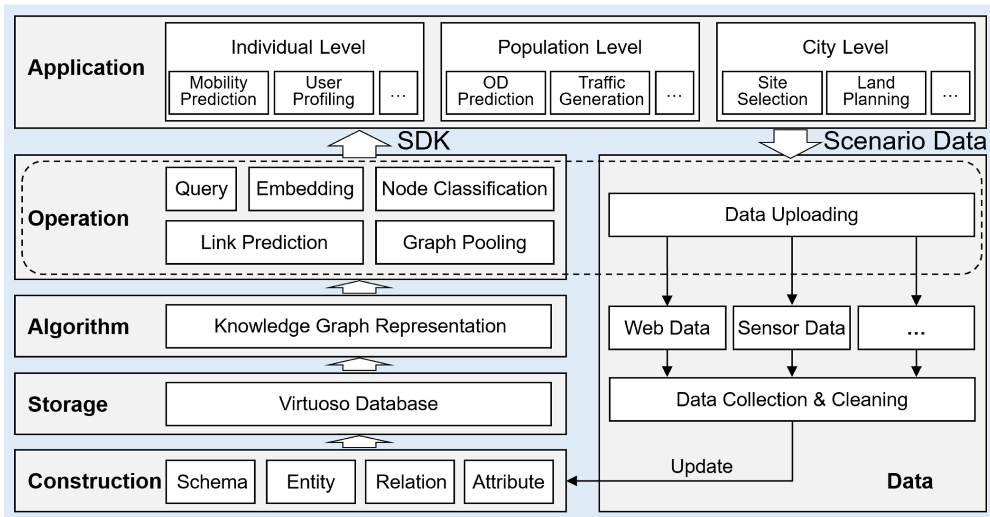


Fig. 1. The high-level system architecture.

the road segment with a sequence of longitude-latitude pairs denoted as $x^{\text{RN}} = \{(\text{lng}_j^{\text{RN}}, \text{lat}_j^{\text{RN}}), \ldots, (\text{lng}_{j+n}^{\text{RN}}, \text{lat}_{j+n}^{\text{RN}}\}$.

User Behavior Data. The user behavior data refer to the user-generated data through various online and offline behaviors, such as the mobility trajectory with a sequence of location points denoted as $tr^u = \{l_1^u, \ldots, l_n^u\}$ with user u and location-timestamp point $l_i^u = (\text{lng}_i^u, \text{lat}_i^u, \tau_i^u)$ and the check-in record at POI p at timestamp τ denoted as (u, p, τ). Additionally, the users generate online behavior data via app interaction, such as clicking or searching certain locations via online map service. Therefore, both the online and offline user behavior data describe the connections and interactions between users and the urban environment.

Attribute Data. The attribute data enrich other data sources with further attribute or property information provided. Such attribute data cover the text description and category information of the POI, the demographic information of the user, and other types of auxiliary information, which are also known as features for objects. The demographic information includes gender, age, income level, education level, and so on.

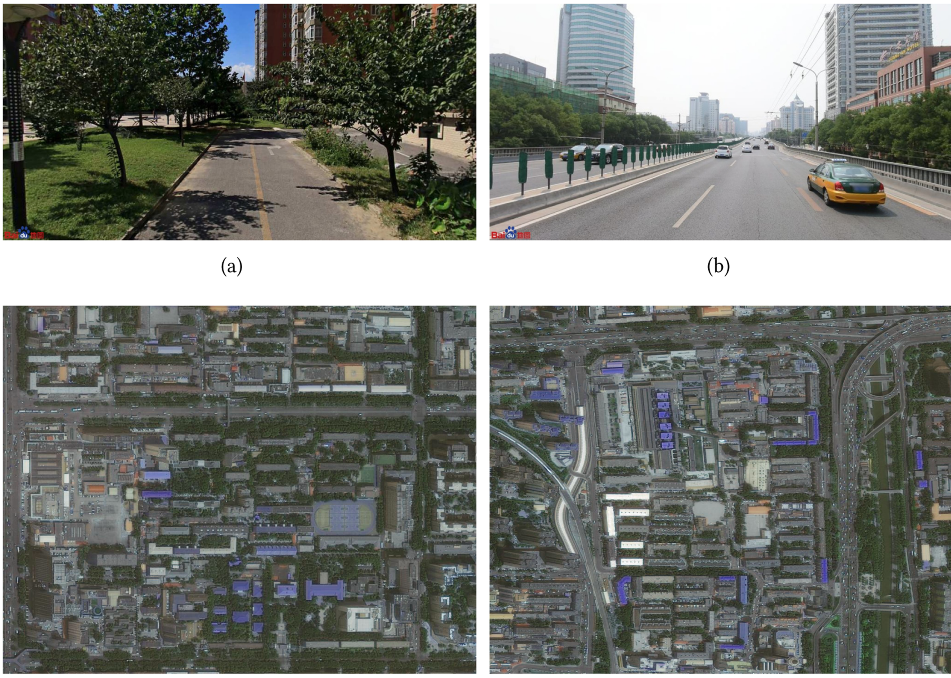
Sensing Data. The sensing data focus on the vision data of street view images and remote sensing images. The street view image[3] provides interactive panoramas from positions along the streets, whereas the remote sensing image is obtained via satellite. In Figure 2, we present four examples of street view images and remote sensing images.

For better understanding, in the following we further present required data sources, formats, and statistics in Table 1, which can help to quickly and easily replicate the UrbanKG construction procedure in various cities.

3.3 Construction Layer

The overall construction scheme is divided into four parts: schema definition, entity identification, relation extraction, and attribute enrichment. We also present basic statistics of constructed UrbanKGs for better comparison.

[3]https://map.baidu.com.



(a) (b)

(c) (d)

Fig. 2. Street view images (a, b) and remote sensing images (c, d) in collected sensing data.

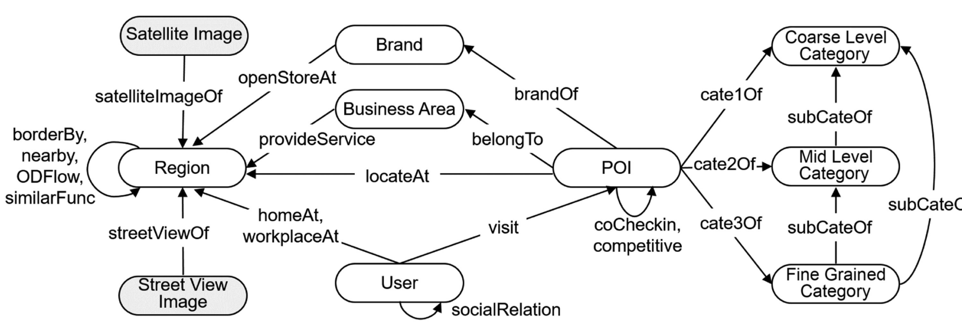


Fig. 3. The schema of UrbanKG. The gray nodes correspond to the cross-modal types of entities.

Schema Definition. The schema or ontology describes the high-level structure of a KG, including the type of entities and relations therein [19]. Especially, following the requirement in Section 2, we consider several goals in the UrbanKG schema definition:

- To capture fundamental entities in the urban environment
- To cover enough relations for entity connection description
- To model spatial-temporal information in the urban data
- To allow for linkage with external KGs.

Figure 3 depicts the overall schema of UrbanKG, where the nodes represent types of entities and the edges describe their relationships in UrbanKG. For easy extension and usage of the UrbanKG

Table 1. Summary of Required Data Sources and Basic Sample Formats for the UrbanKG System

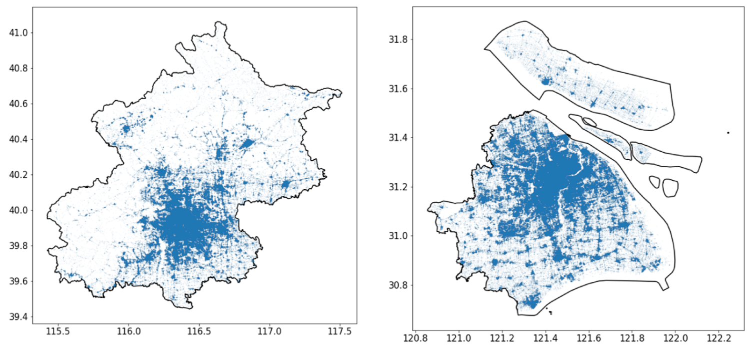
File Name	Basic Sample Format	#Records
road_network.txt	segement_ID | segement_lng-lat_sequence *Seg_1 | [116.5058, 41.1143], . . ., [116.5076, 41.1092]*	2,523 3,479
bussiness_area.txt	Ba_ID | Ba_margin | Ba_name *Ba_1 | [116.3440, 40.0107], . . ., [116.3446, 39.9920] | Wudaokou*	365 342
poi.txt	POI_ID | POI_lng-lat | POI_name | POI_category | POI_address *POI_1 | [116.3325, 40.0017] | Tsinghua University* *| Education | 30 Shuangqing Rd, Haidian*	1.6 million 2.1 million
category.txt	fine_cate_id | fine_cate_name | mid_cate_id | mid_cate_name | coarse_cate_id | coarse_cate_name *CateF_1 | Sichuan Cuisine | CateM_1 | Chinese Food | CateC_1 | Food*	409 522
brand.txt	brand_ID | brand_name *Brand_1 | KFC*	2,001 2,001
image.txt	image_ID | image file	0.4 million 0.7 million
user_trajectory.txt	user_ID | location-timestamp point sequence *User_1 | [116.4950, 39.9800, 1573719753], . . .* *[116.2900, 39.8600, 1572643993]*	90 million 84 million
user_checkin.txt	user_ID | POI_ID | timestamp *User_1 | POI_1382 | 1572644991*	100 million 100 million
user_social.txt	user_ID | user_ID | social_relationship *User_1 | User_2 | Follow*	– –
user_attribute.txt	user_ID | age | gender | education | income | occupation *User_1 | 30-40 | male | undergraduate | medium | managers*	100,000 27,000

An example for each file is presented in italic font. The numbers in the #Records column correspond to the statistics for raw data in Beijing and Shanghai, respectively.

schema, we denote the prefix and namespace of UrbanKG as `ukgs` and http://www.urbankg.org/schema/ such that the entities and relations in UrbanKG can be referred like `ukgs:UKGEntity` nad `ukgs:UKGRelation`, respectively. Then we introduce the types of entities and relations in UrbanKG.

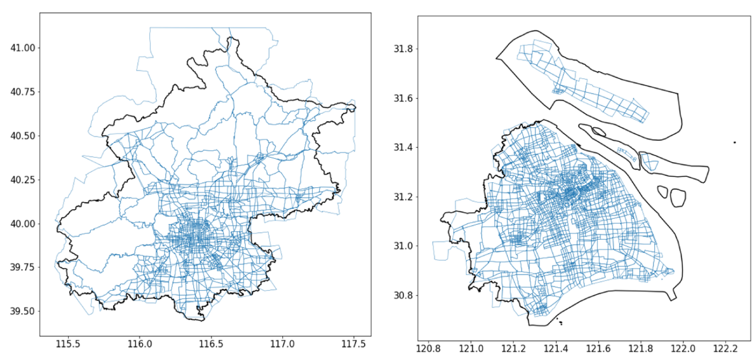
Entity Identification. Based on the observation from the urban data as well as the literature from urban computing [75] and urban planning, the fundamental entities in the urban environment include following types:

- *POIs*: POIs represent the basic functional units and venues in the city, such as schools, hospitals, and markets, which are key spatial points where human activities happen. Figure 4 visualizes the spatial distribution of POI entities in the Beijing and Shanghai datasets.
- *Regions*: Regions are spatial divisions of the city following certain criterion, which can represent basic functional areas in the city. We adopt the spatial divisions partitioned by the road network to keep the road segment information in the UrbanKG. Note that the system can easily adapt to various spatial division schemes with a specific criterion. Figure 5 visualizes the spatial distribution of region entities in the Beijing and Shanghai datasets.
- *Business areas*: Business areas are the commercial and business centers of the city, which usually contain commercial offices like the central business district in the city.
- *Brands*: Brands correspond to the name or symbol of service providers in business, marketing, and advertising, and each brand usually owns several chain stores (e.g., KFC and Pizza Hut, with chain restaurants around the city). Organizations like governmental agencies and public service organizations are also defined as brands.



(a) Beijing (b) Shanghai

Fig. 4. Visualization of POI entities in the Beijing (a) and Shanghai (b) datasets.



(a) Beijing (b) Shanghai

Fig. 5. Visualization of region entities in the Beijing (a) and Shanghai (b) datasets.

- *Categories*: Categories correspond to the property or the function of POIs, such as the food category to a restaurant POI and the shopping category to a shopping mall. Especially, to adapt to various application demands, we adopt the three-level categories in the UrbanKG system (i.e., the coarse-level category, the mid-level category, and the fine-grained category) just like the food category, the Chinese food category, and the Sichuan Cuisine category.
- *Street view images*: Street view images mainly describe urban environment along streets.
- *Satellite images*: Satellite images provide the bird's eye view to each block of the city.
- *Users*: Users refer to human living in the urban environment, which dominate the activities as well as interactions happening in the environment.

Relation Extraction. Based on the identified entity types in UrbanKG, we extract typical relations to describe semantic connections between entities, which are classified as follows:

- *Spatial relations*: Spatial relations model the spatial relationships between urban entities. For example, the `borderBy` relation connects two region entities that share the boundary, whereas the `nearBy` relation connects neighboring region entities. The `locatAt` and `belongTo` relations model the spatial relationships between POI entities and region entities as well as business area entities, respectively.

- *Affiliated relations*: Affiliated relations describe property-based and taxonomy-based relationships between urban entities. The `brandOf` relation identify POIs' affiliated brands. Moreover, the `cate1Of`/`cate2Of`/`cate3Of` relations indicate the three-level categories a POI belongs to, whereas the `subCateOf` relation corresponds to the hierarchical structure across different levels of categories.
- *Functional relations*: Functional relations emphasize the semantic relatedness between urban entities, which are summarized from empirical tries in urban computing research [29, 41]. Specifically, for region entities, the `ODFlow` relation connects region entities with significant origin-destination flow transition—that is, two region entities will be linked by the `ODFlow` relation if their transited flow exceeds a certain threshold. The `similarFunc` relation connects region entities with similar POI distribution—that is, two region entities will be linked by the `similarFunc` relation if the cosine similarity of their POI distribution vectors exceeds a certain threshold. As for POI entities, the `coCheckin` relation describes the concurrent of POIs in check-in records, whereas the `competitive` relation focuses on the competitiveness between POIs—that is, POI entities of the same category within certain distance will be linked by the `competitive` relation [29] Moreover, the `provideService` relation connects business area entities with their nearby region entities, and the `openStoreAt` relation indicates the site opening of brand entities to specific region entities. Such relations are mainly extracted from public data, and more functional relations can be extracted with further input data.
- *Personal relations*: Personal relations focus on the individual knowledge. For instance, the `homeAt` and `workplaceAt` relations identify region entities to a user's home and workplace, whereas the `visit` relation connects the user entities with their visited POI entities. Considering the social network among users, we introduce the `socialRelation` relation to model such social relationships.
- *Cross-modal relations*: Cross-modal relations are utilized to describe cross-modal relationships between urban entities. The `satelliteImageOf` and `streetViewOf` relations connect satellite image entities and street view image entities with their focused region entities. Additionally, the `hasTextDescription` relation can be built between text description and urban entities when corresponding data are available, such as the POI entities with text reviews.

Considering the spatial-temporal characteristic in the urban scenario, we further introduce the temporal KG into UrbanKG. Specifically, for relations with possible timestamps, we add the timestamp attribute to the relational edge. For example, both the `visit` relation and the `ODFlow` relation can be temporally extended for more appropriate semantics. We summarize the relations extracted in the UrbanKG system in Table 2.

Attribute enrichment: To fuse more urban data into UrbanKG, the system further enriches the entities with attribute provided, which are described as follows:

- *POI attributes*: POI attributes include the basic information of name, location, address, category, and activeness indicator like the number of check-ins. Moreover, for POIs about food and shopping such as restaurants, we collect attributes like price, consumer rating, average score, and text review from an online life service platform.
- *Region attributes*: Region attributes include the border range, area, and POI distribution therein. Additionally, we collect demographic information like population from a public website.[4] To further describe land types of regions, we apply PSPNet [74] on regions' satellite images for semantic segmentation of planting, building, river, and road.

[4]https://www.worldpop.org/.

Table 2. Summary of Relations and Corresponding Semantics Captured in UrbanKG

Relation	Head Entity	Tail Entity	Semantics
`borderBy`	Region	Region	Boundary sharing
`nearBy`	Region	Region	Close distance
`locateAt`	POI	Region	Spatial coverage
`belongTo`	POI	Ba	Spatial coverage
`brandOf`	POI	Brand	Affiliation
`cate1Of`	POI	Category	Coarse level
`cate2Of`	POI	Category	Mid level
`cate3Of`	POI	Category	Fine-grained
`subCateOf`	Category	Category	Hierarchy
`ODFlow`	Region	Region	Flow transition
`similarFunc`	Region	Region	POI distribution
`coCheckin`	POI	POI	Check-in concurrence
`competitive`	POI	POI	Competitiveness
`provideService`	Ba	Region	Service support
`openStoreAt`	Brand	Region	Site selection
`homeAt`	User	Region	Home
`workplaceAt`	User	Region	Workplace
`visit`	User	POI	Visiting
`socialRelation`	User	User	Social relationship
`satelliteImageOf`	SI	Region	Cross-modal
`streetViewOf`	SV	Region	Cross-modal

Ba, SI, and SV denote business area, satellite image, and street view image, respectively.

- *Business area attributes*: Business area attributes include the basic information of name, location, and POI distribution therein.
- *Brand attributes*: For brand entities, we match their names with text descriptions in a Chinese encyclopedia-based knowledge base.[5] The brand entities include business-related companies and concepts, as well as governmental institutions and social organizations.
- *Category attributes*: The category attributes include the name and the number of POIs in the corresponding category. Table 3 presents a summary of category names.
- *Image attributes*: To easily adapt cross-modal entities in downstream tasks, for satellite image entities and street view image entities, we use ResNet [18] to extract feature map as the attributes.
- *User attributes*: Table 4 presents user attributes including demographics such as gender, age, education, income, and occupation, which can be enriched with available data.

Hence, guided by the schema, the UrbanKG system identifies entities and extracts relations in the urban scenario, which are further enriched by corresponding attributes. Note that several entities in UrbanKG are also in external KGs, which can be aligned together. For example, the business area entity `Wudaokou` in UrbanKG corresponds to entity `Q1191930`[6] in Wikidata KG [49].

Statistics. Following the preceding creation process, the UrbanKG system constructs two UrbanKGs in the large cities of Beijing and Shanghai in China, whose basic statistics are shown in Table 5. It can be observed that the UrbanKG system contains millions of entities and more than 10

[5] https://baike.baidu.com/.
[6] https://www.wikidata.org/wiki/Q1191930.

Table 3. Name Summary of Complete Coarse-Level Categories and Partial Mid-Level Categories

Coarse Level	Mid Level
Food	Chinese food, Southeast Asian food, Western food, dessert, barbecue
Shopping	Market, mall, bazaar
Leisure sports	KTV, cinema, gym, concert hall
Accommodation	Hotel, chain hotel
Business	Company, finance
Residence	Residential district
Life services	Express, salon, home service, laundry
Transportation	Bus station, airport, subway station
Car services	Parking lot, gas station, driving school
Education	School, university
Medical services	Hospital, clinic, pharmacy
Resort	Homestay, scenic spot
Government	Government organization, scientific institution
Factory	Industrial park, agriculture factory

Table 4. Summary of User Attributes and Classified Groups

Attribute	Classified Group
Gender	Male, female
Age (years)	0–30, 30–40, 40–60, 60–99
Income	Low, lower medium, upper medium, high
Education	Junior high school, senior high school, undergraduate, postgraduate
Occupation	Administration support, healthcare and technicians, managers, professionals, sales workers, services, transport, and production

Table 5. Basic Information of the Constructed UrbanKGs

UrbanKG	Overall Statistics			Entity Types							
	#Entity	#Relation	#Triple	#POI	#Region	#Ba	#Brand	#Category	#SV	#SI	#User
Beijing	1,574,082	22	7,535,876	1,481,100	1,900	333	1,545	14/56/367	78,268	1,900	8,599
Shanghai	1,972,385	22	9,871,283	1,957,674	2,597	280	954	14/56/480	–	2,597	7,733

Ba, SI, and SV denote business area, satellite image, and street view image, respectively. #Category denotes the number of coarse-level/mid-level/fine-grained categories.

million triples on urban knowledge, which is comparable to large-scale KGs like YAGO [11] and WordNet [39]. Two sample datasets of UrbanKG in Beijing and Shanghai are provided in the link shown in the footnote[7] for better understanding and reproducibility.

3.4 Storage Layer

With UrbanKG constructed in the construction layer, the storage layer focuses on the triple storage and query interface. First, for research purposes, the UrbanKG system provides the simple text format for triples, where we maintain one file for each relation that lists the pair of entities involved in such relation. Additionally, triples of the whole UrbanKG are stored in one text file for easy load or transformed to other types of files like RDF, XML, and Turtle.

[7] https://anonymous.4open.science/r/UrbanKG_System_Sample-88E1/.

Moreover, the UrbanKG system adopts the Virtuoso [10] database for storage with business and development purposes. Specifically, the triples in Urban KG are transformed into the standard RDF data structure for serialization (storage and transmission), which are then loaded into the Virtuoso database to support graph-based processing as well as SPARQL query. Based on the Virtuoso database, the UrbanKG system enables parallelization of query execution and can be scaled to multiple clusters for larger KG storage.

3.5 Algorithm Layer

As described before, the triples in UrbanKG are stored in discrete symbols with entities' and relations' ids, which are not suitable for direct use in downstream tasks, especially with deep learning [61]. To overcome such challenges, recent studies propose KG representation [19, 61], which learns low-dimensional continuous representation vectors (a.k.a. embeddings) for entities and relations while preserving the inherent structure and semantics of the KG. Therefore, the KG representation learning process can be viewed as the knowledge distillation from the urban data.

ALGORITHM 1: KG Representation Algorithm.

```
Input: UrbanKG G = (E, R, F);
Init: Initialize entity and relation embeddings E, R with random vectors or attribute vectors.
Init: Initialize parameters of scoring function φ.
for i = 1, 2, ..., n_epoch do
    for (h, r, t) ∈ F do
        Compute the score φ(h, r, t′) for all entities t′ ∈ E;
        Compute L_(h,r,t) with pre-defined loss function;
        Update parameters of embeddings and φ, w.r.t. gradients using ∇L;
    end
end
Output: Entity and relation embeddings E, R.
```

For better understanding, we present the procedure of the KG representation algorithm in Algorithm 1. Given an UrbanKG $\mathcal{G}$, the KG representation algorithm designs various scoring functions ϕ on entity and relation embeddings, so as to calculate higher scores for valid triples than invalid ones. Based on pre-defined loss function $\mathcal{L}$ like cross-entropy loss and hinge loss [40], the KG representation algorithm updates embedding parameters until convergence.

Especially, representative scoring functions for KG representation include translation-based [6], tensor decomposition based [4, 47], and neural network based ones [8]. For example, given a triple (h, r, t) in UrbanKG, the translation-based model TransE [6] designs the scoring function based on the distance assumption that head and tail entities are close via relation-specific operation.

The tensor decomposition models develop various tensor decomposition techniques for scoring function design. For instance, TuckER [4] applies Tucker decomposition to measure the plausibility of a given fact, which is expressed as

$$\phi(h, r, t) = \mathcal{W} \times_1 \boldsymbol{h} \times_2 \boldsymbol{r} \times_3 \boldsymbol{t}, \tag{1}$$

where $\mathcal{W}$ is the core tensor to model the interaction between entities and relations.[8] Additionally, the neural network models design scoring functions via neural networks, which are computationally intensive and prone to overfitting [47]. Therefore, in the UrbanKG system, we mainly implement two other types of KG representation algorithms like TransE [6] and TuckER [4].

[8] $\times_i$ is the tensor product along the i-th mode.

Table 6. Basic Operations in the UrbanKG System

Operation	Return	Description
`ukg_query(SPARQL_Command)`	`return answer`	Executes the SPARQL query on UrbanKG
`ukg_emb(x, alg)`	`return x_emb`	Obtains the embeddings of `x` (entity or relation)
`node_cls(ent, K, alg)`	`return class_score`	Executes the classification task on the entity's embedding given `K` classes
`link_pred(src_ent, tar_ent, rel, alg)`	`return link_score`	Calculates the plausibility score with embeddings given an input triple
`graph_pool(alg, p, ent_1, ..., ent_N)`	`return graph_emb`	Obtains pooling representation of the subgraph on input `N` entities

3.6 Operation Layer

The UrbanKG system supports typical functions in traditional graph systems [67] such as SPARQL and embedding access. Moreover, to easily adapt the system for urban scenario applications, we also develop three types of functions on different levels of UrbanKG, which are abstracted basic operations in Table 6.

`ukg_query`. This operation accepts the SPARQL query from the user, which returns the corresponding results on UrbanKG. For example, the user can obtain the information of a KFC restaurant in the Wudaokou[9] business area via the following query command.

Query Application Example

```
SELECT ?POI.id, ?POI.name, ?POI.lng, ?POI.lat
WHERE ?POI.id ukgs:brandOf wiki:KFC
AND ?POI.id ukgs:belongTo wiki:Q1191930
```

`ukg_emb`. This operation provides the direct interface to access the embedding of the entity or relation in UrbanKG. Especially, the embeddings of various KG representation algorithms can be provided based on the input augment `alg`.

`node_cls`. This operation achieves the node classification task in the KG. By providing the index of corresponding entity `ent` as well as the number of classes `K`, the UrbanKG system calls this operation to do classification based on the embedding learned by the KG representation algorithm `alg`.

`link_pred`. This operation supports the relational link prediction between two entities of `src_ent` and `tar_ent`, which calculates a score via the scoring function in the KG representation algorithm `alg`. Moreover, the user can call this operation for downstream tasks with obtained scores.

`graph_pool`. This operation executes subgraph extraction and pooling on given input entities `ent_1`, ..., `ent_N`. The pooling augment `p` can be either simple max/mean/sum/min pooling functions or user-implemented pooling functions.

Note that the last three operations can be called in both training and evaluation steps in a deep learning task, and the calling at the training step updates the embedding parameters together for better application performance. Furthermore, most of applications in the urban scenario can be

[9]Here, ukgs:Ba_Wudaokou in UrbanKG is aligned with https://www.wikidata.org/wiki/Q1191930 in Wikidata KG.

Table 7. Supported Applications of the UrbanKG System in Different Levels and Types

Level	Application	Type		
		Node Based	**Link Based**	**Graph Based**
Individual	User profiling	✓		
	Mobility prediction		✓	
	Location recommendation		✓	
Population	Traffic generation	✓		
	Flow prediction	✓		
	OD prediction		✓	
City	Land planning	✓		
	Site selection		✓	
	Socioeconomic prediction			✓

achieved by a combination of the preceding basic operations, which will be shown later. Especially, we implement such proposed operations with the interfaces provided by OpenKE[10] and PyKEEN.[11]

3.7 Application Layer

According to the preceding basic operations as well as the ways of using UrbanKG, the applications in the UrbanKG system can be classified into four types:

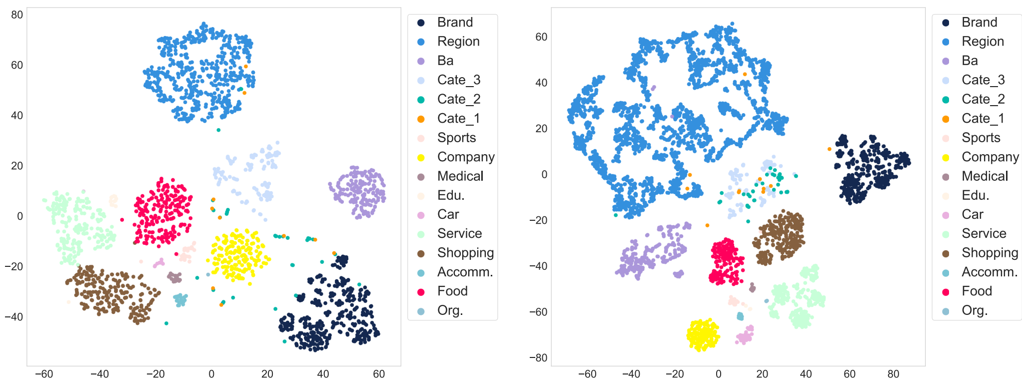
- *Query-based application*: This type of application is a typical application for a KG-based system [9, 11, 49], which provides complex query applications based on the query operation implementation in the operation layer.
- *Node-based application*: This type of application focuses on the nodes/entities in UrbanKG, which leverages entity embedding for classification, regression, and so on. Based on the semantic connections in UrbanKG, such node-based applications can absorb more context knowledge than simply feature input in traditional applications.
- *Link-based application*: This type of application focuses on the edges/links in UrbanKG, which modify scoring functions with application characteristics to explore pairwise relationships between entities.
- *Graph-based application*: This type of application focuses on the subgraph or the whole UrbanKG, which aggregates the information of selected entities as well as relational edges among them to describe the local status for specific application demands.

It is worth mentioning that the preceding applications are supported by the software development toolkit of basic operations introduced before. For a specific application, the user can call various operations as well as operation combinations for the task demands.

Moreover, in Table 7, we list representative applications in respective of ways of using UrbanKG as well as the focus scales. According to the table, we can observe that the UrbanKG system enables various applications across multiple scales, from the micro-scale individual-level scenarios to the macro-scale population-level scenarios, and further to the large-scale city-level scenarios:

- *Individual-level application*: This level of application focuses on a single user. For example, the user profiling problem aims to infer the user demographic information [60] based on user entity embedding, whereas the mobility prediction problem [52] and location

[10] https://github.com/thunlp/OpenKE.
[11] https://github.com/pykeen/pykeen.



(a) Beijing (b) Shanghai

Fig. 6. Entity embedding visualization results for UrbanKG in Beijing and Shanghai.

recommendation problem [32] aim to predict the interaction links between the user entity and other entities.
- *Population-level application*: This level of application focuses on a group of users and their connected regions. For example, the traffic generation problem [31] and flow prediction problem [30] leverage region embeddings for cellular traffic generation and crow flow prediction, respectively, which are mainly node-based applications. Additionally, the OD prediction problem [33, 41] predicts the crowd flow between a pair of region entities in UrbanKG.
- *City-level application*: This level of application focuses on the region and the whole city. For example, the land planning problem [50] can be formulated as the node classification problem on UrbanKG with region entities, whereas the site selection problem [34] is formulated as the link prediction problem between brand entities and region entities on UrbanKG. Moreover, the socioeconomic prediction problem [36, 66, 76] is to predict the indicators like population and GDP of region entities, which requires the neighboring subgraph information for inference, like the connected POI entities as well as satellite image entities.

Therefore, the UrbanKG system supports various applications in the urban scenario, with different ways of using UrbanKG as well as adapted to different scales, which further demonstrates the generalization and flexibility of the system.

4 EVALUATION

In this section, we evaluate the effectiveness and wide applicability of our designed UrbanKG system from embedding analysis and use cases across various applications introduced earlier.

4.1 Embedding Analysis

To validate the effectiveness of the construction layer and the algorithm layer in the UrbanKG system, we adopt the TuckER model [4] to learn entity embeddings on UrbanKG for Beijing.

Specifically, in the experiment, we set the embedding dimensionality to 32 and utilize the early stopping strategy for training—that is, the training stops when the training loss does not decrease for 10 iterations. Due to the large amount of POI entities in UrbanKG, we randomly sample 50,000 POI entities and keep other types of entities for training. In Figure 6, we visualize the learned entity embedding using t-SNE, and 10,000 POI entities are randomly selected for visualization. Note that entities in different types and POI entities in different categories are shown in different colors.

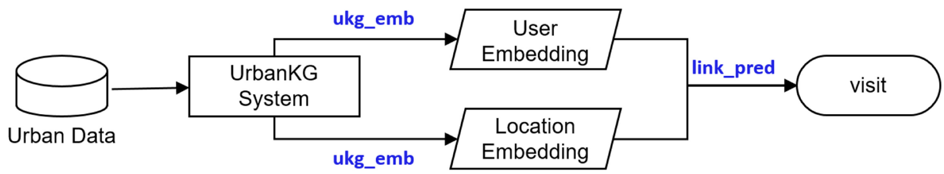


Fig. 7. An illustration of leveraging the UrbanKG system for the mobility prediction problem.

According to the figure, we can observe that entities in different types or categories are clearly separated in space and the clustering phenomenon can easily be found in different groups. The visualization results indicate that learned embeddings preserve the semantics of entities—that is, distill the knowledge in the urban data in some way. Therefore, the construction layer and algorithm layer together provide effective representations for entities, which support the urban applications in the upper layer of the system.

4.2 Use Cases Analysis

In this section, we investigate representative applications in the urban scenario and demonstrate the effectiveness of the UrbanKG system from two aspects:

- The UrbanKG system achieves better performance than traditional solutions in applications.
- The UrbanKG system provides better representations to promote the performance of traditional solutions.

Additionally, the superiority of the UrbanKG system can be shown by providing more explainable results, reflecting the reasoning process, and so on. Specifically, we present use cases across individual-level problems of mobility prediction, user profiling, population-level problems of traffic generation, and city-level problems of site selection.

Baseline and Metrics. First, for each use case, we have chosen several baselines for performance comparison. For use cases like mobility prediction and site selection following the first demonstration aspect, we report the best baseline results for intuitive comparison. For use cases like user profiling and traffic generation following the second demonstration aspect, we report the ablation results to investigate the promotion using the UrbanKG system. Second, we follow the commonly used metrics in corresponding task for performance comparison.

4.2.1 Mobility Prediction. The mobility prediction use case [52] formulates the traditional trajectory prediction problem into the link prediction problem on UrbanKG, which is stated as follows.

Problem 1 (UrbanKG-Based Mobility Prediction Problem). *Given the temporal-based UrbanKG $\mathcal{G}^T = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F}^T\}$, a record in the user trajectory like the user u visits the POI p at timestamp τ can be expressed as $(e_u, r_{\text{visit}}, e_p, \tau)$ with e_u and e_p as entities and r_{visit} as the relation therein. Hence, the mobility prediction problem of predicting the visiting location at query timestamp τ_{query} for user u can be formulated as the link prediction problem of $(e_u, r_{\text{visit}}, ?, \tau_{\text{query}})$ in UrbanKG.*

The overall framework is illustrated in Figure 7. Especially, the application layer calls the operation **`link_pred`** to predict if there exist visiting links between user entities and POI (location) entities with the time attribute considered.

To evaluate the proposed framework, we sample two user mobility datasets of Beijing and Shanghai, whose basic statistics are summarized in Table 8. We split the datasets by 7:1:2 as the train/valid/test datasets. The first 70% of records of each user's mobility trajectory form the training set, the middle 10% of records are the valid set, and the remaining records are used for testing.

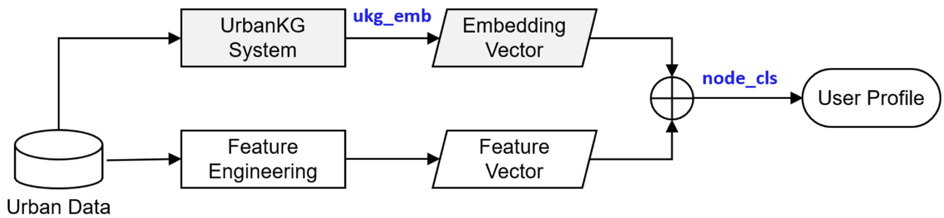


Fig. 8. An illustration of leveraging the UrbanKG system for the user profiling problem. The gray parts are removed for the without UrbanKG comparison.

Table 8. Statistics of Two Datasets for Mobility Prediction

Dataset	#Users	#POIs	#Records	Duration
Beijing	3,083	12,597	650,578	45 days
Shanghai	926	22,627	114,811	82 days

Table 9. Result Comparison of the Mobility Prediction Task

	Beijing			Shanghai		
	Acc@1 ↑	Acc@10 ↑	MRR ↑	Acc@1 ↑	Acc@10 ↑	MRR ↑
ARNN	0.442	0.685	0.532	0.379	0.605	0.469
UrbanKG	0.531	0.800	0.637	0.429	0.696	0.528

Table 9 presents the result comparison of the state-of-the-art baseline ARNN [16] with respect to accuracy and **Mean Reciprocal Rank (MRR)**. With the multi-source data fused, the UrbanKG system significantly outperforms the traditional baseline.

4.2.2 User Profiling. The user profiling problem is to infer the user demographic information from user behavior. Considering the common human mobility in the urban scenario, here we investigate the mobile user profiling problem [60], which leverages user mobility data for demographic inference. Especially, the UrbanKG-based mobile user profiling problem is stated as follows.

PROBLEM 2 (URBANKG-BASED MOBILE USER PROFILING PROBLEM). *Given the UrbanKG $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$ as well as user mobility data $\mathcal{D}_{\text{mobi}} = \{(\text{lng}_1^u, \text{lat}_1^u, \tau_1^u), \ldots, (\text{lng}_k^u, \text{lat}_k^u, \tau_k^u) | u \in \mathcal{U}\}$ with the user set of $\mathcal{U}$, each user $u \in \mathcal{U}$ corresponds to the user entity $e_u \in \mathcal{E}$ in UrbanKG. Hence, the mobile user profiling problem of inferring the demographic information y_u for user u can be formulated as the node classification problem of $y_u = f(e_u, \mathcal{D}_{\text{mobi}})$ in UrbanKG with the classification model f.*

The overall framework is illustrated in Figure 8. As for the application layer in the UrbanKG system, the user embedding can be obtained via the operation **`ukg_emb`**. Moreover, the knowledgeable embedding is concatenated with user's mobility features for classification via the operation **`node_cls`**.

To evaluate the proposed framework, we sample two mobile user profiling datasets of Beijing and Shanghai, whose basic statistics are summarized in Table 10. We randomly split the datasets into five subsets and report the average performance of fivefold cross validation.

The result comparison with the F1-score (the higher is the better) is shown in Table 11, —that is, only using features from traditional methods. The results show that UrbanKG embeddings can further improve the performance of traditional feature engineering solutions [64] across various profiles.

Table 10. Statistics of Two Datasets for Mobile User Profiling

Dataset	#Users	Profiles	#Records	Duration
Beijing	8,599	Income (5), gender (2), age (4), occupation (7)	57,794,023	92 days
Shanghai	7,733	Income (3), gender (2), age (6), occupation (3)	1,038,648	62 days

The values in brackets in the Profiles column are the numbers of classes.

Table 11. Result Comparison of the User Profiling Task

	Beijing		Shanghai	
F1-Score ↑	UrbanKG	w/o UrbanKG	UrbanKG	w/o UrbanKG
Income	0.466	0.444	0.429	0.404
Gender	0.565	0.551	0.566	0.557
Age	0.610	0.590	0.406	0.380
Occupation	0.336	0.313	0.389	0.354

The "w/o UrbanKG" columns denote the model without UrbanKG embeddings.

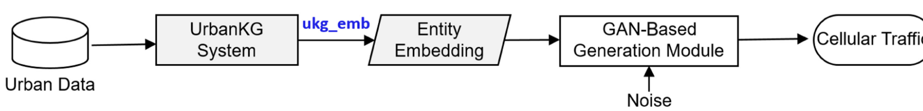


Fig. 9. An illustration of leveraging the UrbanKG system for the traffic generation problem. The gray parts are removed for the w/o UrbanKG comparison.

4.2.3 *Traffic Generation.* The traffic generation problem aims to generate population-level cellular traffic based on historical data as well as urban environmental data [31]. Especially, in this use case, we introduce base station entities into the original UrbanKG connecting with their nearby urban entities of regions and business areas. Then the UrbanKG-based traffic generation problem can be stated as follows.

Problem 3 (UrbanKG-Based Traffic Generation Problem). *Given the UrbanKG $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$ as well as cellular traffic data $\{V_t^b\}_{t=1}^T$ for base stations in source areas $b \in \mathcal{B}_{\text{source}}$, each base station $b \in \mathcal{B}_{\text{source}} \cup \mathcal{B}_{\text{target}}$ corresponds to the base station entity $e_b \in \mathcal{E}$ in UrbanKG. Hence, the traffic generation problem of generating cellular traffic $\{V_t^b\}_{t=1}^T$ for base stations in target areas $b \in \mathcal{B}_{\text{target}}$ can be formulated as the conditional generation problem with the conditional input embedding of e_b from UrbanKG.*

The overall framework is illustrated in Figure 9. In this case, we design a GAN-based model to simulate the traffic patterns and further leverage the UrbanKG system to distill multiple factors affecting cellular traffic in the surrounding urban environment. Then, the application layer calls the operation **ukg_emb** to provide knowledgeable embeddings for the generated model.

To evaluate the proposed framework, we sample a dataset with network traffic records of 5,326 base stations in Shanghai. The collected time spans 1 month. Especially, the dataset is divided into three sub-datasets of base stations in the center area, suburb area, and outer suburb area. Then, we train the proposed framework on each sub-dataset and test the trained model on the other sub-datasets with average performance reported.

Table 12 presents the result comparison of the traffic generation task on traffic volume, variation, and daily periodicity with metrics of **Jensen-Shannon Divergence (JSD)** and root mean square error (RMSE). The "w/o UrbanKG" row corresponds to randomly initializing input without KG

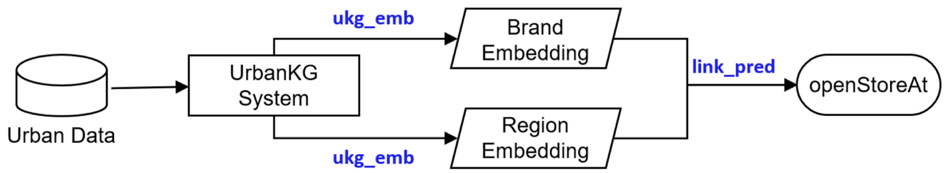


Fig. 10. An illustration of leveraging the UrbanKG system for the site selection problem.

Table 12. Result Comparison of the Traffic Generation Task

	Traffic Volume	**First-Order Difference**	**Daily Frequency Component**
	JSD ↓	JSD ↓	RMSE ↓
UrbanKG	0.2879	0.0744	0.0201
w/o UrbanKG	0.3072	0.0850	0.0211

The "w/o UrbanKG" row denotes the model without UrbanKG embeddings.

Table 13. Statistics of Two Datasets for Site Selection

Dataset	**Brand**	**Region**	**Train**	**Valid**	**Test**
Beijing	398	528	15,022	5,007	5,008
Shanghai	441	2,042	29,006	9,669	9,669

embedding. It can be observed that the UrbanKG system successfully distills useful environmental knowledge and enhances the performance of the traditional GAN-based solution.

4.2.4 Site Selection. The site selection use case [34] focuses on the city-level problem, which determines candidate regions for various brands opening stores. Especially, such problem requires providing explainable results with multiple site selection factors considered, which can be stated as follows.

PROBLEM 4 (URBANKG-BASED SITE SELECTION PROBLEM). *Given the UrbanKG $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$ with brand entities $b \in \mathcal{B}$ and region entities $a \in \mathcal{A}$ therein, the site selection problem investigates if the brand $b \in \mathcal{B}$ should open a new store at corresponding region $a \in \mathcal{A}$. Hence, the UrbanKG-based site selection problem can be formulated as the link prediction problem of measuring the validity of $(e_b, r_{\text{openStoreAt}}, e_a)$ in UrbanKG.*

The overall framework is illustrated in Figure 10. First, by calling the operation **`link_pred`**, the UrbanKG system formulates the problem into a link prediction problem with relation `openStoreAt` between brand entities and region entities. Moreover, various site selection strategies can be modeled as relational paths, and a task-specific KG representation algorithm can be fed into the system via the augment `alg`.

To evaluate the proposed framework, we sample two site selection datasets of Beijing and Shanghai, whose basic statistics are summarized in Table 13. We randomly split the dataset into train/valid/test sets by a proportion of 6:2:2.

Table 14 presents the result comparison of the best traditional solution NeuMF-RS [26] with respect to NDCG, hit ratio, and precision. The results further demonstrate the effectiveness of the UrbanKG system, which not only fuses the urban data from multiple sources but also distills task-specific knowledge via algorithm implementation.

To summarize, the UrbanKG system supports various applications in the urban scenario across multiple using ways and scales, as shown in Table 7. The preceding use case analysis provides a strong validation of effectiveness and wide applicability in practice.

Table 14. Result Comparison of the Site Selection Task

	Beijing			Shanghai		
	N@10 ↑	H@10 ↑	P@10 ↑	N@10 ↑	H@10 ↑	P@10 ↑
NeuMF-RS	0.178	0.653	0.155	0.168	0.615	0.148
UrbanKG	0.219	0.713	0.186	0.205	0.671	0.177

Table 15. Comparison of Related Works on Geographic KG and UrbanKG

KG	Venue	Entity Types	Relation Types	Applications	APIs
LGD [3]	ISWC '09	Country, city, POI	Spatiality, attribute, taxonomy	Semantic-spatial search	Query
GeoKG [62]	ISPRS '19	Geographic object (e.g., Yangzi River)	Spatiality, attribute, time, state	Geographic question answering	Query
Spindra [45]	ICDE '19	Region, food type, POI	Spatiality, taxnomoy	Location-aware search	Query
WorldKG [9]	CIKM '21	Country, point, geographic object	Attribute, taxonomy	POI recommendation	Query
UrbanKG	–	POI, region, business area, brand, category, image, user	Spatiality, affiliation, function, person, cross-modality	Individual/ population/ city-level applications	Query, embedding, node classification, link prediction, graph pooling

5 RELATED WORK

Here we summarize the related work into two aspects of KG-based systems and urban computing based systems.

KG-Based Systems. Traditional KG-based systems include Freebase [5], DBpedia [25], Wikidata [49], WordNet [39], and YAGO [11], among others. These systems mainly focus on general-purpose or encyclopedia-based knowledge, constructed from massive unstructured text data as well as structured semantic web data like Wikipedia. For example, the former three systems harvest structured knowledge via individual contributions on Wikipedia and WordNet provides formal linguistic knowledge on words, whereas YAGO automatically extracts Wikipedia facts and unifies with WordNet by rule-based and heuristic methods. Moreover, several geographic KG-based systems have been proposed recently [3, 9, 45, 62]. For example, LinkedGeoData (LGD) [3] and WorldKG [9] investigate transforming OpenStreetMap data into a KG. Both GeoKG [62] and Spindra [45] build a KG for geographic knowledge management with location-aware queries supported. Especially, Table 15 compares the proposed UrbanKG with a geographic KG in existing studies. According to the table, these KG-based systems are built upon general concepts or geographic concepts, but they ignore the real entities in the urban scenario, thus failing to support urban computing applications. Moreover, the UrbanKG system supports far more applications and provides more comprehensive APIs.

Urban Computing Based Systems. With the development of urban computing in past years [70, 75], recent studies also design systems for spatial-temporal data management [1, 38, 44, 55]. City-Eyes [70] supports the real-time display for data visualization in the city. JUST [27] proposes a holistic distributed system to manage spatial-temporal data with indexing techniques developed, whereas both JUST-Traj [17] and TrajMesa [28] focus on trajectory data management. LibCity [55] develops a standardized framework for traffic prediction [21, 22], and DeepMob [43, 44] is designed for urban emergency management in the city [23, 53]. Nevertheless, these systems emphasize the spatial-temporal data management, especially the trajectory data, but they largely ignore the urban data from other sources like road network and cross-modal data. In comparison to our investigation, our designed UrbanKG system is the first, to the best of our knowledge, to introduce the KG for urban data fusion as well as knowledge distillation, which provides an effective and flexible platform for urban computing.

6 CONCLUSION

In this article, we presented the UrbanKG system, a KG-based system for the urban scenario. The system develops a systematic scheme to construct KG from the urban data in different structures and modalities with data fusion achieved. Moreover, the multiple layers of storage, algorithm, operation, and application are built upon the constructed UrbanKG to provide user-friendly services. Several representative use cases demonstrate the system capability of enhancing various urban applications, which has the potential to be applied in various urban computing research.

REFERENCES

[1] Md. Mahbub Alam, Luis Torgo, and Albert Bifet. 2021. A survey on spatio-temporal data analytics systems. *ACM Computing Surveys* 54, 10s (2021), Article 219, 38 pages.

[2] K. M. Annervaz, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati. 2018. Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. *arXiv preprint arXiv:1802.05930* (2018).

[3] Sören Auer, Jens Lehmann, and Sebastian Hellmann. 2009. LinkedGeoData: Adding a spatial dimension to the web of data. In The Semantic Web—ISWC 2009. Lecture Notes in Computer Science, Vol. 5823. Springer, 731–746.

[4] Ivana Balažević, Carl Allen, and Timothy M. Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of EMNLP 2019*.

[5] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of SIGMOD 2008*. 1247–1250.

[6] Antoine Bordes, Nocolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26 (NIPS'13)*.

[7] Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2022. Temporal knowledge graph completion: A survey. *arXiv preprint arXiv:2201.08236* (2022).

[8] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D knowledge graph embeddings. In *Proceedings of AAAI 2018*, Vol. 32.

[9] Alishiba Dsouza, Nicolas Tempelmeier, Ran Yu, Simon Gottschalk, and Elena Demidova. 2021. WorldKG: A world-scale geographic knowledge graph. In *Proceedings of CIKM 2021*. 4475–4484.

[10] Orri Erling and Ivan Mikhailov. 2009. RDF support in the virtuoso DBMS. In *Networked Knowledge-Networked Media*. Springer, 7–24.

[11] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A core of semantic knowledge unifying WordNet and Wikipedia. In *Proceedings of WWW 2007*. 697–706.

[12] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting human mobility with attentional recurrent networks. In *Proceedings of WWW 2018*.

[13] Yupeng Fu and Chinmay Soman. 2021. Real-time data infrastructure at Uber. In *Proceedings of SIGMOD 2021*. 2503–2516.

[14] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Erez Lieberman Aiden, and Li Fei-Fei. 2017. Using deep learning and Google Street View to estimate the demographic makeup of neighborhoods across the United States. *PNAS* 114, 50 (2017), 13108–13113.

[15] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2022. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering* 34 (2022), 3549–3568.

[16] Qing Guo, Zhu Sun, Jie Zhang, and Yin-Leng Theng. 2020. An attentional recurrent neural network for personalized next location recommendation. In *Proceedings of AAAI 2020*, Vol. 34. 83–90.

[17] Huajun He, Ruiyuan Li, Jie Bao, Tianrui Li, and Yu Zheng. 2021. JUST-Traj: A distributed and holistic trajectory data management system. In *Proceedings of ICAGIS 2021*.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of CVPR 2016*. 770–778.

[19] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, et al. 2021. Knowledge graphs. *ACM Computing Surveys* 54, 5 (2021), Article 71, 37 pages.

[20] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of WSDM 2019*. 105–113.

[21] Jiahao Ji, Jingyuan Wang, Zhe Jiang, Jiawei Jiang, and Hu Zhang. 2022. STDEN: Towards physics-guided neural networks for traffic flow prediction. In *Proceedings of AAAI 2022*, Vol. 36. 4048–4056.
[22] Jiahao Ji, Jingyuan Wang, Zhe Jiang, Jingtian Ma, and Hu Zhang. 2020. Interpretable spatiotemporal deep learning model for traffic flow prediction based on potential energy fields. In *Proceedings of ICDM 2020*. IEEE, Los Alamitos, CA, 1076–1081.
[23] Jiahao Ji, Jingyuan Wang, Junjie Wu, Boyang Han, Junbo Zhang, and Yu Zheng. 2022. Precision CityShield against hazardous chemicals threats via location mining and self-supervised learning. In *Proceedings of KDD 2022*. 3072–3080.
[24] Michael Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P. Xing. 2019. Rethinking knowledge graph propagation for zero-shot learning. In *Proceedings of CVPR 2019*. 11487–11496.
[25] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, et al. 2015. DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195.
[26] Nuo Li, Bin Guo, Yan Liu, Yao Jing, Yi Ouyang, and Zhiwen Yu. 2018. Commercial site recommendation based on neural collaborative filtering. In *Proceedings of UbiComp 2018*.
[27] Ruiyuan Li, Huajun He, Rubin Wang, Yuchuan Huang, Junwen Liu, Sijie Ruan, Tianfu He, Jie Bao, and Yu Zheng. 2020. JUST: JD Urban Spatio-Temporal data engine. In *Proceedings of ICDE 2020*.
[28] Ruiyuan Li, Huajun He, Rubin Wang, Sijie Ruan, Tianfu He, Jie Bao, Junbo Zhang, Liang Hong, and Yu Zheng. 2023. TrajMesa: A distributed NoSQL-based trajectory data management system. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2023), 1013–1027.
[29] Shuangli Li, Jingbo Zhou, Tong Xu, Hao Liu, Xinjiang Lu, and Hui Xiong. 2020. Competitive analysis for points of interest. In *Proceedings of KDD 2020*. 1265–1274.
[30] Ziqian Lin, Jie Feng, Ziyang Lu, Yong Li, and Depeng Jin. 2019. DeepSTN+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. In *Proceedings of AAAI 2019*, Vol. 33. 1020–1027.
[31] Zinan Lin, Alankar Jain, Chen Wang, Giulia Fanti, and Vyas Sekar. 2020. Using GANs for sharing networked time series data: Challenges, initial promise, and open questions. In *Proceedings of IMC 2020*. 464–483.
[32] Chang Liu, Chen Gao, Depeng Jin, and Yong Li. 2021. Improving location recommendation with urban knowledge graph. *arXiv preprint arXiv:2111.01013* (2021).
[33] Jia Liu, Tianrui Li, Shenggong Ji, Peng Xie, Shengdong Du, Fei Teng, and Junbo Zhang. 2021. Urban flow pattern mining based on multi-source heterogeneous data fusion and knowledge graph embedding. *IEEE Transactions on Knowledge and Data Engineering* 35, 2 (2021), 2133–2146.
[34] Yu Liu, Jingtao Ding, and Yong Li. 2021. Knowledge-driven site selection via urban knowledge graph. *arXiv preprint arXiv:2111.00787* (2021).
[35] Yu Liu, Jingtao Ding, and Yong Li. 2022. Developing knowledge graph based system for urban computing. In *Proceedings of GeoKG 2022*.
[36] Yu Liu, Xin Zhang, Jingtao Ding, Yanxin Xi, and Yong Li. 2023. Knowledge-infused contrastive learning for urban imagery-based socioeconomic prediction. In *Proceedings of WWW 2023*.
[37] Thomas Lodato, Emma French, and Jennifer Clark. 2021. Open government data in the smart city: Interoperability, urban knowledge, and linking legacy systems. *Journal of Urban Affairs* 43, 4 (2021), 586–600.
[38] Zijian Ma, Dan Lu, Qian Liu, Jingyuan Wang, and Zhang Xiong. 2017. City-Eyes: A multi-source data integration basec smart city analysis system. In *Proceedings of IEEE WoWMoM 2017*. 1–3.
[39] George A. Miller. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.
[40] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2019. You can teach an old dog new tricks! On training knowledge graph embeddings. In *Proceedings of ICLR 2019*.
[41] Hongzhi Shi, Qi Guo, Yaguang Li, Lingyu Zhang, Jieping Ye, Yong Li, and Yan Liu. 2020. Predicting origin-destination flow via multi-perspective graph convolutional network. In *Proceedings of ICDE 2020*.
[42] Amit Singhal. 2012. Introducing the knowledge graph: Things, not strings. *Official Google Blog* 5 (2012), 16.
[43] Xuan Song, Ryosuke Shibasaki, Nicholos Jing Yuan, Xing Xie, Tao Li, and Ryutaro Adachi. 2017. DeepMob: Learning deep knowledge of human emergency behavior and mobility from big and heterogeneous data. *ACM Transactions on Information Systems* 35, 4 (2017), Article 41, 19 pages.
[44] Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, and Ryosuke Shibasaki. 2014. Intelligent system for urban emergency management during large-scale disaster. In *Proceedings of AAAI 2014*, Vol. 28.
[45] Yuhan Sun, Jia Yu, and Mohamed Sarwat. 2019. Demonstrating Spindra: A geographic knowledge graph management system. In *Proceedings of ICDE 2019*. IEEE, Los Alamitos, CA, 2044–2047.
[46] Jiyuan Tan, Qianqian Qiu, Weiwei Guo, and Tingshuai Li. 2021. Research on the construction of a knowledge graph and knowledge reasoning model in the field of urban traffic. *Sustainability* 13, 6 (2021), 3191.
[47] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of ICML 2016*.

[48] Ke Tu, Peng Cui, Daixin Wang, Zhiqiang Zhang, Jun Zhou, Yuan Qi, and Wenwu Zhu. 2021. Conditional graph attention networks for distilling and refining knowledge graphs in recommendation. In *Proceedings of CIKM 2021*. 1834–1843.

[49] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Communications of the ACM* 57, 10 (2014), 78–85.

[50] Dongjie Wang, Yanjie Fu, Pengyang Wang, Bo Huang, and Chang-Tien Lu. 2020. Reimagining city configuration: Automated urban planning via adversarial learning. In *Proceedings of ICAGIS 2020*. 497–506.

[51] Hongjian Wang, Daniel Kifer, Corina Graif, and Zhenhui Li. 2016. Crime rate inference with big data. In *Proceedings of KDD 2016*. 635–644.

[52] Huandong Wang, Qiaohong Yu, Yu Liu, Depeng Jin, and Yong Li. 2021. Spatio-temporal urban knowledge graph enabled mobility prediction. In *Proceedings of UbiComp 2021*.

[53] Jingyuan Wang, Chao Chen, Junjie Wu, and Zhang Xiong. 2017. No longer sleeping with a bomb: A duet system for protecting urban safety from dangerous goods. In *Proceedings of KDD 2017*. 1673–1681.

[54] Jingyuan Wang, Jiahao Ji, Zhe Jiang, and Leilei Sun. 2022. Traffic flow prediction based on spatiotemporal potential energy fields. *IEEE Transactions on Knowledge and Data Engineering*. Early access, November 10, 2022.

[55] Jingyuan Wang, Jiawei Jiang, Wenjun Jiang, Chao Li, and Wayne Xin Zhao. 2021. LibCity: An open library for traffic prediction. In *Proceedings of SIGSPATIAL 2021*. 145–148.

[56] Jingyuan Wang, Xin Lin, Yuxi Liu, Kai Feng, and Hui Lin. 2020. A knowledge transfer model for COVID-19 predicting and non-pharmaceutical intervention simulation. *arXiv preprint arXiv:2004.12433* (2020).

[57] Jingyuan Wang, Xiaojian Wang, and Junjie Wu. 2018. Inferring metapopulation propagation network for intra-city epidemic control and prevention. In *Proceedings of KDD 2018*. 830–838.

[58] Jingyuan Wang, Ning Wu, and Wayne Xin Zhao. 2021. Personalized route recommendation with neural network enhanced search algorithm. *IEEE Transactions on Knowledge and Data Engineering* 34, 12 (2021), 5910–5924.

[59] Jingyuan Wang, Ning Wu, Wayne Xin Zhao, Fanzhang Peng, and Xin Lin. 2019. Empowering A* search algorithms with neural networks for personalized route recommendation. In *Proceedings of KDD 2019*. 539–547.

[60] Pengyang Wang, Kunpeng Liu, Lu Jiang, Xiaolin Li, and Yanjie Fu. 2020. Incremental mobile user profiling: Reinforcement learning with spatial knowledge graph for modeling event streams. In *Proceedings of KDD 2020*. 853–861.

[61] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.

[62] Shu Wang, Xueying Zhang, Peng Ye, Mi Du, Yanxu Lu, and Haonan Xue. 2019. Geographic knowledge graph (GeoKG): A formalized geographic knowledge representation. *ISPRS International Journal of Geo-Information* 8, 4 (2019), 184.

[63] Yuandong Wang, Hongzhi Yin, Hongxu Chen, Tianyu Wo, Jie Xu, and Kai Zheng. 2019. Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling. In *Proceedings of KDD 2019*. 1227–1235.

[64] Lun Wu, Liu Yang, Zhou Huang, Yaoli Wang, Yanwei Chai, Xia Peng, and Yu Liu. 2019. Inferring demographics from human trajectories and geographical context. *Computers, Environment and Urban Systems* 77 (2019), 101368.

[65] Ning Wu, Jingyuan Wang, Wayne Xin Zhao, and Yang Jin. 2019. Learning to effectively estimate the travel time for fastest route recommendation. In *Proceedings of CIKM 2019*. 1923–1932.

[66] Yanxin Xi, Tong Li, Huandong Wang, Yong Li, Sasu Tarkoma, and Pan Hui. 2022. Beyond the first law of geography: Learning representations of satellite imagery by leveraging point-of-interests. In *Proceedings of WWW 2022*. 3308–3316.

[67] Anze Xie, Anders Carlsson, Jason Mohoney, Roger Waleffe, Shanan Peters, Theodoras Rekatsinas, and Shivaram Venkataraman. 2021. Demo of Marius: A system for large-scale graph embeddings. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2759–2762.

[68] Yanyan Xu, Luis E. Olmos, Sofiane Abbar, and Marta C. Gonzalez. 2020. Deconstructing laws of accessibility and facility distribution in cities. *Science Advances* 6, 37 (2020), eabb4112.

[69] Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. 2018. Deep distributed fusion network for air quality prediction. In *Proceedings of KDD 2018*. 965–973.

[70] ChuanTao Yin, Zhang Xiong, Hui Chen, Jingyuan Wang, Daven Cooper, and Bertrand David. 2015. A literature survey on smart cities. *Science China Information Sciences* 58, 10 (2015), 1–18.

[71] Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2020. JAKET: Joint pre-training of knowledge graph and language understanding. *arXiv preprint arXiv:2010.00796* (2020).

[72] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022. A survey of knowledge-enhanced text generation. *ACM Computing Surveys* 54, 11s (2022), Article 227, 38 pages.

[73] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys* 52, 1 (2019), Article 5, 38 pages.

[74] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of CVPR 2017*. 2881–2890.

[75] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology* 5, 3 (2014), Article 38, 55 pages.

[76] Zhilun Zhou, Yu Liu, Jingtao Ding, Depeng Jin, and Yong Li. 2023. Hierarchical knowledge graph learning enabled socioeconomic indicator prediction in location-based social network. In *Proceedings of WWW 2023*.

[77] Fengbin Zhu, Wenquiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774* (2021).

Received 3 October 2022; revised 10 February 2023; accepted 3 March 2023